

IBM[®] DB2 Universal Database[™]



Data Warehouse Center Administration Guide

Version 8

IBM[®] DB2 Universal Database[™]



Data Warehouse Center Administration Guide

Version 8

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 - 2002. 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	xiii
Who should read this book.	xiii
Prerequisite publications	xiii
Chapter 1. About data warehousing.	1
What is data warehousing?	1
Warehouse tasks	2
Data warehouse objects	2
Subject areas	2
Warehouse sources	3
Warehouse targets	3
Warehouse agents and agent sites	3
Processes and steps	4
Chapter 2. Setting up your warehouse.	7
Building a data warehouse	7
Starting the Data Warehouse Center	7
Starting and stopping the Data Warehouse Center server and logger	8
Starting and stopping the warehouse server and logger (Windows NT, Windows 2000, Windows XP)	8
Starting and stopping the warehouse server and logger (AIX)	8
Verifying that the warehouse server and logger daemons are running (AIX)	9
Starting the warehouse agent daemon	10
Warehouse agent daemons	10
Connectivity requirements for the warehouse server and the warehouse agent	10
Starting the warehouse agent daemon (Windows NT, Windows 2000, or Windows XP)	10
Starting the iSeries warehouse agent daemon	11
Verifying that the iSeries warehouse agent started	11
Verifying that the iSeries warehouse agent daemon is still running	11
Configuring TCP/IP on z/OS	12
Starting the zSeries agent daemon in the foreground	12
Starting the zSeries warehouse agent daemon in the background	13
Verifying that the zSeries warehouse agent is running.	13
Verifying communication between the warehouse server and the warehouse agent	14
Stopping the warehouse agent daemon	15
Stopping the warehouse agent daemon (Windows NT, Windows 2000, Windows XP)	15
Stopping the warehouse agent daemon (AIX, Solaris Operating Environment, Linux)	15
Stopping the iSeries warehouse agent daemon	16
Stopping the warehouse agent daemon (zSeries)	16
Defining agent sites	16
Defining an agent site.	17
Agent site configurations.	17
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 NT, Windows 2000, Windows XP)	26
Setting up connectivity for DB2 DRDA databases (Windows NT, Windows 2000, Windows XP)	27
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).	29
Setting up connectivity for a DB2 Universal Database source (Solaris Operating Environment, Linux)	29

Setting up connectivity for a DRDA data source (Solaris Operating Environment, Linux)	30	Supported non-DB2 data sources	43
Setting up connectivity for DB2 sources (OS/2)	31	Warehouse agent support for non-DB2 sources.	43
Setting up connectivity for a DB2 Universal Database source (OS/2).	31	Column size truncation for non-DB2 warehouse sources	44
Setting up connectivity for a DRDA data source (OS/2)	31	Setting up connectivity for non-DB2 sources (Windows NT, Windows 2000, Windows XP)	45
Accessing source databases	32	Setting up connectivity for a Sybase Adaptive Server source – without client (Windows NT, Windows 2000, Windows XP)	45
Setting up access to DB2 Universal Database data sources.	32	Setting up connectivity for an Oracle source (Windows NT, Windows 2000, Windows XP)	45
Defining privileges to DB2 Universal Database data sources.	32	Setting up connectivity for an Informix 9.2 source – with client (Windows NT, Windows 2000, Windows XP)	46
Establishing connectivity to DB2 Universal Database sources	33	Setting up connectivity for an Informix 9.2 source – without client (Windows, NT, Windows 2000, Windows XP)	47
Setting up access to DB2 DRDA data sources.	33	Setting up connectivity for a Microsoft SQL Server data source (Windows NT, Windows 2000, Windows XP)	47
Defining privileges to DB2 DRDA data sources.	34	Setting up connectivity for a Microsoft Access source (Windows NT, Windows 2000, Windows XP)	48
Setting up a DB2 Connect gateway site (Windows NT, Windows 2000, Windows XP)	35	Setting up connectivity for a Microsoft Excel data source (Windows NT, Windows 2000, Windows XP)	48
Connecting to DB2 DRDA data sources	35	Setting up connectivity for an IMS, or a VSAM data source (Windows NT, Windows 2000, Windows XP)	49
Connectivity requirements for remote databases	36	Managing the warehouse agent for non-DB2 sources.	50
Connectivity requirements for remote databases from the iSeries warehouse agent	36	Verifying the warehouse agent (AIX, Solaris Operating Environment, Linux)	50
Connecting to local and remote databases from the iSeries warehouse agent	37	Switching between versions of the warehouse agent	50
Example of how to catalog local and remote database names for the iSeries warehouse agent	38	Setting up connectivity for non-DB2 sources (AIX)	51
Viewing, adding, changing, and removing remote relational database directory entries	38	Setting up connectivity for a Sybase Adaptive Server (without client) source (AIX)	51
Requirements for accessing remote data sources with the zSeries warehouse agent	39	Setting up connectivity for an Oracle source (AIX)	51
Requirements for accessing DB2 Relational Connect data sources with the zSeries warehouse agent	39	Setting up connectivity for an Informix 9.2 source – with client (AIX)	52
DB2 warehouse sources	39	Setting up connectivity for an Informix 9.2 source – without client (AIX)	53
Ordinary identifiers	39		
Delimited identifiers	40		
Importing metadata from tables	40		
Agent sites	40		
Defining DB2 warehouse sources	40		
Chapter 4. Setting up non-DB2 warehouse sources	43		

Setting up connectivity for a Microsoft SQL Server source (AIX)	54	Configuring the ODBC driver for a Microsoft SQL Server (Windows NT, Windows 2000, Windows XP)	68
Setting up connectivity for non-DB2 sources (Solaris Operating Environment, Linux)	54	Configuring a Microsoft Access client (Windows NT, Windows 2000, Windows XP)	69
Setting up connectivity for a Sybase Adaptive Server source – without client (Solaris Operating Environment, Linux)	54	Creating a Microsoft Access source database (Windows NT, Windows 2000, Windows XP)	70
Setting up connectivity for an Oracle 8 data source (Solaris Operating Environment, Linux)	55	Creating a target warehouse database (Windows NT, Windows 2000, Windows XP)	70
Setting up connectivity for an Informix 9.2 source – with client (Solaris Operating Environment, Linux)	56	Defining a warehouse that uses the Microsoft Access and warehouse target databases (Windows NT, Windows 2000, Windows XP)	71
Setting up connectivity for an Informix 9.2 source – without client (Solaris Operating Environment)	57	Importing table definitions from a Microsoft Access database (Windows NT, Windows 2000, Windows XP)	72
Setting up connectivity for a Microsoft SQL Server source (Solaris Operating Environment, Linux)	58	Cataloging a Microsoft Access source database in ODBC (Windows NT, Windows 2000, Windows XP)	73
Configuring non-DB2 warehouse sources	59	Cataloging a target warehouse database for use with a Microsoft Access source database (Windows NT, Windows 2000, Windows XP)	74
Configuring an Informix 9.2 client (Windows NT, Windows 2000, Windows XP)	59	Using a Microsoft Excel spreadsheet as a warehouse source (Windows NT, Windows 2000, Windows XP)	75
Configuring an Informix 9.2 server – with client (Windows NT, Windows 2000, Windows XP)	59	Cataloging a Microsoft Excel data source in ODBC (Windows NT, Windows 2000, Windows XP)	75
Configuring a host for an Informix 9.2 source — with client (Windows NT, Windows 2000, Windows XP)	60	Creating named tables for Microsoft Excel data sources (Windows NT, Windows 2000, Windows XP)	76
Configuring the ODBC driver for Informix 9.2 – without client (Windows NT, Windows 2000, Windows XP)	61	Creating a target warehouse database for use with a Microsoft Excel data source (Windows NT, Windows 2000, Windows XP)	76
Configuring the Informix 9.2 server – without client (Windows NT, Windows 2000, Windows XP)	62	Cataloging a target warehouse database for use with a Microsoft Excel data source (Windows NT, Windows 2000, Windows XP)	77
Configuring the ODBC driver for a Sybase Adaptive Server source – without client (Windows NT, Windows 2000, Windows XP)	62	Defining sources and targets to the Data Warehouse Center that use a Microsoft Excel data source (Windows NT, Windows 2000, Windows XP)	77
Configuring an Oracle 8 client (Windows NT, Windows 2000, Windows XP)	64	Accessing an IMS or VSAM source database from the Data Warehouse Center	78
Configuring the ODBC driver for an Oracle 8 client (Windows NT, Windows 2000, Windows XP)	67		
Testing an Oracle 8 client configuration (Windows NT, Windows 2000, Windows XP)	67		
Configuring a Microsoft SQL Server client (Windows NT, Windows 2000, Windows XP)	68		

Configuring non-DB2 warehouse sources (AIX, Linux, Solaris Operating Environment)	79	Requirements for accessing a remote file from a file server (Windows NT, Windows 2000, Windows XP)	91
Configuring an Informix 9.2 client (AIX, Solaris Operating Environment)	79	Setting up connectivity for a remote file source (Windows NT, Windows 2000, Windows XP)	91
SQL Hosts file	79	Setting up connectivity for file sources (AIX)	92
Configuring the ODBC driver for accessing an Informix 9.2 database (AIX, Solaris Operating Environment)	79	Setting up connectivity to a z/OS or VM file source (AIX)	92
Configuring the ODBC driver for the Sybase Adaptive Server – without client (AIX, Linux, Solaris Operating Environment)	80	Setting up connectivity for a local file source (AIX)	93
Configuring an Oracle 8 client (AIX, Linux, Solaris Operating Environment)	81	Setting up connectivity for a remote file source (AIX)	93
Example of an entry for a tnsnames.ora file (AIX, Linux, Solaris Operating Environment)	81	Setting up connectivity for file sources (Solaris Operating Environment, Linux)	94
Configuring the ODBC driver for an Oracle 8 database (AIX, Linux, Solaris Operating Environment)	82	Setting up connectivity for a z/OS or VM file source (Solaris Operating Environment, Linux)	94
Configuring a Microsoft SQL Server client (AIX, Linux, Solaris Operating Environment)	82	Setting up connectivity for a local file source (Solaris Operating Environment, Linux)	94
Configuring the ODBC driver for accessing a Microsoft SQL Server database (AIX, Linux, Solaris Operating Environment)	83	Setting up connectivity for a remote file source (Solaris Operating Environment, Linux)	95
Defining a non-DB2 warehouse source in the Data Warehouse Center	83	Example .odbc.ini file entry for a warehouse file source (AIX, Solaris Operating Environment, Linux)	95
Specifying database information for a non-DB2 warehouse source in the Data Warehouse Center	84	Network File System (NFS) protocol	96
Defining warehouse sources for use with DB2 Relational Connect	85	Accessing data files with FTP	96
Server mappings and nickname tables for warehouse sources accessed by DB2 Relational Connect	86	Accessing data files with the Copy file using FTP warehouse program	97
Defining source tables for DB2 Relational Connect warehouse sources	87	Defining a file source to the Data Warehouse Center	98
Chapter 5. Setting up warehouse file sources	89	Chapter 6. Setting up access to a warehouse	99
Warehouse agent support for file sources	89	Setting up a DB2 Universal Database warehouse	99
Setting up connectivity for file sources (Windows NT, Windows 2000, Windows XP)	89	Defining privileges for DB2 Universal Database warehouses	99
Setting up connectivity for a z/OS or VM file source (Windows NT, Windows 2000, Windows XP)	90	Connecting to DB2 Universal Database and DB2 Enterprise Server Edition warehouses	100
Setting up connectivity for a local file source (Windows NT, Windows 2000, Windows XP)	90	Setting up a DB2 for iSeries warehouse	100
		Defining privileges to DB2 for iSeries warehouses	100
		Setting up a DB2 Connect gateway site (iSeries)	101
		Connecting to a DB2 for iSeries warehouse with DB2 Connect	101

Checking the current level of your iSeries operating system	102	User-defined programs	132
Verifying that all of the servers in an iSeries system are running	102	Column mapping	132
Setting up access to a DB2 for iSeries warehouse with iSeries Client Access/400.	103	Running warehouse steps	134
Setting up a DB2 for z/OS warehouse	104	Step promotion and testing	134
Setting up a DB2 Universal Database for z/OS warehouse	104	Verifying the results of a step that is run in test mode	135
Defining privileges for a DB2 for z/OS warehouse	104	Running steps that use transient tables as sources	135
Connecting to a DB2 for z/OS warehouse database	104	Running a step from outside the Data Warehouse Center	136
Defining a DB2 for z/OS warehouse to the Data Warehouse Center	105	Syntax for starting the external trigger server	136
Creating the Data Warehouse Center status table	105	Syntax for starting the external trigger client	137
Setting up a DB2 Enterprise Server Edition warehouse	106	Data Warehouse Center messages for the external trigger program	138
Defining privileges to the DB2 Enterprise Server Edition database	107	Printing step information to a text file	139
Defining the DB2 Enterprise Server Edition database to the Data Warehouse Center	107	Promoting warehouse processes	139
Defining a warehouse for use with DB2 Relational Connect	108	About promoting warehouse processes	139
Creating target tables in a remote database	110	Managing step task flow	139
Moving a target table from a DB2 Relational Connect database to a remote database	111	Step task flow	140
Updating a table in a remote database	112	Cascading processes	140
		Process task flow	140
		Monitoring the progress of steps and processes	141
		Monitoring the progress of a step or process using the Work in Progress window	141
		Step and process error messages	142
		Sampling data	142
Chapter 7. Setting up a warehouse target	113	Chapter 9. Selecting, inserting and updating source data in a target table	143
Warehouse targets	113	SQL steps	143
Supported warehouse targets	114	Defining an SQL step	143
Defining a warehouse target	115	Incremental commit	144
Warehouse primary and foreign keys	116	Joining source columns	145
		Defining a join	145
Chapter 8. Defining and running processes	117	Removing a join	146
Warehouse processes	117	Transforming codes	146
Defining the transformation and movement of data within the Data Warehouse Center	118	About adding nulls to joins	147
Warehouse steps	118	Star joins	149
Warehouse file programs	121	Filtering data	150
DB2 warehouse utilities	122	Specifying search conditions when filtering data	151
DB2 OLAP Server programs	123	Creating composite summary steps	151
Replication programs	125	Defining a calculated column	152
Warehouse Deprecated programs	126	Adding a calculated column to a step	152
Warehouse transformers	127	Adding a function to an expression in the Expression Builder	153
Statistical transformers	129		

Adding a constant to an expression	154	Setting up replication in the Data Warehouse Center	177
Chapter 10. Loading and exporting data	155	Replication source tables in the Data Warehouse Center	178
Data Warehouse Center load and export utilities	155	Creating replication control tables	178
Exporting data	155	Defining a replication step in the Data Warehouse Center	179
Defining values for a DB2 UDB export utility	155	Using a replication step in a process	179
Defining values for the Data export with ODBC to file utility	156	Replication password files	180
Loading data	157	Chapter 12. Transforming data.	183
Defining values for a DB2 Universal Database load utility	157	Using transformers in the Data Warehouse Center	183
Defining a DB2 for iSeries Data Load Insert utility	158	Transforming target tables	183
Defining a DB2 for iSeries Data Load Replace utility	160	Clean Data transformer	184
Modstring parameters for DB2 for iSeries Load utilities	161	Rules tables for Clean Data transformers	188
Trace files for the DB2 for iSeries Load utilities	162	Key columns	190
Viewing trace files for the DB2 for iSeries Load utilities	163	Period table.	191
Defining a file extension to Client Access/400	164	Invert Data transformer.	191
Defining a DB2 for z/OS Load utility	165	Pivot data transformer	192
Copying data between DB2 utilities.	166	FormatDate transformer	193
SAP R/3	167	Changing the format of a date field.	193
Loading data into the Data Warehouse Center from an SAP R/3 system	167	Specifying the input format for a date field	194
Defining an SAP source in the Data Warehouse Center	168	Specifying the output format for a date field	194
Defining the properties of SAP source business objects in the Data Warehouse Center	169	Cleaning name and address data.	194
WebSphere Site Analyzer	169	Cleaning name and address data with the Trillium Software System	194
Loading data from a WebSphere Site Analyzer database into the Data Warehouse Center	169	Trillium Software System components	196
Defining a WebSphere Site Analyzer source in the Data Warehouse Center	170	Trillium metadata.	196
		Importing Trillium metadata	197
		Trillium Batch System JCL files	198
		Example: Job step that includes a SYSTEM DD statement	199
		Trillium Batch System user-defined program	199
		Parameters for the Trillium Batch System script or JCL	199
		Error handling for Trillium Batch System programs	200
		Error return codes for name and address cleansing.	201
Chapter 11. Moving files and tables	173	Chapter 13. Calculating statistics.	203
Manipulating files using FTP or the Submit JCL jobstream warehouse program	173	Defining statistical transformers in the Data Warehouse Center	203
Unable to access a remote file on a secure UNIX or UNIX System Services system	175	ANOVA transformer.	203
Replication	175	Calculate Statistics transformer	204
Replication in the Data Warehouse Center	175	Calculate Subtotals transformer	205

Chi-square transformer	206
Correlation analysis	207
Moving averages	209
Regression transformer	210

Chapter 14. Exporting and importing metadata in the Data Warehouse Center . 211

Exporting metadata	211
Tag language and XML files	212
Importing metadata	212
Publishing warehouse metadata	214
MQSeries	214
MQSeries data	214
Creating views for MQSeries messages	215
Importing MQSeries messages and XML metadata.	216
MQSeries user-defined program	217
Error return codes for the MQXMLXF user-defined program	218
Error logging for MQSeries	218
Importing ERwin metadata	219
IBM ERwin Metadata Extract Program	219
Starting the IBM ERwin Metadata Extract program	219
Creating tag language files for the IBM ERwin Metadata Extract program	220
Merging ERwin metadata with existing database data	221
Returning the ER1 file to a writable state	221
Importing ERwin metadata into the Data Warehouse Center	221
Changing a DB2 database definition to a source in the Data Warehouse Center	222
IBM ERwin metadata extract program errors	223

Chapter 15. Extending the Data Warehouse Center . 225

User-defined programs	225
What is a program group?	225
What is a user-defined program?	225
Using a user-defined program in the Data Warehouse Center	226
Agent sites for user-defined programs	226
Parameters for user-defined programs	227
Writing your own program for use with the Data Warehouse Center	227
Microsoft OLE DB and Data Transaction Services support	234

Creating views for OLE DB table functions.	235
Creating views for DTS packages	236

Chapter 16. Creating a star schema from within the Data Warehouse Center . . . 237

Star schemas	237
Designing the warehouse schema in the Data Warehouse Center	238
Warehouse schemas	239
Adding tables and views to the warehouse schema	239
Exporting a warehouse schema to the DB2 OLAP Integration Server	240
Working with a warehouse schema in the DB2 OLAP Integration Server.	240
Completing the design for your multidimensional cube	241
Creating an outline in the DB2 OLAP Integration Server.	242
Creating a load script to load data into the multidimensional cube in the DB2 OLAP Integration Server	243
Creating a batch file to load the command script for the DB2 OLAP Integration Server.	244
Exporting the metadata to the Data Warehouse Center	244
Loading the multidimensional cube from the Data Warehouse Center.	246
Data Warehouse Center objects that are created and associated with the target tables in the warehouse when you export metadata from the DB2 OLAP Integration Server.	246
Creating a schedule to populate the multidimensional cube	247
About publishing metadata about your warehouse schema	247

Chapter 17. Maintaining the warehouse database 249

Reorganizing data	249
Defining values for DB2 UDB REORG or DB2 for z/OS REORG utilities	249
Defining values for a DB2 for z/OS Utility.	249
Parameters for a DB2 for z/OS Utility	250
Updating system catalog statistics	254

Defining values for a DB2 UDB RUNSTATS utility	254	Running warehouse agents as a user process (Windows NT, Windows 2000, Windows XP)	268
Defining values for a DB2 for z/OS RUNSTATS utility	254	Running a Data Warehouse Center component trace	269
Chapter 18. Managing the control database	257	Error logging for warehouse programs and transformers	269
Backing up data	257	Tracing errors created by the Apply program	270
Stopping the Data Warehouse Center services (Windows NT, Windows 2000, Windows XP)	257	Start error trace files	271
Backing up the warehouse control database	257	Appendix A. Metadata mappings	273
Expanding your warehouse	258	Metadata mappings between the DB2 OLAP Integration Server and the Data Warehouse Center	273
Adding or deleting administrative interfaces and warehouse agents in the Data Warehouse Center	258	Mapping ERwin object attributes to Data Warehouse Center tags	274
Initializing a warehouse database	258	Trillium DDL to Data Warehouse Center metadata mapping	276
Changing the active warehouse control database	258	Metadata mappings between the Data Warehouse Center and CWM XML objects and properties	276
Initializing a warehouse control database during installation	259	Appendix B. Using Classic Connect with the Data Warehouse Center.	279
Data Warehouse Center configuration	260	What is Classic Connect?	279
Data Warehouse Center configuration	260	What does it do?	279
Chapter 19. Refreshing an OLAP Server database	261	Which data sources does it access?	279
Loading data into the OLAP Server database from the Data Warehouse Center.	261	How is it used?	280
Running calculations on OLAP Server data from the Data Warehouse Center.	262	What are its components?	280
Loading data from a flat file into an OLAP Server database	263	Requirements for setting up integration between Classic Connect and the Data Warehouse Center	288
Updating an OLAP Server outline from the Data Warehouse Center	263	Hardware and software requirements	288
Chapter 20. Data Warehouse Center logging and trace data	265	Installing and configuring prerequisite products	288
The basic logging function	265	Configuring communications protocols between z/OS, Windows NT, Windows 2000 and Windows XP	289
Warehouse log files	265	Communications protocols between z/OS and Windows NT, Windows 2000, or Windows XP	289
Viewing run-time errors using the basic logging function	265	Configuring the TCP/IP communications protocol	290
Viewing build-time errors using the basic logging function	266	Configuring a Windows NT, Windows 2000, Windows XP client	299
Viewing log entries in the Data Warehouse Center	266	Installing the Classic Connect Drivers component (Windows NT, Windows 2000, Windows XP)	299
Data Warehouse Center component trace data	266	Installing the CROSS ACCESS ODBC driver	299
Component trace data	266		

Configuring data sources	299	Defining values for the DWC 7.2 Clean Data transformer	321
Setting up TCP/IP communications.	302	DWC 7.2 Clean Data transformer	321
Setting up LU 6.2 communications	304	Rules tables for DWC 7.2 Clean Data transformer	322
Appendix C. Defining values for warehouse deprecated programs and transformers	311	Defining a DWC 7.2 Clean Data transformer	324
Defining values for Visual Warehouse 5.2 compatible programs.	311	Appendix D. Data Warehouse Center environment structure	325
Defining values for a Visual Warehouse 5.2 DB2 UDB Data Export (VWPEXPT1) program	311	Data Warehouse Center environment variables	325
Defining values for a Visual Warehouse 5.2 DB2 UDB Data Load Insert (VWPLOADI) program	312	Data Warehouse Center registry updates	325
Defining values for a Visual Warehouse 5.2 DB2 UDB Data Load Replace (VWPLOADR) program.	314	Data Warehouse Center configuration files	326
File type modifiers	316	Notices	327
Defining values for a Visual Warehouse 5.2 DB2 UDB REORG (VWPREORG) program	317	Trademarks	330
Defining values for a Visual Warehouse 5.2 DB2 UDB RUNSTATS (VWPSTATS) program	318	Index	333
Defining values for a Visual Warehouse 5.2 Load flat file into DB2 Enterprise Extended Edition (VWPLDPR) program (AIX)	319	Contacting IBM	339
		Product information	339

About this book

This book describes the steps that are required to use the IBM® 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 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

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 is data warehousing?

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.

About data warehousing

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, creating 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.
- Defining 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.

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.

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, file, or predefined nickname), SAP R/3 source, 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. Like a warehouse source, users can use warehouse targets to provide data to other warehouse targets. A central warehouse can provide data to departmental servers, or a main fact table in the warehouse can provide data to summary tables.

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.

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.

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.

About data warehousing

Processes and steps

A process contains a series of steps that perform 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. You can schedule a step to run one time only, or you can schedule it to run repeatedly, such as every Friday. You can also schedule steps to run in sequence, so that when one step finishes running, the next step begins

running. You can schedule steps to run upon completion, either successful or not, 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

There are 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

There are several types of program steps: DB2 for iSeries™ programs, DB2 for z/OS™ programs, DB2 for UDB programs, Visual Warehouse™ 5.2 DB2 programs, OLAP Server programs, File programs, and Replication. These steps run predefined programs and utilities.

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.

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 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.

About data warehousing

Connectors

The following connectors are included with DB2 Warehouse Manger, but are purchased separately. These connectors can help you to extract data and metadata from e-business repositories.

- DB2 Warehouse Manager Connector for SAP R/3
- DB2 Warehouse Manager Connector for the Web

With the Connector for SAP R/3, you can add the extracted data to a data warehouse, transform it using the DB2 Data Warehouse Center, or analyze it using DB2 tools or other vendors' tools. With the Connector for the Web, you can bring "clickstream" data from IBM® WebSphere Site Analyzer into a data warehouse.

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 the warehouse that they reflect. Because the Data Warehouse Center is easy to install and use, you can make changes 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.

Building a data warehouse

This topic describes the tasks involved in building a data warehouse.

Procedure:

To build a warehouse:

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

Starting the Data Warehouse Center

To start the Data Warehouse Center, you need to start the components of the Data Warehouse Center. 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.

Procedure:

To start the Data Warehouse Center:

Setting up your warehouse

1. Start the warehouse server and logger. This step is necessary only if the warehouse server and logger services are set to start manually, or if you stop the DB2 server.
2. Start the warehouse agent daemon. This step is required only if you are using an iSeries or zSeries warehouse agent, or if you are using a remote Windows NT or Windows 2000 warehouse agent that must be started manually.
3. Start the Data Warehouse Center administrative interface.
4. Define an agent site to the Data Warehouse Center.
5. Define security for the Data Warehouse Center.

Starting and stopping the Data Warehouse Center server and logger

Starting and stopping the warehouse server and logger (Windows NT, Windows 2000, Windows XP)

The warehouse server and the warehouse logger run as services on Windows NT, Windows 2000, or Windows XP. To start the services, you must restart the system after you initialize the warehouse control database. Then the warehouse server and logger will automatically start every time you start Windows NT, Windows 2000, or Windows XP 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. You must stop and restart the warehouse server and logger after you stop and restart the DB2 server to restore connections.

Procedure:

To manually start the warehouse server and logger on Windows NT, Windows 2000, Windows XP:

1. From the Windows NT, Windows 2000, or Windows XP 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 above, then click **Stop**. This will stop both the warehouse server and warehouse logger.

Starting and stopping the warehouse server and logger (AIX)

On AIX, the root user can manually start or stop the warehouse server (iwh2serv) and logger (iwh2log) daemons using the shell script db2vwsvr. This can be done if for any reason the iwh2serv and iwh2log daemons are not

running. When starting the daemons, the command adds an entry in the `inittab` file with identifier `db2vwsvr`. As a result, this command will be restarted automatically if the warehouse server daemon stops.

Procedure:

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, the `iwh2serv` and `iwhlog` processes are restarted and the entry in the `inittab` with identifier `db2vwsvr` is updated. When you stop the daemons, the command also removes the `inittab` entry. As a result, the daemons will not be restarted until the command is run again with the `start` option.

Note: The `db2vwsvr` first runs the `IWH.environment` script to initialize the environment. It will assume that the script is 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` is changed, the daemons should be restarted using the command "`db2vwsvr start`"

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

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

Procedure:

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

```
ps -ef | grep iwh2serv | grep -v grep and ps -ef | grep iwh2log | grep -v grep
```

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

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`.

Setting up your warehouse

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.

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.

Starting the warehouse agent daemon (Windows NT, Windows 2000, or Windows XP)

If you installed a Windows NT, Windows 2000, or Windows XP warehouse agent other than the local agent, you can manually start the warehouse agent daemon just as you start any Windows NT, Windows 2000, or Windows XP 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 NT, Windows 2000, or Windows XP.

Procedure:

To manually start the Windows NT, Windows 2000, or Windows XP 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.

Setting up your warehouse

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:

Port name	Port number
<code>vwkernel</code>	<code>11000/tcp</code>
<code>vwd</code>	<code>11001/tcp</code>
<code>vwlogger</code>	<code>11002/tcp</code>

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.

Procedure:

To start the warehouse agent daemon in the foreground:

1. Telnet to 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 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:

Setting up your warehouse

```
$ ps -ef | grep vwd
MVSUSR2      198  16777537  - 13:13:22 tty0013  0:00 grep vwd
MVSUSR2  16777446  16777538  - 09:57:21 tty0002  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 tty0013  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 NT, Windows 2000, Windows XP)

Stop the warehouse agent daemon when you change the environment variables for the warehouse agent and warehouse agent daemon on Windows NT, Windows 2000, or Windows XP.

Procedure:

To stop the Windows NT, Windows 2000, or Windows XP 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.

Note: The IWH.environment will also be referenced by the AIX warehouse server.

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`

Setting up your warehouse

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 on the command line:
 - `ps -ef | grep vwd`
 - `D OMVS,A=ALL`
2. Stop the daemon process by entering the following command on the command line:
`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+v on 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.

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.

Agent site configurations

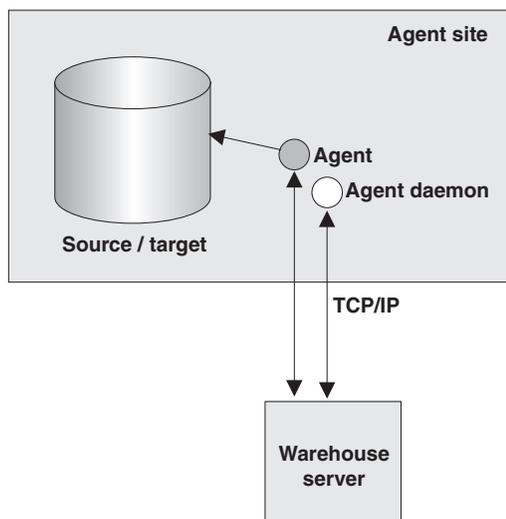
The warehouse agent receives SQL commands from the warehouse server and then passes the commands to the source or target databases.

You can set up the warehouse agent in one of three 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.

Setting up your warehouse



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

In this configuration, the warehouse agent passes SQL statements that extract data from the source tables. The warehouse agent transforms the data, if necessary, and then writes it to the target table.

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

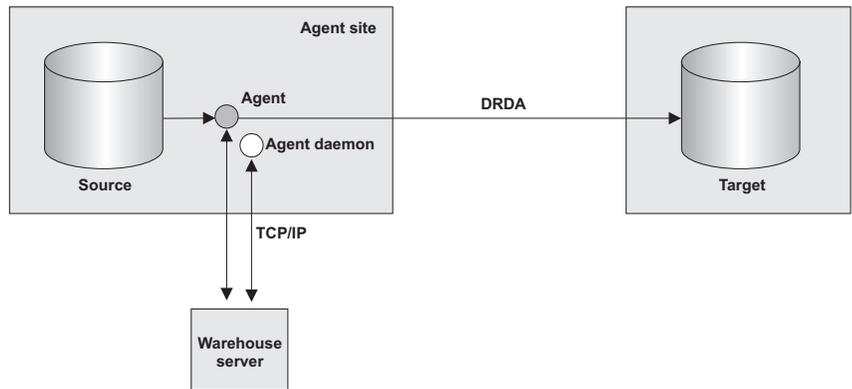
The warehouse agent is installed on the system that contains the warehouse target table.

In this configuration, the warehouse agent passes SQL statements that extract data from a remote source. The warehouse agent transforms the data, if necessary, and then writes the data to the target table on the local database.

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.



In this configuration, the warehouse agent passes SQL that extracts data from the warehouse source on the local database. The warehouse agent transforms the data, if necessary, and then writes the data to the target table on the remote database.

After you set up access to your data and you determine the location of your warehouse agent, you must define security for your warehouse.

Updating your environment variables on z/OS

You might need to update your environment variables.

Procedure:

To update your environment variables, add the following variables to the .profile file in the home directory of the user who starts the agent daemon:

Variable

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

Add the following variables to your ODBC initialization file if you want to receive ODBC (CLI) traces:

Variable

```
APPLTRACE
```

Setting up your warehouse

APPLTRACEFILENAME

DIAGTRACE

TRACEFILENAME

Data Warehouse Center security

Because the Data Warehouse Center stores user IDs and passwords for various databases and systems, there is a Data Warehouse Center security structure that 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 the users the authorization perform functions. Warehouse users and warehouse groups do not need to match the DB users and DB 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*.

Requirement: 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 that corresponds to someone who uses the Data Warehouse Center. You might then define a warehouse group that is authorized to access certain warehouse sources, and add the new user to the new warehouse group. The new user is authorized to access the warehouse sources that are included in the 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.

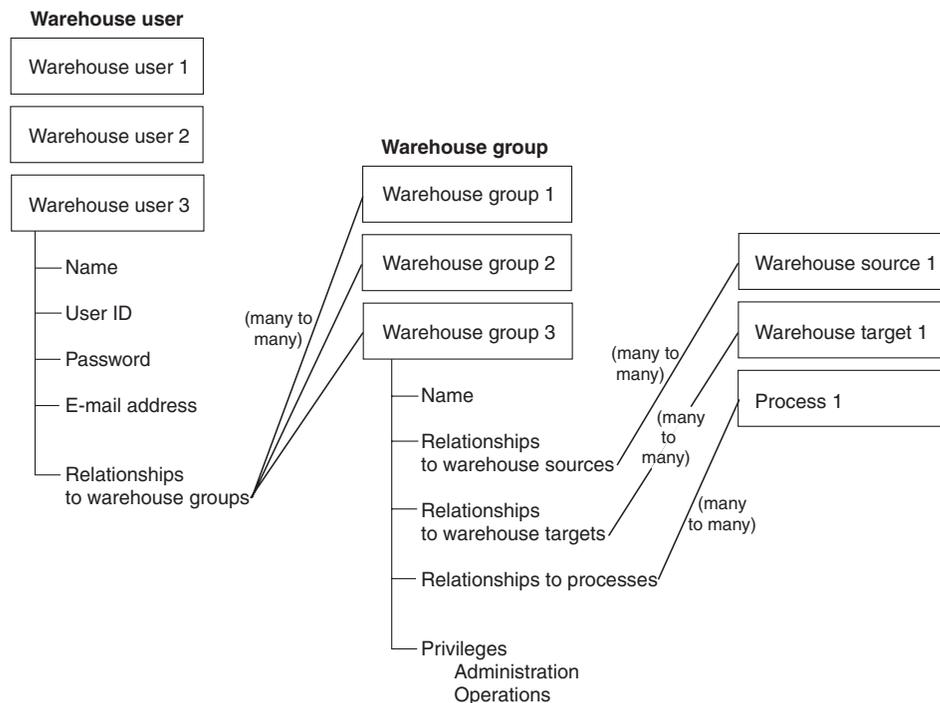
The list of tables or views that users can access from a source will be limited by their group membership as well, so that they will be able to choose from among the tables and views to which they have access. Further, the set of actions available to the user through the Data Warehouse Center will be

Setting up your warehouse

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:



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.

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.

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 - 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 - 7.2
DB2 Universal Database for iSeries	3.7 - 5.1
DB2 Universal Database for z/OS	5.1 or later
DB2 for VM	5.3.4 or later
DB2 for VSE	8.1

Setting up DB2 warehouse sources

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

Warehouse agent support for DB2 sources

The following warehouse agents support DB2 sources.

- Windows NT, Windows 2000, or Windows XP 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 NT, Windows 2000, Windows XP)

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

Prerequisites:

Database access program

None

Source/agent connection

ODBC

Client enabler program

DB2 Universal Database Version 8 server or DB2 client

Procedure:

To set up connectivity for DB2 Universal Databases:

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 NT, Windows 2000, Windows XP)

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

Prerequisites:

Database access program

None

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 has 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:

Database access programs

None

Source/agent connection

ODBC

Client enabler program

DB2 Universal Database Version 8 server or DB2 client

Restrictions:

Setting up DB2 warehouse sources

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_36.so
```

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

```
[SAMPLE]
Driver=/usr/opt/db2_08_01/lib/libdb2_36.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 has 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:

Database access programs

None

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/libdb2_36.so
```

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

```
[SAMPLE]
Driver=/usr/opt/db2_08_01/lib/libdb2_36.so
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 has 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:

Database access programs

None

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/IBMdb2/V8.1/lib/libdb2_36.so.1
```

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

Setting up DB2 warehouse sources

```
[SAMPLE]
Driver=/opt/IBM/db2/V8.1/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 has 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:

Database access programs

None

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/IBMdb2/V8.1/lib/libdb2_36.so
#Driver=/opt/IBMdb2/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/IBMdb2/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.
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.

Prerequisites:

Database access programs

None

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 has 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:

Database access programs

None

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.

Setting up DB2 warehouse sources

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 sections describe how to set up access to source databases.

Setting up access to DB2 Universal Database data sources

You can use any DB2 Universal Database database as a source database for your warehouse.

Procedure:

To set up access to DB2 Universal Database source databases:

1. Define privileges to the source databases.
2. Establish connectivity to the source databases.

Defining privileges to DB2 Universal Database data sources

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

Procedure:

To define privileges to DB2 Universal Database sources:

1. Ask the system administrator of the source system to set up a user ID with the following privileges:
 - 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 Universal Database source, you must establish connectivity to it.

Establishing connectivity to DB2 Universal Database 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.
2. If the database is remote, catalog the node.
3. Catalog the database.
4. Register the database as an ODBC system DSN 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. 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.
5. 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:

Use a gateway site to access data from the 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.

Setting up DB2 warehouse sources

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 NT, Windows 2000, Windows XP)

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. For Windows NT, Windows 2000, or Windows XP install Microsoft SNA Server.
2. Install DB2 Connect.
3. Configure your 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 procedure in this topic can be used for establishing connectivity in either case.

Prerequisites:

Setting up DB2 warehouse sources

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:

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.

Tip: On Windows NT, Windows 2000, or Windows XP, you can use the DB2 UDB Configuration Assistant to complete this task.

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.

Tip: 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).

Connecting to local and remote databases from the iSeries warehouse agent

You must catalog the names of local and remote database that you plan to use as a warehouse source or target 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 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 add a database name entry to the iSeries Relational Database directory, enter the following command at an iSeries command prompt:

```
ADDRDBDIRE databasename locationname
```

where *databasename* is the name of your iSeries database, and *locationname* is 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.

Attention: 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.

Setting up DB2 warehouse sources

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

Fred 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 describes how Fred should catalog each database on each workstation.

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, adding, changing, and removing 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 Relational Connect is used

The zSeries agent allows you to access IMS™ and VSAM remote databases through the Classic Connect ODBC driver. The Classic Connect ODBC driver cannot be used to access DB2 Universal Database databases directly.

Requirements for accessing DB2 Relational Connect data sources with the zSeries warehouse agent

When you use the zSeries™ agent to access DB2® Relational Connect and other DB2 Relational Connect data sources, Systems Network Architecture (SNA) LU 6.2 is required as the communications protocol. Although DB2 Relational Connect supports TCP/IP as an application requester, it does not have an application server. Also, DB2 Relational Connect can not be used as a target for z/OS, because DB2 Relational Connect does 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

Setting up DB2 warehouse sources

- 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 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, underscores, and spaces
- Can contain a double quotation mark, represented by two consecutive quotation marks: ""

Importing 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.

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.

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.

Restrictions:

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:

Setting up DB2 warehouse sources

- 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.

On iSeries Version 4 Release 2 and Version 4 Release 3, you must select the View folder to import system tables.

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” in the *Help: Data Warehouse Center*

Setting up DB2 warehouse sources

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 decided 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.

Source database	Windows NT, Windows 2000, Windows XP agent	AIX, Solaris Operating Environment, Linux agents
Informix (with and without client)	9.x	9.x
Oracle	8.1.7	8.1.7
Microsoft [®] SQL Server	7.0	7.0
Microsoft Excel	97	Not supported
Microsoft Access	97	Not supported
Sybase Adaptive Server (without client)	12.0	11.9.2 - 12.0
IMS	5.1	5.1

The Data Warehouse Center includes several DataDirect Technologies, Inc. ODBC drivers for accessing non-DB2 data.

Warehouse agent support for non-DB2 sources

The following table shows which warehouse agents support which sources.

Setting up non-DB2 warehouse sources

Data Source	Windows NT, Windows 2000, Windows XP warehouse agent	AIX or Solaris Operating Environment warehouse agent	Linux warehouse agent	iSeries warehouse agent	zSeries warehouse agent
DB2 Relational Connect	✓	✓			✓
Oracle	✓	✓	✓		
Sybase Adaptive Server	✓	✓	✓		
Microsoft SQL Server	✓	✓	✓		
Informix	✓	✓	✓		
Generic ODBC driver	✓	✓	✓		
IMS	✓ ¹	✓ ¹	✓ ¹		✓ ^{1 2}
VSAM	✓ ¹	✓ ¹	✓ ¹		✓ ^{1 2}

1. To access IMS or VSAM data when DB2 Relational Connect is on a remote workstation, catalog the node on which DB2 Relational Connect resides, and catalog the DB2 Relational Connect database on the agent site.
2. To access IMS or VSAM data with the zSeries agent, use the Classic Connect ODBC driver on z/OS. These sources can also be accessed via DataJoiner and DB2 Relational Connect.

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.

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

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 NT, Windows 2000, Windows XP)

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

Prerequisites:

Programs

None.

Database access program

None

Source/agent connection

ODBC

Client enabler program

DataDirect Version 3.7 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 your source.

Related tasks:

- “Configuring the ODBC driver for a Sybase Adaptive Server source – without client (Windows NT, Windows 2000, Windows XP)” on page 62

Setting up connectivity for an Oracle source (Windows NT, Windows 2000, Windows XP)

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

Prerequisites:

Database access program

None

Source/agent connection

ODBC

Client enabler program

Setting up non-DB2 warehouse sources

- Oracle SQL*Net V2
- DataDirect Version 3.7 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. Install SQL*NET on the agent site.
2. Catalog the remote database according to client enabler instructions.
3. Identify the user ID and password with access to the source database.

Setting up connectivity for an Informix 9.2 source – with client (Windows NT, Windows 2000, Windows XP)

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:

Database access program

None

Source/agent connection

ODBC

Client enabler program

- For Informix 9.x, i-connect 9.x
- DataDirect Version 3.7 Driver Manager and Informix driver

Database client requirements

None

Procedure:

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

Related tasks:

- “Configuring the Informix 9.2 server – without client (Windows NT, Windows 2000, Windows XP)” on page 62
- “Setting up connectivity for an Informix 9.2 source – without client (Windows, NT, Windows 2000, Windows XP)” on page 47

Setting up connectivity for an Informix 9.2 source – without client (Windows, NT, Windows 2000, Windows XP)

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:

Database access program

None

Source/agent connection

ODBC

Client enabler program

DataDirect Version 3.7 Driver Manager and Informix driver

Database client requirements

None

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 NT, Windows 2000, Windows XP)

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

Prerequisites:

Database access program

None

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 3.7 Driver Manager and Microsoft SQL Server driver

Setting up non-DB2 warehouse sources

Procedure:

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

1. Install the Microsoft SQL Server DB-Library and Net-Library on the agent site.
2. Catalog the remote database according to client enabler instructions.
3. Identify the user ID and password that has access to the source database.

Setting up connectivity for a Microsoft Access source (Windows NT, Windows 2000, Windows XP)

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

Prerequisites:

Database access program

None

Source/agent connection

ODBC

Client enabler program

None

Procedure:

To set up connectivity for a Microsoft Access source, use the generic ODBC connect string. See the Microsoft Access help topics for a mapping of the ANSI SQL data types supported by Microsoft Access.

Setting up connectivity for a Microsoft Excel data source (Windows NT, Windows 2000, Windows XP)

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

Prerequisites:

Database access program

None

Source/agent connection

ODBC

Client enabler program

None

Procedure:

Setting up non-DB2 warehouse sources

To set up connectivity for a Microsoft Excel data source, use the generic ODBC connect string. See the Microsoft Excel help topics for a mapping of the ANSI SQL data types supported by Microsoft Excel.

Setting up connectivity for an IMS, or a VSAM data source (Windows NT, Windows 2000, Windows XP)

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 Relational Connect to establish connectivity.

Prerequisites:

Database access programs

If you are using the zSeries warehouse agent, use Classic Connect as your database access program. If you are not using the zSeries warehouse agent, use one of the following combinations of programs:

- The CROSS ACCESS ODBC driver and DB2 Relational Connect Classic Connect
- DB2 Relational Connect and DB2 Relational Connect Classic Connect

Source/agent connection

- If you are using the CROSS ACCESS ODBC driver, use ODBC as your database access program.
- If you are using DB2 Relational Connect, use TCP/IP or APPC.

Client enabler program

None

Procedure:

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. Install and configure the CROSS ACCESS ODBC driver on the agent site.
4. 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 DB2 Relational Connect workstation using DB2 Relational Connect:

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.

Setting up non-DB2 warehouse sources

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

1. Catalog the node on which DB2 Relational Connect resides.
2. Catalog the DB2 Relational Connect database.

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 (without client) source (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:

Programs

None.

Database access program

None

Source/agent connection

ODBC

Client enabler program

DataDirect Version 3.7 Driver Manager and Sybase driver

Database client requirements

None.

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

```
[SAMPLE]
Driver=/usr/opt/db2_08_01/odbc/lib/ibase16.so
Description=Sybase 11 ODBC Database
Database=sample
ServerName=sybase1192
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 (AIX)

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

Setting up non-DB2 warehouse sources

Prerequisites:

Database access programs

None

Source/agent connection

ODBC

Client enabler program

DataDirect Version 3.7 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:

```
[SAMPLE]
Driver=/usr/opt/db2_08_01/odbc/lib/ibor816.so
Description=Oracle 8 Database
Database=SAMPLE
UID=xxxx
PWD=xxxx
ServerName=ora816
```

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 – with client (AIX)

When you set up a data warehouse to use an Informix source, you must set up connectivity to it. You are not required to use an Informix client when you access an Informix 9.2 source. This topic describes how to set up connectivity for an Informix 9.2 data source when you are using an Informix client.

Prerequisites:

Database access programs

None

Source/agent connection

ODBC

Client enabler program

DataDirect Version 3.7 Driver Manager and Informix driver

Database client requirements

Informix-Connect and ESQL/C version 9.1.4 or later

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

```
[SAMPLE]
Driver=/usr/opt/db2_08_01/odbc/lib/ibinf16.so
Description=Informix Database
Database=SAMPLE
HostName=yourHostName.stl.ibm.com
PortNumber=1648
LoginID=xxxx
Password=xxxx
ServerName=inf92
Service=ifmx92
```

Procedure:

To set up connectivity for an Informix data source:

1. Install the version of the AIX warehouse agent with ODBC access.
2. Identify the user ID and password that has 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:

Database access programs

None

Source/agent connection

ODBC

Client enabler program

DataDirect Version 3.7 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:

```
[SAMPLE]
Driver=/usr/opt/db2_08_01/odbc/lib/ibinf16.so
Description=Informix Database
Database=SAMPLE
HostName=yourHostName.stl.ibm.com
LoginID=xxxx
Password=xxxx
ServerName=inf92
Service=ifmx92
```

Setting up non-DB2 warehouse sources

Setting up connectivity for a Microsoft SQL Server source (AIX)

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

Prerequisites:

Database access programs

None

Source/agent connection

ODBC

Client enabler program

DataDirect Version 3.7 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=xxyyy.zzz.ibm.com
AnsiNPW=yes
Database=test7
UID=xxxx
PWD=xxxx
QuotedID=no
TDS=4.2
UseProcForPrepare=1
```

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 has access to the source database.

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

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

Setting up connectivity for a Sybase Adaptive Server source – without client (Solaris Operating Environment, Linux)

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

Prerequisites:

Database access programs

None

Source/agent connection

ODBC

Client enabler program

DataDirect Version 3.7 Driver Manager and Sybase driver

Database client requirements

None.

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

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

```
[SAMPLE]
Driver=/opt/IBM/db2/V8.1/odbc/lib/ibase16.so
Description=Sybase database
Database=SAMPLE
ServerName=sybase1192
NetworkAddresss=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 (without client) data source, install and configure the version of the Solaris Operating Environment or Linux warehouse agent with ODBC access.

Setting up connectivity for an Oracle 8 data source (Solaris Operating Environment, Linux)

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

Prerequisites:

Database access programs

None

Source/agent connection

ODBC

Database client requirements

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

Setting up non-DB2 warehouse sources

There are two versions of the Solaris Operating Environment or Linux warehouse agents: one for ODBC access and one for CLI access. You must use the ODBC version of the warehouse agent for non-DB2 sources. Verify the version of the warehouse agent that you have installed before you begin this procedure.

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

```
[SAMPLE]
Driver=/opt/IBM/db2/V8.1/odbc/lib/ibor816.so
Description=Oracle 8 Database
Database=SAMPLE
UID=xxxx
PWD=xxxx
ServerName=ora816
```

Procedure:

To set up connectivity for an Oracle data source:

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

Related tasks:

- “Verifying communication between the warehouse server and the warehouse agent” on page 14
- “Verifying the warehouse agent (AIX, Solaris Operating Environment, Linux)” on page 50

Setting up connectivity for an Informix 9.2 source – with client (Solaris Operating Environment, Linux)

The Informix client is optional for Informix 9.2. If you use a client for an Informix 9.2 source, follow this procedure for setting up connectivity.

Prerequisites:

Database access programs

None

Source/agent connection

ODBC

Client enabler program

DataDirect Version 3.7 Driver Manager and Informix driver

Database client requirements

Informix-Connect and ESQ/C version 9.2 or later.

Setting up non-DB2 warehouse sources

Verify that you are using the ODBC version of the Solaris Operating Environment or Linux warehouse agent.

Procedure:

To set up connectivity for an Informix source:

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

The following example shows a sample `odbc.ini` file entry:

```
[SAMPLE]
Driver=/opt/IBM/db2/V8.1/odbc/lib/ibinf16.so
Description=Informix 9.2
Database=vwdwc
HostName=yourHostName.stl.ibm.com
LoginID=xxxx
Password=xxxx
ServerName=inf92
Service=ifmx92
```

Setting up connectivity for an Informix 9.2 source – without client (Solaris Operating Environment)

The Informix client is optional for Informix 9.2. If you are not using a client for an Informix 9.2 source, follow this procedure for setting up connectivity.

Prerequisites:

Database access programs

None

Source/agent connection

ODBC

Client enabler program

DataDirect Version 3.7 Driver Manager and Informix driver

Verify that you are using the ODBC version of the Solaris Operating Environment or Linux warehouse agent.

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 the Solaris Operating Environment and Linux:

Setting up non-DB2 warehouse sources

```
[SAMPLE]
Driver=/opt/IBM/db2/V8.1/odbc/lib/ibifcl16.so
Description=Informix 9.2
Database=vwdwc
HostName=yourHostName.st1.ibm.com
PortNumber=1648
LoginID=xxxx
Password=xxxx
ServerName=inf92
Service=ifmx92
```

Note: PortNumber=1648 is required for the clientless configuration.

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:

Database access programs

None

Source/agent connection

ODBC

Client enabler program

DataDirect Version 3.7 Driver Manager ODBC 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:

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 has 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=xxyyy.zzz.ibm.com
AnsiNPW=yes
Database=test7
UID=xxxxL
```

```
PWD=xxxx  
QuotedID=no  
TDS=4.2  
UseProcForPrepare=1
```

Configuring non-DB2 warehouse sources

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

Configuring an Informix 9.2 client (Windows NT, Windows 2000, Windows XP)

The Informix client is not required for accessing an Informix 9.2 data source. You can connect directly to the source through ODBC.

Prerequisites:

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

Procedure:

To configure an Informix client:

1. Configure the Informix server.
2. Configure the Informix host.
3. Configure the ODBC driver.

Related tasks:

- “Configuring an Informix 9.2 server – with client (Windows NT, Windows 2000, Windows XP)” on page 59
- “Configuring a host for an Informix 9.2 source — with client (Windows NT, Windows 2000, Windows XP)” on page 60
- “Configuring the ODBC driver for Informix 9.2 – without client (Windows NT, Windows 2000, Windows XP)” on page 61

Configuring an Informix 9.2 server – with client (Windows NT, Windows 2000, Windows XP)

If you use an Informix client, you must configure the Informix server and host information by using the Informix-Setnet 32 utility before you can access Informix sources. This topic describes how to configure the Informix server. If you are not using a client, you are only required to configure the ODBC driver that is required to access Informix.

Setting up non-DB2 warehouse sources

Prerequisites:

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

Procedure:

To configure an Informix server:

1. Click **Start** → **Programs** → **Informix Client** → **Setnet32** to open the Setnet32 window.
2. Click the **Server Information** tab.
3. Select an existing Informix database server from the **Informix Server** list, or type the name of a new database server in the **Informix Server** field.
4. Select the host with the database server that you want to use from the **HostName** list, or type the name of a new host computer in the **HostName** field.
5. Select the required network protocol from the **Protocolname** list.
6. In the **Service Name** field, type the service name or the port number that is assigned to the database server on the host computer.
The service name must be defined in the services file in the Windows NT installation directory on the client workstation.
7. Click the **Environment** tab.
8. Type INFORMIXSERVER in the **Variable Name** field.
9. Click **Set**.
10. Click **OK**.
11. Configure the Informix host.

Configuring a host for an Informix 9.2 source — with client (Windows NT, Windows 2000, Windows XP)

If you use an Informix client, you must configure the Informix server and host information by using the Informix-Setnet 32 utility before you can access Informix sources. This topic describes how to configure the Informix host. If you are not using a client, you are only required to configure the ODBC driver that is required to access an Informix source.

Prerequisites:

- Install the ODBC driver that is required to access an Informix database. If you do not have the ODBC driver that is required to access an Informix database, you can get the driver from the DB2 Universal Database CD-ROM by using the Custom installation choice.
- Configure the Informix server.

Procedure:

To configure the Informix host information:

1. Click the **Host Information** tab in the Setnet32 window.
2. In the **Current Host** list, select the name of the host computer that you want to use to establish a network connection or type a name in the field to define a new host name.
The user name entered must be the user name for an account on the selected host computer.
3. Type your password in the **Password** field.
4. Click **OK**.
5. Configure the ODBC driver.

Configuring the ODBC driver for Informix 9.2 – without client (Windows NT, Windows 2000, Windows XP)

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 an Informix database. If you do not have the ODBC driver that is required to access an Informix database, you can get the driver from the DB2 Universal Database CD-ROM 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.
7. Click the **General** tab.
8. Type the database alias in the **Data Source Name** field.
9. Type a description of the database in the **Database Description** field.
10. Type the name of the database in the **Database Name** field.
11. Click the **Connection** tab.
12. Type the user ID in the **Default User Name** field.

Setting up non-DB2 warehouse sources

13. Type the name of the server in the **Host Name** field.
14. Type the service name in the **Service Name** field.
15. Select **onsoctcp** from the **Protocol Type** list.
16. Click **OK**.
17. In the **System Data Sources** window, select the database alias that you want to use.
18. Click **OK**.
19. Close the ODBC windows.

Configuring the Informix 9.2 server – without client (Windows NT, Windows 2000, Windows XP)

You are not required to configure the Informix 9.2 client to access a source. You can access the data directly using only the ODBC driver.

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.
7. Click the **General** tab.
8. Type the database alias in the **Data Source Name** field.
9. Type a description of the database in the **Description** field.
10. Type the name of the server in the **Host Name** field.
11. Type the port number from the Informix sqlhosts file in the **Port Number** field.
12. Type the name of the server in the **Server Name** field.
13. Type the name of the database in the **Database Name** field.
14. Type your user ID in the **User Name** field.
15. Click **OK**.
16. Close the ODBC windows.

Configuring the ODBC driver for a Sybase Adaptive Server source – without client (Windows NT, Windows 2000, Windows XP)

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

Prerequisites:

The ODBC driver that is required to access a Sybase database must be installed. If you do not have the ODBC driver that is required to access a Sybase database, you can get the driver from the DB2 Universal Database CD-ROM using the Custom installation choice.

After the ODBC driver is installed, you must set up access to your Sybase database by registering the database as a system database source name (DSN) in ODBC.

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. Optional: To enable delimited identifier support for Sybase, select the **Enable Quoted Identifiers** check box in the Advanced page of the ODBC Driver Setup notebook.
7. Click **Finish**.
A driver setup window opens.
8. Click the **General** tab.
9. Type the database alias in the **Data Source Name** field.
10. Type a description of the database in the **Description** field.
11. Select **Winsock** from the **Network Library Name** list.
12. Type the network address in the **Network Address** field. The network address is the server name and port number from the Sybase interfaces file.
13. Type the name of the database in the **Database Name** field.
14. Click the **Connection** tab.
15. Type the logon ID in the **Default Logon ID** field.
16. Type the ID of the workstation in the **Workstation ID** field.
17. Click **OK**.
18. Select the database alias that you want to use from the System Data Sources window.
19. Click **OK**.
20. Close the ODBC windows.

Setting up non-DB2 warehouse sources

Configuring an Oracle 8 client (Windows NT, Windows 2000, Windows XP)

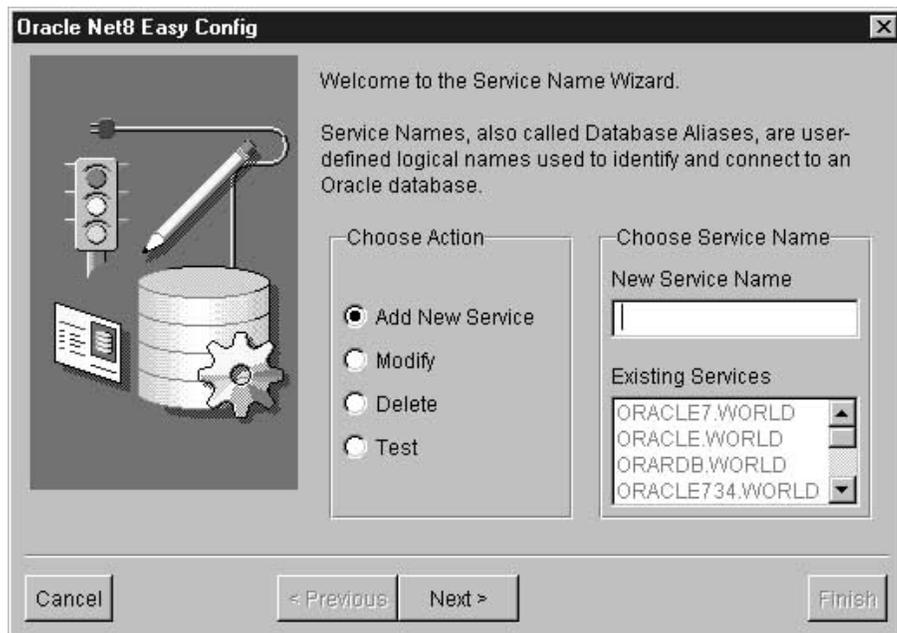
Before you can access an Oracle 8 source, you must configure the Oracle 8 client.

Procedure:

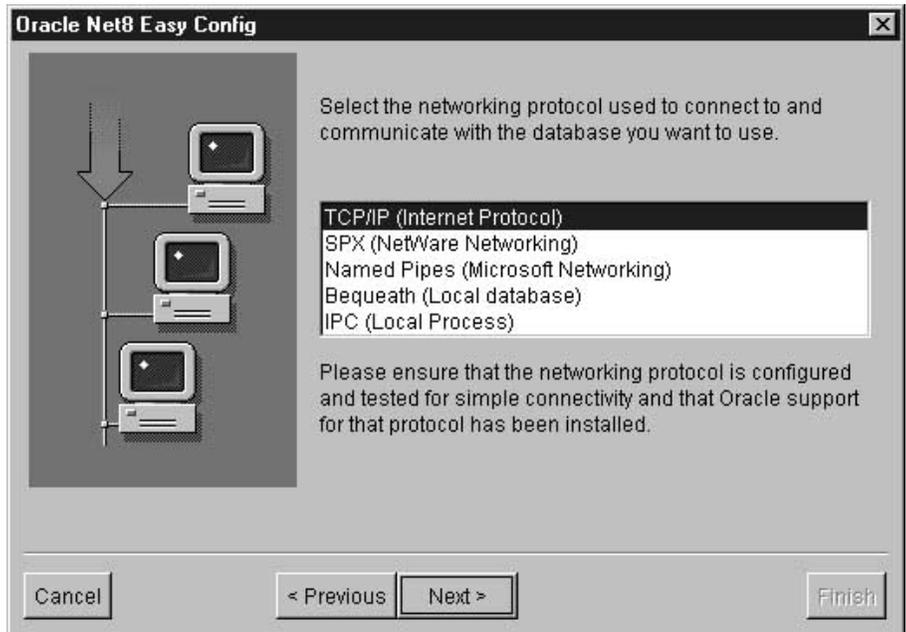
To configure an Oracle 8 client:

1. Click **Start** → **Programs** → **Oracle for Windows NT** → **Oracle Net8 Easy Configuration**.

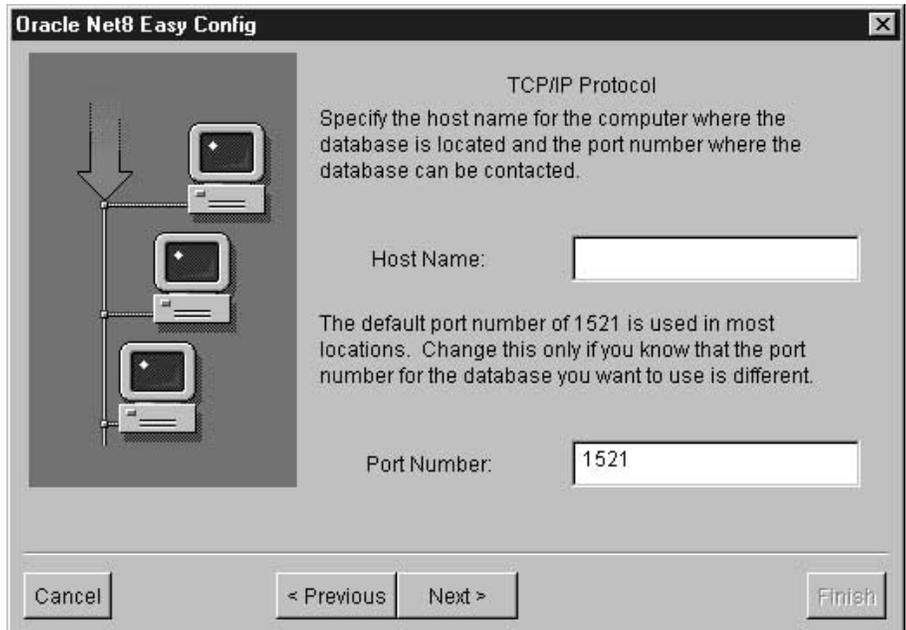
The Oracle Net8 Easy Configuration window opens and displays several choices for working with your client configurations.



2. Click the radio button next to the client configuration choice you want, and add a new client configuration, change, or view an existing configuration. If you click **Add**, you must also type a database alias name in the **New Service Name** field.
3. Click **Next**.
4. Select the type of networking protocol that you want from the list in the Protocol window.

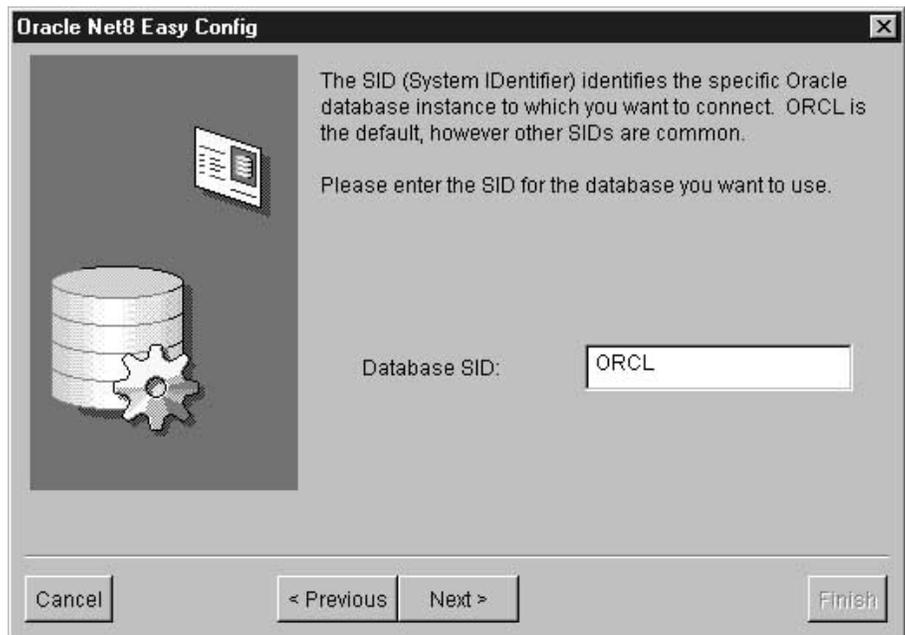


5. Click **Next**.
6. Type the TCP/IP host name in the **Host Name** field in the TCP/IP Protocol window.



Setting up non-DB2 warehouse sources

7. Type the TCP/IP port number in the **Port Number** field.
8. Click **Next**.
9. Type the database SID in the **Database SID** field.



10. Click **Next**.
The Test window opens.
11. Optional: Click **Test** to test the configuration.
12. Click **Next**.
13. Click **Done**.
A confirmation window opens.
14. Review the information in the confirmation window.
15. Click **Finish** if the information in the confirmation window is correct.
If the information in the confirmation window is not correct:
 - a. Click **Previous** until you reach the window with the incorrect information.
 - b. Correct the information.
 - c. Repeat steps 8 through 15.

After you configure the Oracle 8 client, you can configure the Oracle ODBC driver.

Configuring the ODBC driver for an Oracle 8 client (Windows NT, Windows 2000, Windows XP)

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 an Oracle database. If you do not have the ODBC driver that is required to access an Oracle database, you can get the driver from the DB2 Universal Database CD-ROM 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.

7. Select the **General** tab.
8. Type the database alias in the **Data Source Name** field.
9. Type the name of the database in the **Database Name** field.
10. Type the name of the server in the **Server Name** field.
11. Click **OK**.
12. Select the desired database alias from the System Data Sources window.
13. Click **OK**.
14. Close the ODBC windows.

Testing an Oracle 8 client configuration (Windows NT, Windows 2000, Windows XP)

When you configure an Oracle 8 client, you can test the configuration.

Procedure:

To test an Oracle 8 client configuration:

1. From the Test window in the Oracle Net8 Easy Config notebook, click **Test**.
2. Type the user ID in the **User ID** field.

Setting up non-DB2 warehouse sources

3. Type the password in the **Password** field.
4. Click **Test**.

Configuring a Microsoft SQL Server client (Windows NT, Windows 2000, Windows XP)

To set up access to the Microsoft SQL Server client, you must configure the Microsoft SQL Server client software using the Microsoft SQL Server Client Network Utility.

Procedure:

To configure a Microsoft SQL Server client:

1. Click **Start** → **Programs** → **Microsoft SQL Server for Windows NT** → **Client Net Utility**.

The SQL Server Client Network Utility window opens.

2. Click **TCP/IP** from the **Default network library** list.
3. Specify a server by either selecting the server name that you want from the **Server alias configurations** list or by adding a new server to the list.

To add a new server to the list:

- a. Click **Add** to add a new server to the list. The Add Network Library Configuration window opens.
 - b. Type the name of the server in the **Server alias** field.
 - c. Type the name of the server in the **Computer name** field.
This name must be the same name that you typed in the **Server alias** field.
 - d. Type the port number in the **Port number** field.
 - e. Click **OK** to close the Add Network Library Configuration window.
 - f. Select the server name that you just created from the **Server alias configurations** list.
4. Click **OK**.

Configuring the ODBC driver for a Microsoft SQL Server (Windows NT, Windows 2000, Windows XP)

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

Prerequisites:

Install the ODBC driver that is required to access a Microsoft SQL Server database. If you do not have the ODBC driver that is required to access a

Setting up non-DB2 warehouse sources

Microsoft SQL Server database, you can get the driver from the DB2 Universal Database CD-ROM using the Custom installation choice.

Procedure:

To register a DSN for an ODBC driver:

1. If you use the Data Warehouse Center ODBC drivers to access Microsoft SQL Server, verify that the system path contains the `sqllib\odbc32` directory.
2. Click **Start** → **Settings** → **Control Panel**.
3. Double-click **ODBC Data Sources**.
4. Click the **System DSN** tab.
5. Click **Add**.
6. Select the ODBC driver that you want to register.
7. Click **Finish**.
The ODBC SQL Server Driver Setup window opens.
8. Click the **General** tab.
9. Type the database alias name in the **Data Source Name** field.
10. Type a description of the database in the **Description** field.
11. Type the name of the server that you want to use in the **Server Name** field.
12. Type the name of the database that you want to use in the **Database Name** field.
13. Click the **Advanced** tab.
14. Type the user name in the **Default Logon** field.
15. Type the workstation ID in the **Workstation ID** field.
16. To enable delimited identifier support for Microsoft SQL Server, select the **Enable Quoted Identifiers** check box.
17. Click **OK**.
18. Select the database alias that you want to use from the System Data Sources window.
19. Click **OK**.
20. Close the ODBC windows.

Configuring a Microsoft Access client (Windows NT, Windows 2000, Windows XP)

Before you can access a Microsoft Access source, you must configure the Microsoft Access client.

Procedure:

Setting up non-DB2 warehouse sources

To configure the Microsoft Access client:

1. Create and catalog a Microsoft Access database.
2. Create and catalog a target warehouse database.
3. Define a warehouse that uses the databases.

Related tasks:

- “Cataloging a Microsoft Access source database in ODBC (Windows NT, Windows 2000, Windows XP)” on page 73
- “Cataloging a target warehouse database for use with a Microsoft Access source database (Windows NT, Windows 2000, Windows XP)” on page 74

Creating a Microsoft Access source database (Windows NT, Windows 2000, Windows XP)

The first step in accessing a Microsoft Access source database is to create a Microsoft Access database.

Procedure:

To create a Microsoft Access database:

1. Open Microsoft Access.
2. Click **Database Wizard**.
3. Click **OK**.
The New window opens and displays database templates.
4. Select a template.
5. Click **OK**.
6. Type the name of the database in the **File name** field.
7. Click **Create**.

The Database Wizard opens.

8. Follow the prompts, and click **Finish** to create your database.
Note the path and file name of the database, because you will use it later.
9. Create tables and enter data into the tables.

After you create the Microsoft Access database, catalog the database in ODBC.

Creating a target warehouse database (Windows NT, Windows 2000, Windows XP)

This topic explains how to create a target warehouse database.

Procedure:

To create a target warehouse database:

Setting up non-DB2 warehouse sources

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.
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.

Defining a warehouse that uses the Microsoft Access and warehouse target databases (Windows NT, Windows 2000, Windows XP)

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

Prerequisites:

Before beginning this procedure, you must create and catalog a Microsoft Access database and a warehouse target database.

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. Type any additional parameters in the **ODBC Connect String** field.
 - 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.

Setting up non-DB2 warehouse sources

5. Create a step that uses one or more source tables from the warehouse source for the Microsoft Access database, and 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.

Importing table definitions from a Microsoft Access database (Windows NT, Windows 2000, Windows XP)

This topic describes how to import table definitions from a Microsoft Access database on Windows NT, Windows 2000, or Windows XP.

Prerequisites:

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

Restrictions:

The Memo field of a Microsoft Access database is represented in a Data Warehouse Center source as a data type of LONG VARCHAR with column size exceeding 1 GB. To support practical system configurations, the Data Warehouse Center truncates the values exceeding 128 KB. To avoid truncating Memo field values in the warehouse source, 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. If you do not change the data type of the column, any values larger than 128 KB will be truncated.

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 Version 6. The CLOB data type is supported for iSeries beginning with Version 4, Release 4 with DB FixPak 4 or later (PTF SF99104). For iSeries, the install disk Version 4, Release 4 dated February 1999 also contains support for the CLOB data type.

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 NT, Windows 2000, Windows XP)

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

Prerequisites:

Create a Microsoft Access database.

Procedure:

To catalog a Microsoft Access database in ODBC:

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

Setting up non-DB2 warehouse sources

17. Click **OK** in the ODBC Microsoft Access Setup window.
18. Click **Close**.

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

Related tasks:

- “Setting up connectivity for a Microsoft Access source (Windows NT, Windows 2000, Windows XP)” on page 48
- “Configuring a Microsoft Access client (Windows NT, Windows 2000, Windows XP)” on page 69
- “Cataloging a target warehouse database for use with a Microsoft Access source database (Windows NT, Windows 2000, Windows XP)” on page 74

Cataloging a target warehouse database for use with a Microsoft Access source database (Windows NT, Windows 2000, Windows XP)

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

Prerequisites:

You must create a target database before you can complete this procedure.

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. Click **IBM DB2 ODBC 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 **Close**.

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

Using a Microsoft Excel spreadsheet as a warehouse source (Windows NT, Windows 2000, Windows XP)

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 NT, Windows 2000, Windows XP)

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:

- You must have a Microsoft Excel spreadsheet to catalog.
- 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. Select **Microsoft Excel Driver** from the **Installed ODBC Drivers** list.
6. Click **OK**.
7. Type the spreadsheet alias in the **Data Source Name** field.
8. Optional: Type a description of the spreadsheet in the **Description** field.
9. Select **Excel 97** from the **Version** list.
10. Click **Select Workbook**.

Setting up non-DB2 warehouse sources

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**.

Creating named tables for Microsoft Excel data sources (Windows NT, Windows 2000, Windows XP)

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 have selected in step 1. If not, click the icon on the far right of the **Refers to** field to include all the cells that you selected.
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.

Creating a target warehouse database for use with a Microsoft Excel data source (Windows NT, Windows 2000, Windows XP)

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. From 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.

Cataloging a target warehouse database for use with a Microsoft Excel data source (Windows NT, Windows 2000, Windows XP)

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. Select **IBM DB2 ODBC 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 **Close**.

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

Defining sources and targets to the Data Warehouse Center that use a Microsoft Excel data source (Windows NT, Windows 2000, Windows XP)

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

Procedure:

To present multiple steps:

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 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.

Setting up non-DB2 warehouse sources

- b. Click **OK** on the Filter window to list all the tables in your spreadsheet.
Verify that the tables that you created in the database are in the **Available Tables** list.
- c. Select the table that you want to use from the **Available Tables** list.
- 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.

Accessing an IMS or VSAM source database from the Data Warehouse Center

You can use Classic Connect and the CROSS ACCESS ODBC driver to access data in an IMS or VSAM database.

Procedure:

To access data in an IMS or VSAM database:

1. Use Classic Connect to map the nonrelational data into a pseudorelational format.
2. Use the CROSS ACCESS ODBC driver to access the pseudorelational data.
3. Define an IMS or VSAM warehouse source in the Data Warehouse Center that corresponds to the pseudorelational data.

Related tasks:

- “Setting up connectivity for an IMS, or a VSAM data source (Windows NT, Windows 2000, Windows XP)” on page 49

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

This section describes how to set up the Informix, Sybase Adaptive Server, Oracle, and Microsoft SQL Server sources on AIX, Linux, or the Solaris Operating Environment.

Configuring an Informix 9.2 client (AIX, Solaris Operating Environment)

You are not required to configure a client in order to access an Informix 9.2 data source. You can access an Informix 9.2 data source directly using the ODBC driver.

Procedure:

To configure the Informix client:

1. Open the `sqlhosts` file in a text editor.
2. Add a new entry listing to the file.
To keep the same format, copy and paste the example listing. Then, change the entry information as described in the following steps.
3. Type the database name.
4. Type the protocol type `onsoctcp`.
5. Type the host name.
6. Type the port number.

The port number must be the same as the port number defined in the `/etc/services` file in the Windows installation directory on the client computer.

After you configure the Informix client, configure the ODBC driver.

SQL Hosts file

The following example shows an SQL Hosts file with a new entry listing.

```
# Informix V9
database1  oonsoctcp  test0     ifmxfrst1
database2  oonsoctcp  test0     ifmxfrst2
```

Configuring the ODBC driver for accessing an Informix 9.2 database (AIX, Solaris Operating Environment)

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

Prerequisites:

Setting up non-DB2 warehouse sources

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

Procedure:

To set up access to your Informix database:

1. Open the `.odbc.ini` file in a text editor.
2. Define your Informix database alias at the top of the file.
3. Add a new entry to the file.
4. Type the directory location of the ODBC driver.
5. Type a description of the database.
6. Type the database name.
7. Type the server host name.
8. Type the port number.
9. Type the login ID.
10. Type the password for the login ID.
11. Type the server name.
12. Type the service name.
13. Save and close the file.

Configuring the ODBC driver for the Sybase Adaptive Server – without client (AIX, Linux, Solaris Operating Environment)

To use a Sybase Adaptive Server – without client source as a warehouse source, you must configure the ODBC driver for the Sybase Adaptive Server. No client is necessary for accessing a Sybase database source from the Data Warehouse Center. ODBC drivers are used to register the source, target, and warehouse control databases that the Data Warehouse Center will access.

Prerequisites:

If you do not have the ODBC driver that is required to access a Sybase database, you can get the driver from the DB2 Universal Database CD-ROM using the Custom installation choice.

Procedure:

To configure an ODBC driver Sybase Adaptive Server:

1. Open the `.odbc.ini` file in a text editor.
2. Define your Sybase database alias at the top of the file.
3. Add a new entry to the file.

4. Type the directory location of the ODBC driver.
5. Type a description of the database.
6. Define the database type as **master**.
7. Type the server name.
8. Enter the network address. The network address is the server name and port number from the Sybase interfaces file.
9. Type the logon ID.
10. Type the password for the logon ID.
11. Type the directory location of the interfaces file.
12. Optional: To enable delimited identifier support, edit the Sybase data source to include the connect attribute `EQI=1`.
13. Save and close the file.

Configuring an Oracle 8 client (AIX, Linux, Solaris Operating Environment)

Before you can access an Oracle 8 source database from the Data Warehouse Center, you must configure the Oracle 8 client.

Procedure:

To configure an Oracle 8 client:

1. Open the `tnsnames.ora` file in a text editor.
2. Add a new entry listing to the file.
To keep the same format, copy and paste the example listing. Then, change the entry information as described in the following steps.
3. Type the protocol you want to use.
4. Type the server host name.
5. Type the port name.
6. Type the SID.

After you configure the Oracle client, configure the ODBC driver.

Example of an entry for a `tnsnames.ora` file (AIX, Linux, Solaris Operating Environment)

This is a sample entry for a `tnsnames.ora` file.

```
# Oracle 8.1.5
Oracle8i=
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS =
        (PROTOCOL = TCP)
        (HOST = superman)
```

Setting up non-DB2 warehouse sources

```
        (PORT = 2000)
    )
)
(CONNECT_DATA =
  (SID=oracle8i)
)
)
```

Configuring the ODBC driver for an Oracle 8 database (AIX, Linux, Solaris Operating Environment)

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

Prerequisites:

If you do not have the ODBC driver that is required to access a Sybase database, you can get the driver from the DB2 Universal Database CD-ROM using the Custom installation choice.

Procedure:

To configure the ODBC driver for an Oracle database:

1. Open the .odbc.ini file in a text editor.
2. Define your Oracle database alias at the top of the file.
3. Add a new entry to the file.
4. Type the directory location of the ODBC driver.
5. Type the server name.
6. Type a description of the database.
7. Save and close the file.

Configuring a Microsoft SQL Server client (AIX, Linux, Solaris Operating Environment)

To use a Microsoft SQL Server source as a warehouse source, you must configure the Microsoft SQL Server Client.

Procedure:

To configure your Microsoft SQL Server client on an AIX, Linux, or Solaris Operating Environment, configure TCP/IP on the AIX, Linux, or Solaris Operating Environment client and the Windows server where Microsoft SQL Server is installed. No other actions are required to configure your client because the Microsoft SQL Server client configuration is integrated with the ODBC driver configuration in the INI file.

After you configure the Microsoft SQL Server client, configure the ODBC driver.

Configuring the ODBC driver for accessing a Microsoft SQL Server database (AIX, Linux, Solaris Operating Environment)

After you configure the Microsoft SQL Server client, you can configure the ODBC driver. ODBC drivers are used to register the source, target, and warehouse control databases that the Data Warehouse Center will access.

Prerequisites:

If you do not have the ODBC driver that is required to access a Microsoft SQL Server database, you can get the driver from the DB2 Universal Database CD-ROM using the Custom installation choice.

Procedure:

To configure the ODBC driver:

1. Open the `.odbc.ini` file in a text editor.
2. Define your Microsoft SQL Server database alias at the top of the file.
3. Add a new entry to the file.
4. Type the network address of the server running Microsoft SQL Server.
5. Type your preference for ANSI-defined behaviors the driver should use.
6. Type the database name.
7. Type the user ID.
8. Type the password for the user ID.
9. Type your preference for recognizing quotation marks in SQL statements.
10. Type the version of TDS that is used with your database.
11. Save and close the file.

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.

Setting up non-DB2 warehouse sources

Restrictions:

If an .odbc data source is created using Relational Connect and the Create Nickname statement, the data source will not be available for importing tables in 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.

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:

- “Specifying database information for a non-DB2 warehouse source in the Data Warehouse Center” on page 84
- “ Defining a warehouse source based on relational databases : Data Warehouse Center help” in the *Help: Data Warehouse Center*

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.
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, and their columns, that you want to use in your source.

Defining warehouse sources for use with DB2 Relational Connect

DB2 Relational Connect provides several advantages for accessing data for steps. Instead of using ODBC support for non-IBM databases, you can use DB2 Relational Connect to access those databases directly using the native database protocols. You can also use DB2 Relational Connect to write to an Oracle database or other non-IBM databases. With DB2 Relational Connect, 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. DB2 Relational Connect optimizes the SQL statement to enhance performance.

Prerequisites:

- You should have a basic understanding of warehouse components before you attempt this task.
- You should be familiar with creating server mappings and nicknames in DB2 Relational Connect.
- Before you define the warehouse sources, you must map each source database to a DB2 Relational Connect database through DB2 Relational Connect's server mapping.
- You must create nicknames for each data source table that you want to use with the Data Warehouse Center.

Setting up non-DB2 warehouse sources

Restrictions:

The Data Warehouse Center transformers are not supported with a DB2 Relational Connect target database.

Procedure:

To define Data Warehouse Center steps that take advantage of DB2 Relational Connect's benefits, first you define warehouses that use DB2 Relational Connect databases. Then, you define steps that write to those warehouses.

Define a separate warehouse source definition in the Data Warehouse Center for each DB2 Relational Connect source database.

You might also need to create a user mapping that maps the DB2 Relational Connect 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 DB2 Relational Connect database.

Server mappings and nickname tables for warehouse sources accessed by DB2 Relational Connect

The following example shows how to create a server mapping and create a nickname for a table:

```
create wrapper sqlnet options (DB2_FENCED 'Y')
create server oracle1 type oracle version 8.1.7 wrapper sqlnet
authorization iwhserve 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 command registers the sqlnet wrapper for the Oracle data source.

The CREATE SERVER command (extended across multiple lines for readability) defines a source database called Oracle 1, where:

oracle1 The name that identifies the remote database in DB2[®] Relational Connect.

oranode

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

Oracle The type of database.

8.1.7 The version of the database.

sqlnet The method to use to access the database, such as NET8 for Oracle or dblib or ctlib for Sybase.

The DBCNAME option of the create server command is not needed because Oracle allows only one database per node. For some other data sources, you can specify a database.

The create user command specifies the user ID that DB2 Relational Connect will use to connect to the remote database (Oracle). The keyword USER is a DB2 special register that specifies the currently logged-on user. The user connects to the remote Oracle database by using the specified user ID and password (iwhserve and VWPW).

The create nickname command specifies a nickname, *iwh.oracle_target*, for a remote Oracle table, *ORACLE1.iwhserve.oratar*. After you create the nickname in DB2 Relational Connect, you can substitute the nickname for the name of the remote table. For example, if you specify `SELECT * FROM iwh.oracle_target`, DB2 Relational Connect 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 DB2 Relational Connect nicknames as table definitions. In this example, you would import *iwh.oracle_target* from DB2 Relational Connect.

Defining source tables for DB2 Relational Connect 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 DB2 Relational Connect nicknames as table definitions.

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.

Data Source	Windows NT, Windows 2000 or Windows XP 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. Copy File using FTP is used to copy files on the agent site to or from a remote host.

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

This section explains how to set up connectivity for Windows NT, Windows 2000, and Windows XP file sources.

Setting up warehouse file sources

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)

Client enabler program

None

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:

Database access program

None

Source/agent connection

ODBC

Client enabler program

DataDirect v3.7 ODBC driver manager and text file driver.

Restriction:

The zSeries warehouse agent cannot connect to file sources. Instead they are accessed 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 NT, Windows 2000, Windows XP)

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[®] NT, Windows 2000, or Windows XP file server, be aware of 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  
NET USE drive: //hostname/sharedDrive password /USER:userid
```

where:

- *drive* is the drive letter that represents the shared drive on the agent site
- *hostname* is the TCP/IP hostname of the remote workstation
- *sharedDrive* is the drive on the remote workstation that contains the file
- *password* is the password that is required to access the shared drive
- *userid* is the user ID that is required to access the shared drive

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

When you define the agent site, specify the user ID and password that are used to access the file.

When you define the warehouse source for the file, 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.

You can also define a similar .bat file to delete the link to the remote drive after the Data Warehouse Center processes the files. If 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:

Database access program

None

Setting up warehouse file sources

Source/agent connection

ODBC

Client enabler program

DataDirect v3.7 ODBC driver manager and text file driver.

Restriction:

The zSeries warehouse agent cannot connect to file sources. Instead they are accessed 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)

These sections 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. Instead they are accessed 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 v3.7 ODBC driver manager and text file driver.

Restriction:

The zSeries warehouse agent cannot connect to file sources. Instead they are read through standard commands.

Procedure:

To set up connectivity for 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 (AIX)

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

Prerequisites:

Database access programs

None

Source/agent connection

TCP/IP

Client enabler program

DataDirect v3.7 ODBC driver manager and text file driver.

Restriction:

The zSeries warehouse agent cannot connect to file sources. Instead they are accessed through standard open, read, and close commands.

Procedure:

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

Related tasks:

- “Accessing data files with FTP” on page 96

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:

Database access programs

None

Source/agent connection

TCP/IP

Client enabler program

DataDirect v3.7 ODBC driver manager and text file driver.

Restriction:

The zSeries warehouse agent cannot connect to file sources. Instead they are accessed through standard open, read, and close commands.

Procedure:

Setting up warehouse file sources

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 sections 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. The z/OS agent cannot connect to file sources. Instead they are accessed 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 v3.7 ODBC driver manager and text file driver.

Restriction:

The zSeries warehouse agent cannot connect to file sources. Instead they are accessed through standard open, read, and close commands.

Procedure:

To set up connectivity for 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 (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:

Database access programs

None

Source/agent connection

TCP/IP

Client enabler program

DataDirect Version 3.7 ODBC driver manager and text file driver.

Restriction:

The zSeries warehouse agent cannot connect to file sources. Instead, they are accessed 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:

Database access programs

None

Source/agent connection

TCP/IP

Client enabler program

DataDirect v3.7 ODBC driver manager and text file driver.

Restriction:

The zSeries warehouse agent cannot connect to file sources. Instead, they are accessed 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.

Setting up warehouse file sources

```
[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
```

Network File System (NFS) 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 NFS command (which is NFS LINK if you use Maestro from Hummingbird). 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
```

Accessing data files with FTP

You can use FTP to access data files on a remote workstation. In general, use FTP if the file size is 20 megabytes or less. When you promote a step that uses remote files to test mode, the files will be transferred just as they would be if the step was promoted to production mode. If the file is large, promotion can take a long time, and the agent site can run out of space.

One way to avoid this problem is to place a dummy file on the remote workstation during testing. Another way is to use the Copy file using FTP warehouse program instead of FTP.

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, change the account information in the JCL template for your z/OS system.

Procedure:

To access a data file using 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.

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. Use the Copy file using FTP warehouse program if the file size is larger than 20 megabytes. The Data Warehouse Center does not run warehouse programs when a step is promoted to test status, so the file will not be transferred. You can also specify the location of the target file for the Copy file using FTP warehouse program.

Procedure:

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

1. Declare the file with a warehouse source type of Local File.
2. Define two steps to access a file of this size. Define the first step to use the Copy file using FTP warehouse program. Use this step to copy the file to the agent site. Define the second step to use the warehouse source that you create for the file. The step will access the file as a local file. This file is the output file of the first step.

Related tasks:

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

Setting up warehouse file sources

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:

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” in the *Help: Data Warehouse Center*

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
- DB2 Enterprise Server Edition
- Warehouse databases that are accessed by DB2 Relational Connect

Setting up a DB2 Universal Database warehouse

This section explains how to set up a DB2 Universal Database warehouse.

Defining privileges for DB2 Universal Database warehouses

To set up a warehouse, you must have a database on the target system and a user ID with the required privileges.

Prerequisites:

Create a database on the target system.

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

Setting up access to a warehouse

- 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.

Setting up a DB2 for iSeries warehouse

You can use DB2 Connect or Client Access/400 to access a DB2 for iSeries warehouse.

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.

After you have 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

After you define the required privileges and set up a DB2 Connect gateway site, 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.

Setting up access to a warehouse

Checking the current level of your iSeries operating system

When you are setting up Client Access/400 for access to a DB2 for iSeries database, you must have the correct level of the iSeries operating system.

Procedure:

To check the current level of your iSeries operating system:

1. Type the following command:
GO LICPGM
2. Select Option 10.
3. Press F11.

Verifying that all of the servers in an iSeries system are running

All of the servers in an iSeries system must be running when you set up CA/400 for access to a DB2 for iSeries database.

Procedure:

To verify that all of the servers are running, enter the following command from a DOS command prompt on the Windows NT, Windows 2000, or Windows XP workstation:

```
cwbping hostname ip
```

You will receive a response like this:

```
C:\>cwbping demo400 ip
```

```
IBM iSeries Client Access for Windows NT  
Version 3 Release 1 Level 2
```

```
(C) Copyright IBM Corporation and Others 1984, 1995.  
All rights reserved.
```

```
U.S. Government Users Restricted Rights - Use, duplication or  
disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
```

```
Licensed Materials - Property of IBM
```

```
[9.180.160.17]
```

```
pinging server Port Mapper successful  
pinging server as-central successful  
pinging server as-database successful  
pinging server as-dtaq successful  
pinging server as-file successful  
pinging server as-netprt successful  
pinging server as-rmtcmd successful  
pinging server as-signon successful
```

If the servers are not started, enter the following command on the iSeries system to start the servers:

```
STRHOSTSVR SERVER (*ALL)
```

Setting up access to a DB2 for iSeries warehouse with iSeries Client Access/400

You can use Client Access/400 to access a DB2 for iSeries warehouse.

Procedure:

To set up Client Access/400 for access to a DB2 for iSeries database:

1. Install the latest Client Access PTFs on the iSeries system.
To see a full list of the latest informational APARs that will tell you which PTFs are needed for your iSeries system, visit the following Web page:
<http://www.ibm.com/servers/eserver/iseries/clientaccess/>
2. In the navigation frame, click **Service Packs**.
On the Client Access Service Packs web page, you can find information about the latest workstation service packs and host PTFs.
3. Check the current level of your operating system.
4. Ensure that the iSeries QUSER password is set (not expiring). You can set the password to no password, if required.
5. Enter the TCP/IP host name for the iSeries system and the Windows NT, Windows 2000, or Windows XP workstation in the WINNT\SYSTEM32\DRIVERS\ETC\HOSTS file. For example:
9.123.456.7 demo400
9.123.456.8 mypc
6. Install Client Access/400 on the Windows NT, Windows 2000, or Windows XP workstation.
7. Verify that all the servers on the iSeries system are running.
8. Verify that the iSeries system has a defined DB2 database name. You can check this entry by issuing the Work with Relational Database Directory Entries (WRKRDBDIRE) command on the iSeries system.
If the iSeries system does not have a DB2 database name, you will receive an SQL0114 error when you try to access the database.
9. Set the DB2CODEPAGE environment variable to 850 in the Windows NT, Windows 2000, or Windows XP system variables (not user variables).
10. Download the latest Client Access/400 service pack from the iSeries Client Access Web site.
<http://www.ibm.com/servers/eserver/iseries/clientaccess/service.htm>
11. Use the Client Access/400 ODBC driver to register the iSeries database as a system DSN. Add other libraries as required, for example, DORMERK,QGPL,MYLIBRARY
12. Test the connection to the iSeries database with Lotus Approach® or Microsoft Query.

Setting up access to a warehouse

If you receive Client Access/400 errors, click **Help** for a detailed explanation of the error.

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.

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 tailor steps to work with target tables in DB2 for z/OS.

Procedure:

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, 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.

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 tablespace in which to create the table. If you do not specify a tablespace, DB2 for z/OS creates the table in the default DB2 database 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. The GRANT command syntax that the Data Warehouse Center creates is not supported by DB2 for VM or DB2 for VSE.
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.

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. If the table was created by Visual Warehouse Version 2.1, you must delete the table and create it again.

Procedure:

Setting up access to a warehouse

To create the Data Warehouse Center status table, enter the following DB2 command:

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

You might need to change the statement as follows:

- If your target warehouse is in DB2 on a Windows NT, Windows 2000, or Windows XP system, DB2 for iSeries, or DB2 on a UNIX-based system, use the statement above as shown.
- 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 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 is used to determine in which partition a particular row of data is stored.

Setting up a DB2 Enterprise Server Edition warehouse

By default, the Data Warehouse Center creates warehouse target tables in a format that is supported by the DB2 Universal Database products. You can tailor steps to work with target tables in DB2 Enterprise Server Edition.

Defining privileges to the DB2 Enterprise Server Edition database

Before you can establish connectivity to a DB2 Enterprise Server Edition warehouse, you must define privileges to the DB2 Enterprise Server Edition database.

Prerequisites:

Create a database on the target system.

Procedure:

To define privileges to the DB2 Enterprise Server Edition database:

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 DB2 Enterprise Server Edition database, you can establish connectivity.

Defining the DB2 Enterprise Server Edition database to the Data Warehouse Center

After setting up access to the DB2 Enterprise Server Edition warehouse, you need to define the database.

Prerequisites:

- Define privileges to the DB2 Enterprise Server Edition warehouse.
- Establish connectivity to the DB2 Enterprise Server Edition warehouse.

Procedure:

To define the DB2 Enterprise Server Edition warehouse to the Data Warehouse Center:

Setting up access to a warehouse

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 DB2 Enterprise Server Edition database.

Defining a warehouse for use with DB2 Relational Connect

DB2 Relational Connect provides several advantages for accessing data for steps. Instead of using ODBC support for non-DB2 databases, you can use DB2 Relational Connect to access those databases directly using the native database protocols. You can also use DB2 Relational Connect to write to non-DB2 databases. With DB2 Relational Connect, 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. DB2 Relational Connect optimizes the SQL statement to enhance performance.

You can define Data Warehouse Center steps that take advantage of DB2 Relational Connect's function. First, you define warehouses that use DB2 Relational Connect databases. 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 a SELECT INSERT statement because the DB2 Relational Connect database is both the source and target database. DB2 Relational Connect then optimizes the query for the DB2 Relational

Setting up access to a warehouse

Connect target databases (such as Oracle and Sybase). You can define steps with sources from more than one database by taking advantage of the DB2 Relational Connect heterogeneous join optimization.

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

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 DB2 Relational Connect database for each remote database, because the Data Warehouse Center requires that the name of the table be BVBESTATUS. One DB2 Relational Connect 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 in DB2 Relational Connect.

Prerequisites:

You should have a basic familiarity with:

- The basic components of a warehouse
- Creating server mappings in DB2 Relational Connect
- Creating nicknames in DB2 Relational Connect
- You must create a BVBESTATUS table in the DB2 Relational Connect database or in each remote database.
- If you create the BVBESTATUS table in the DB2 Relational Connect 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 DB2 Relational Connect:

Setting up access to a warehouse

1. Create a separate logical warehouse definition in the Data Warehouse Center for each DB2 Relational Connect target database. For example, define one warehouse for each Oracle database, another warehouse for each Sybase database, and so on.
2. Specify the same DB2 Relational Connect database as the database for each logical warehouse.

Creating target tables in a remote database

With DB2 Relational Connect, the Data Warehouse Center can create tables directly into a remote database, such as Oracle.

Procedure:

To create the target table:

1. Create a step with a target table in a DB2 Relational Connect 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.

If you are using DB2 Relational Connect with DB2 Version 7 on Windows NT or Windows 2000, or UNIX and FixPak 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.

Moving a target table from a DB2 Relational Connect database to a remote database

You can create and test a step in a DB2 Relational Connect database, and then move it to a remote database.

Procedure:

To create and test a step in a DB2 Relational Connect database and then move it to a remote database:

1. Create a step with a target table in a DB2 Relational Connect database.
2. Promote the step to test mode.
3. 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.
4. Manually move the table to a remote database such as Oracle, by creating the table in the Oracle database, and dropping the DB2 Relational Connect table. (You can also use a modeling or data dictionary tool.) The data types of the DB2 Relational Connect tables and the Oracle tables must be compatible.
5. Manually create a nickname for the remote table in DB2 Relational Connect. The nickname must match the name of the target table for the step in the Data Warehouse Center.
6. Run the step again to test that the data moves through DB2 Relational Connect to the target correctly.
7. 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 the warehouse tables or use existing warehouse tables.
2. Create a nickname for the target table in DB2 Relational Connect.
3. 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.
4. 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.
5. Promote the step to test mode.
6. Run the step to test access to the target tables.
7. 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.

When you are defining a warehouse target, keep the following information in mind:

- The option to create Data Warehouse Center transformers in the target database is not available for z/OS™ because z/OS does not support scripts.
- If the ODBC data source was created using Relational Connect and the Create Nickname statement, it will not be available for importing tables into the Data Warehouse Center. You must define a warehouse target in the Data Warehouse Center without importing the target tables, and then create the target tables manually.

However, if you define the data source as a Client Connect target nickname and 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.

- 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.

- The Data Warehouse Center supports target tables that use ordinary SQL identifiers. An ordinary identifier:
 - Must start with a letter.
 - Can include uppercase letters, number, 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.
- The Data Warehouse Center supports target resource tables that use delimited identifiers. A delimited identifier:

Setting up a warehouse target

- 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: "".
- Table names can be case sensitive, or have other restrictions, for some warehouse target types.
- 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.
 - 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.
- 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**.

Supported warehouse targets

The following table shows the versions and release levels of supported DB2[®] warehouse targets.

Target	Version/Release
DB2 Universal Database™ for Windows NT	6 - 8.1

Target	Version/Release
DB2 Universal Database Enterprise Server Edition	8.1
DB2 Universal Database Enterprise-Extended Edition	6 - 7.2
DB2 Universal Database for iSeries	3.7 - 5.1
DB2 Universal Database for AIX	6 - 8.1
DB2 Universal Database for Solaris Operating Environment	6 - 8.1
DB2 Universal Database for Linux	8.1
DB2 Universal Database for z/OS™	5 and later
DB2 Relational Connect	8.1
DB2 for VM	3.4 - 5.3.4
DB2 for VSE	3.2, 8.1

Note: 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:

The following restrictions apply when you are defining a warehouse target:

- If a data source is created using Relational Connect and the Create Nickname statement, it will not be available for importing tables in Data Warehouse Center. To use the data source as a source table, define a warehouse target in the Data Warehouse Center, but do not import a target table. You must 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 data source as a Client Connect target nickname and 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.
- On iSeries Version 4 Release 2 and Version 4 Release 3, you must use the View folder to import tables.

Procedure:

To define a warehouse target:

Setting up 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.

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.

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.

Defining and running processes

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.

Related tasks:

- “Steps and tasks : Data Warehouse Center help” in the *Help: Data Warehouse Center*

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 is a list of the main step types and the warehouse connectors.

SQL steps

There are two kinds of SQL steps in the Data Warehouse Center. 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 designed for specific DB2 Universal Database platforms

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.

DSNUTILS is a DB2[®] for z/OS[™] stored procedure that runs in a WLM and RRS environment. You can use DSNUTILS to run any installed DB2 utilities by using the user-defined stored procedure interface.

The definition of each warehouse program is added during initialization. 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.

There are two types of transformers:

- Statistical transformers
- Warehouse transformers

In addition to the transformer steps, there is another transformer that 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.

Restriction: The Data Warehouse Center transformers are not supported with a DB2 Relational Connect target database.

Defining and running processes

The definition of each transformer is added during initialization. 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 DB2 database or a DB2 for z/OS subsystem or data-sharing group) that are on the same machine or on different machines 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 the operational table is updated. With replication, you can use incremental updates to keep your data current.

SAP steps

Use an SAP Data Extract step to extract SAP business objects from an SAP R/3 system and import them into the Data Warehouse Center.

Web traffic steps

Use Web Traffic Polling to query the status of WebSphere® Site Analyzer data imports to determine if current Web traffic data has been copied from the data imports to the tables in the WebSphere Site Analyzer Webmart database.

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.

Related tasks:

- “Enabling warehouse transformers” in the *DB2 Warehouse Manager Installation Guide*
- “Running DB2 for z/OS utilities” in the *DB2 Warehouse Manager 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.

		Agent sites				
Name	Description	Windows NT, Windows 2000, Windows XP	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.	✓	✓	✓		✓

Defining and running processes

DB2 warehouse utilities

The following table lists the DB2 warehouse utilities. Step subtypes are organized by utility group. Utility groups are located on the left side of the Process Model window. A *program group* is a logical grouping of related utilities.

Name	Description	Agent sites				
		Windows NT, Windows 2000, Windows XP	AIX or Linux	Solaris Operating Environment	iSeries	z/OS
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 for iSeries load replace	Loads data from a delimited file into a DB2 for iSeries database, replacing existing data in the database with new data.				✓	
DB2 for iSeries load insert	Loads data from a delimited file into a DB2 for iSeries table, appending new data to the existing data in the database.				✓	
DB2 for z/OS load	Loads records into one or more tables in a table space.					✓
DB2 data export	Exports data from a local DB2 database to a delimited file.	✓	✓	✓		

Name	Description	Agent sites				
		Windows NT, Windows 2000, Windows XP	AIX or Linux	Solaris Operating Environment	iSeries	z/OS
DB2 runstats	Runs the DB2 RUNSTATS utility on the specified table.	✓	✓	✓		
DB2 for z/OS runstats						✓
DB2 reorg	Runs the DB2 REORG and RUNSTATS utilities on the specified table.	✓	✓	✓		✓
DB2 for z/OS reorg						✓

DB2 OLAP Server programs

The following table lists the DB2 OLAP Server programs. Step subtypes are organized by program group. Program groups are located on the left side of the Process Model window. A *program group* is a logical grouping of related programs.

Name	Description	Agent sites				
		Windows NT, Windows 2000, or Windows XP	AIX or Linux	Solaris Operating Environment	iSeries	z/OS
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.	✓	✓	✓	✓	

Defining and running processes

Name	Description	Agent sites				
		Windows NT, Windows 2000, or Windows XP	AIX or Linux	Solaris Operating Environment	iSeries	z/OS
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.	✓	✓	✓	✓	
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.	✓	✓	✓	✓	

Replication programs

The following table lists the Replication programs. Step subtypes are organized by program group. Program groups are located on the left side of the Process Model window. A *program group* is a logical grouping of related programs.

Name	Description	Agent sites				
		Windows NT, Windows 2000, or Windows XP	AIX or Linux	Solaris Operating Environment	iSeries	z/OS
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 exactly at the time of the copy.	✓	✓	✓	✓	✓

Defining and running processes

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. Program groups are located on the left side of the Process Model window. A *program group* is a logical grouping of related programs. 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.

Name	Description	Agent sites			
		Windows NT, Windows 2000, or Windows XP	AIX or Linux	Solaris Operating Environment iSeries	z/OS
DB2 load replace (VWPLOADR)	Loads data from a delimited file into a DB2 UDB 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 EEE (AIX only) (VWPLDPR)	Loads data from a delimited file into a DB2 EEE database, replacing existing data in the database with new data.		✓		
DB2 data export (VWPEXPT1)	Exports data from a local DB2 database to a delimited file.	✓	✓	✓	

		Agent sites				
Name	Description	Windows NT, Windows 2000, or Windows XP	AIX or Linux	Solaris Operating Environment	iSeries	z/OS
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.	✓	✓	✓	✓	✓

Warehouse transformers

The following table lists the warehouse transformers. Step subtypes are organized by program group. Program groups are located on the left side of the Process Model window. A *program group* is a logical grouping of related programs.

Defining and running processes

Name	Description	Agent sites				
		Windows NT, Windows 2000, or Windows XP	AIX or Linux	Solaris Operating Environment iSeries	z/OS	
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.	✓	✓	✓	✓	✓

		Agent sites				
Name	Description	Windows NT, Windows 2000, or Windows XP	AIX or Linux	Solaris Operating Environment	iSeries	z/OS
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.	✔	✔	✔	✔	✔

Statistical transformers

The following table lists statistical transformers. Step subtypes are organized by program group. Program groups are located on the left side of the Process Model window. A *program group* is a logical grouping of related programs.

		Agent sites				
Name	Description	Windows NT, Windows 2000, or Windows XP	AIX or Linux	Solaris Operating Environment	iSeries Version 4 Release 5 and later	z/OS
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.	✔	✔	✔	✔	✔

Defining and running processes

Name	Description	Agent sites				
		Windows NT, Windows 2000, or Windows XP	AIX or Linux	Solaris Operating Environment	iSeries Version 4 Release 5 and later	z/OS
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.	✓	✓	✓	✓	✓

		Agent sites				
Name	Description	Windows NT, Windows 2000, or Windows XP	AIX or Linux	Solaris Operating Environment	iSeries Version 4 Release 5 and later	z/OS
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.	✓	✓	✓	✓	✓

Defining and running processes

User-defined programs

The following table lists user-defined programs. Step subtypes are organized by program group. Program groups are located on the left side of the Process Model window. A *program group* is a logical grouping of related programs.

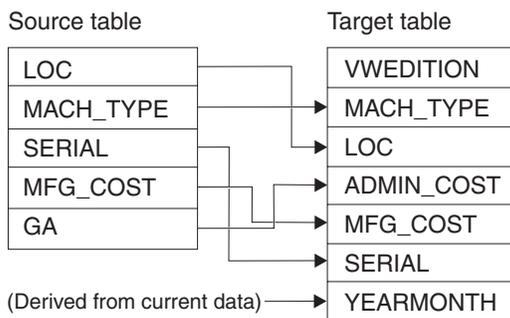
Name	Description	Agent sites			
		Windows NT, Windows 2000, or Windows XP	AIX or Linux	Solaris Operating Environment iSeries	z/OS
Format date and time	Changes the format in a source table date field.	✓	✓	✓	

Column mapping

When you use the Data Warehouse Center, it is easy to manipulate the data. 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.

The following picture shows 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, 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. Use the Column Mapping page to perform the following tasks:

- To create a mapping, click a source column and drag it to a target column. An arrow is drawn between the source column and the target column.
- To map all of the unmapped columns, click **Actions** → **Map All** and click one of the following options:

By Name Only

Source columns will be mapped to target columns with the same name.

By Name and Type

Source columns will be mapped to target columns with matching names and compatible data types.

By Position

Source columns will be mapped to target columns based on their position in the table.

- To remove a mapping, right-click an arrow, and click **Remove**.
- To remove all mappings, click **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.
- A target column that has been created during the current session of the Properties notebook for the step, can be edited or removed on the Column Mapping page. Once 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 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.
- Click **Actions** → **Find Columns** to locate columns in the source columns list, target columns list, or both. Wildcard characters are not supported.

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 may follow certain rules. This information is described, where applicable, in the step subtype descriptions that follow.

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.

Development mode

When you first create a step, it is in development mode. You can change any of the step properties in this mode. The Data Warehouse Center has not created a table for the step 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 were as you expected them to be.

Before you run the 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 the 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 the steps to production mode. Production mode indicates that the steps are in their final format. In production mode, you can change only the settings that will not affect the data 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.

Verifying the results of a step that is run in test mode

After you have promoted a step to test mode, you can verify that the target table was created.

Procedure:

To verify that the target table was created:

1. If the DB2 Control Center is not open, click **Tools** —> **Control Center** to open the DB2 Control Center from the main Data Warehouse Center window.
2. 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 is 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 Sample Contents 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 are tables that contain temporary data. After the data is used by the steps, it is purged from the table.

Restrictions:

You must have more than one step to run a process with a transient table.

Procedure:

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.
3. Schedule the steps to run at a specific date and time, or run the steps manually.

Defining and running processes

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.

Running a step from outside the Data Warehouse Center

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: XTServer and XTClient. XTServer is installed with the warehouse server. XTClient is installed with the warehouse agent for all agent types.

Prerequisites:

- A step must have Run on demand selected on the Processing Options page of the Properties notebook for the step before it can be triggered by the external trigger program.
- To run the client or server trigger, the CLASSPATH must point to sqllib/tools/db2XTrigger.jar and sqllib/java/common.jar.
- To use the external trigger program to start a step, you must have the JDK 1.3 installed on the warehouse server workstation and the agent site. You can also use the JDK 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

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 machine.

The syntax for starting the external trigger server is as follows:

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

The syntax for starting the external trigger client is as follows:

XTClient

```
▶—java—db2_vw_xt.XTClient—TriggerServerHostName—TriggerServerPort—DWUserID—DWUserPassword—▶
▶—StepName—Command—┌─WaitForStepCompletion─┐┌─RowLimit─┐▶
```

ServerHostName

The TCP/IP host name for the workstation on which the External Trigger Server (XTServer) is running.

Specify a fully qualified host name.

ServerPort

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 External Trigger Server (XTServer) is running on.

DWUserID

A user ID with Data Warehouse Center Operations privileges.

DWUserPassword

The password for the user ID.

StepName

The name of the step to start.

The name is case-sensitive. Enclose the name in double 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

Defining and running processes

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.

WaitForStepCompletion

Optional. This parameter indicates whether the external trigger program is to return the results of the step processing. Choose one of the following values:

1 Wait for step completion, and return 0 if the step completes successfully, or return an error if the step failed.

0 or blank

Do not wait for step completion.

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 or blank

Fetch all rows

n Fetch *n* rows

This parameter is valid only when the step is in test mode.

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
```

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, then the external trigger program will wait until the step finishes running and then will return the return code from that run.

The external trigger program returns a nonzero return code if it could not 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.

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 to which you want to print the information

Promoting warehouse processes

This section covers information about promoting warehouse processes.

About 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 has to be enabled.

Related tasks:

- “Promoting a step : Data Warehouse Center help” in the *Help: Data Warehouse Center*
- “Steps and tasks : Data Warehouse Center help” in the *Help: Data Warehouse Center*

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.

Defining and running processes

Step task flow

In the Data Warehouse Center, there are two ways that you can start a step. 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 have scheduled the first step, you can specify that another step is to start after the first step has run, and specify that a third step is to start after the second step has run, and so on.

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, 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 considered when 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 considered when 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.

Only the steps that are in production mode are run in a cascading process.

About scheduling processes

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.

While you can enjoy some flexibility by using cascading processes, please keep the following information in mind when constructing your 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, be careful not to use steps that reference 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.

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

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. You should see 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 open the Work In Progress window from the main Data Warehouse Center window, click **Data Warehouse Center** → **Work in Progress**.

Related tasks:

Defining and running processes

- “Viewing build-time errors using the basic logging function” on page 266
- “Viewing log entries in the Data Warehouse Center” on page 266
- “Work in Progress -- Overview : Data Warehouse Center help” in the *Help: Data Warehouse Center*
- “Canceling steps that are currently running : Data Warehouse Center help” in the *Help: Data Warehouse Center*
- “Purging steps and processes from the Work in Progress window : Data Warehouse Center help” in the *Help: Data Warehouse Center*
- “Rerunning steps from the Work in Progress window : Data Warehouse Center help” in the *Help: Data Warehouse Center*
- “Running steps from the Work in Progress window : Data Warehouse Center help” in the *Help: Data Warehouse Center*
- “Viewing the current status of a step or process : Data Warehouse Center help” in the *Help: Data Warehouse Center*

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.

Sampling data

Use Sample Contents to verify the data in a table in the Data Warehouse Center.

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.

Note: Sample Contents uses the first agent site returned from the metadata.

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 a 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, since the performance of inserts can be impacted when inserting a small numbers of rows.

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.

Prerequisites:

You cannot change an SQL step that is in production mode.

Procedure:

Selecting, inserting and updating source data in a target table

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” in the *Help: Data Warehouse Center*
- “Defining an SQL step: Data Warehouse Center help” in the *Help: Data Warehouse Center*

Incremental commit

The incremental commit option allows you 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).

You should consider the following concerning incremental commit:

- If an error occurs after an incremental commit is performed, you must 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.
- In a step where the population type is Replace, an incremental commit is used 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 issued.
- Steps with a population type of Append with editions can contain partial data if an error occurs after a commit is issued, but the outdated editions will not be deleted.
- The performance of your database can suffer because a significant number of commits can occur.
- If the source and target databases 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.
- 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.

Selecting, inserting and updating source data in a target table

- 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 prior to the error will appear in the target database.
- 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.

Joining source columns

This section describes how to join the columns in source tables.

Defining a join

If you select multiple sources, you also 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:

To join source tables:

1. Define an SQL step.
2. Open the step.
3. Click the **SQL Statement** tab.
4. Click **Build SQL** to have automatically generate SQL. (Otherwise, 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 >.
The column moves from the **Available columns** list to the **Selected columns** list.
Click >> to add all the columns from the **Available columns** list to the **Selected columns** list.
8. Click the **Joins** tab.

Selecting, inserting and updating source data in a target table

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 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 request 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.

Tip: The source tables must exist for you to use the **Test** push button on the SQL Statement page. If you specified that the Data Warehouse Center is to create the tables, you must promote to test mode the steps that link to those tables as target tables to create the tables.

15. Click **OK** to save your changes and close the step Properties notebook.

Removing a join

You can remove a join 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.
6. Click **Unjoin**. The join line is removed.

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

Selecting, inserting and updating source data in a target table

each part. To do this, you must combine the decoding table with the source data that contains the encoded part numbers.

Procedure:

First, define the decoding table and the encoded part numbers table as part of a warehouse source. Then, select those tables as source tables for a step. You then click **Join** on the Joins page of SQL Assist to join the tables.

Another approach is to 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:

- “About adding nulls to joins” on page 147
- “Star joins” on page 149

About adding nulls to joins

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

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

Selecting, inserting and updating source data in a target table

the manager, if one exists, you generate 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 1	
Column A	Column B
1	A
2	B
3	C

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

Column A	Column B	Column C	Column D
2	B	2	X

Selecting, inserting and updating source data in a target table

Left outer join

Column A	Column B	Column C	Column D
1	A	null	null
2	B	2	X
3	C	null	null

Right outer join

Column A	Column B	Column C	Column D
2	B	2	X
null	null	4	2

Full outer join

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:

1
2
3
4

Related tasks:

- “Transforming codes” on page 146

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

Selecting, inserting and updating source data in a target table

table contains information about the books that were ordered from each catalog by each customer during the fiscal year.

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.

To remove search conditions, highlight the portion of the condition that you want to remove in the **Conditions** field, and press the **Delete** key on your keyboard.

Specifying search conditions when filtering data

When you are defining an SQL step, you can specify search conditions when specifying data.

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 a parameter is specified, its value is used in the search condition. A parameter is specified 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.

Selecting, inserting and updating source data in a target table

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:

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
```

Defining a calculated column

You also 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 far 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.

Adding a calculated column to a step

You can type your expression in the **Expression** field, or you can use the fields and controls in the Expression Builder to build your expression.

Procedure:

Selecting, inserting and updating source data in a target table

To build an expression:

1. From the Properties notebook for an SQL step, click **Build SQL** to open SQL Assist.
2. Click **Add** to open the Expression Builder window.
3. Use the **Columns**, **Operators**, and **Case** lists to select the components of the expression. Double-click on a particular column, operator, or case keyword to add it to the **Expression** field. Each item that you double-click is appended to the expression in the **Expression** field, so be sure to select items in the order that you want them displayed.
4. 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.
5. Add a function to your expression.
6. Add a constant to your expression.
7. Use the following buttons to work with your expression:
 - Click **And**, **Or**, **=**, **<>**, **(**, and **)** as needed to add those operators to your expression.
 - Click **Clear** to remove all input from the **Expression** field.
 - Click **Undo** to remove the last change that you made from the **Expression** field.
 - Click **Redo** to reverse the last change that you made in the **Expression** field.
8. After you complete your expression, click **OK**. The Expression Builder window closes, and the column expression is added to the **Selected columns** list on the Columns page.
9. Click the **Name** field of the new column, and type the name of the column.
10. Press Enter.
11. Click **Move Up** and **Move Down** to move the column to the appropriate position in the table.

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.

Selecting, inserting and updating source data in a target table

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 databases.

Data Warehouse Center load and export utilities

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 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.

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\sqlib\logging\` on Windows[®] NT, Windows 2000, and Windows XP, 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 describe how to define the basic values for these utilities.

Exporting data

You can use the Data Warehouse Center utility to export data from a DB2 Universal Database database or a database that is registered in ODBC.

Defining values for a DB2 UDB export utility

Use the Properties notebook for the DB2 UDB export step to define a step that creates DB2 scripts that are run by the warehouse agent. These scripts export data from DB2 Universal Database tables or views to a file located at the agent site.

Loading and exporting data

The database server does not need to be on the agent site.

DB2 UDB export creates the target file if it does not exist, and replaces it if it exists.

This warehouse utility is available for the following operating systems:

- Windows NT
- Windows 2000
- Windows XP
- AIX
- Linux
- Solaris Operating Environment

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, the format must be IXF and the target file type must be Fixed.

Procedure:

To define a DB2 UDB export warehouse utility:

1. Link the source table that you want to export the data from to the step.
2. Link the target file that 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” in the *Help: Data Warehouse Center*

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 warehouse sources and targets a source.

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:

- Windows NT
- Windows 2000
- Windows XP
- AIX
- Linux
- Solaris Operating Environment

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, the format must be IXF and the target file type must be Fixed.

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” in the *Help: Data Warehouse Center*

Loading data

Use the warehouse utilities to load data into a DB2 Universal Database, DB2 for iSeries, or DB2 for z/OS.

Defining values for a DB2 Universal Database load utility

Use the properties notebook for the DB2 Universal Database Load step to create a step 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.

Loading and exporting data

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 utility detects a failure during processing, the table is emptied. 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:

1. Define a flat-file warehouse source for your source file. In the **File name** field, type the fully qualified file name.

Loading and exporting data

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.

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” in the *Help: Data Warehouse Center*

Modstring parameters for DB2 for iSeries Load utilities

This field is used 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:

Loading and exporting data

- 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 on the default values for the CPYFRMIMPF command, see the iSeries™ online help for the CPYFRMIMPF command.

The format for the FileMod string is:

1. 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.
2. Each keyword must be followed immediately by its value. The value must be enclosed in parentheses.
3. Each keyword must be separated from the next keyword by a space.

Requirement: Certain parameters require that you enclose values in two single quotation marks. For example, the FLDDLM command must have the values enclosed by two 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 prevents the command line processor from confusing a single quotation mark with the regular 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 Concepts 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.

Important: Successful completion of this utility 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:

VWxxxxxxx.programName

where:

- xxxxxxxx is the process ID of the VWPLOADI run that produced the file.
- programName is the name of the program, VWPLOADI or VWPLOADR.

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 panel is displayed.

2. Select option **30** to display all information. A message similar to the following message is displayed:

```
Message ID . . . . . : CPF2817
Message file . . . . . : QCPFMSG
Library . . . . . : QSYS
Message . . . . . : Copy command ended because of error.
Cause . . . . . : An error occurred while the file was
                  being copied.
Recovery . . . . . : See the messages previously listed.
                  Correct the errors, and then try the
                  request again.
```

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 cut 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 you may have encountered.

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 the file extension that is appropriate for the program to Client Access/400. For

Loading and exporting data

example, .VWPLOADI for the DB2 for iSeries Load Insert program and .VWPLOADR for the DB2 for iSeries Load Replace utility. Defining this extension allows Client Access/400 to translate the contents of files with this extension from EBCDIC to ASCII.

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 NT, Windows 2000, or Windows XP workstation:

1. Set up a Client Access/400 connection to your iSeries system over TCP/IP.
2. Open the Windows NT, Windows 2000, or Windows XP File Explorer.
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 164

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 NT, Windows 2000, or Windows XP 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**.
5. Select all file extensions.
6. Click **Add**.
7. Click **OK**.
8. Load the file into any ASCII text editor or word processor.

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:

Loading and exporting data

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.
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” in the *Help: Data Warehouse Center*

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 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.

SAP R/3

This section explains how load data into the Data Warehouse Center using the SAP R/3 connector.

Loading data into the Data Warehouse Center from an SAP R/3 system

With an SAP Data Extract step, you can import the data for SAP business objects that are defined in SAP R/3 system into a DB2 data warehouse. Business objects and business components provide an object-oriented view of SAP R/3 business functions. You can then use the power of DB2 and the Data Warehouse Center for data analysis, data transformation, or data mining.

The connector for SAP R/3 runs on Microsoft Windows NT, Windows 2000 and Windows XP. The SAP R/3 server can be on any supported operating system.

Prerequisites:

The SAP R/3 connector must be installed.

Procedure:

To load data into the Data Warehouse Center from an SAP R/3 system:

1. Define an SAP source.
2. Define the properties of each source business object.
3. Define an SAP step.

Related tasks:

- “Defining an SAP source in the Data Warehouse Center” on page 168

Defining an SAP source in the Data Warehouse Center

When you define an SAP source, you see all the metadata about the SAP business object, including the key fields and parameter names. You also see information about the parameter names such as the data type, precision, scale, length, and whether or not the parameters are mandatory. You also see all basic and detailed parameters that are associated with the SAP business object.

Procedure:

To define an SAP source to the Data Warehouse Center:

1. Right-click the **Warehouse Sources** folder, click **Define**, and then click **SAP**. The SAP business object must exist in the SAP R/3 system before you can define it in the Data Warehouse Center.
2. In the **Name** field, type a name for the SAP source.
3. Optional: In the **Administrator** field, type the name of the person who is responsible for the source definition.
4. Optional: In the **Description** field, type a brief description of the source.
5. Optional: In the **Notes** field, type information about the source that might be useful to users who can access the source properties.
6. On the Agent Sites page of the Define Warehouse Source notebook, move one or more agent sites to the **Selected agent sites** list. Select the agent sites where the SAP connector has been installed.
7. On the SAP Source page of the Define Warehouse Source notebook, specify whether the connection type for the SAP source is an application server or a group of servers (as defined to R/3).
 - For an application server, supply the application server name and the server's system number (as defined to R/3). You can also optionally supply the gateway host and service name, and the specific destination (an entry in the `saprfc.ini` file on the local work station).
 - For a group of servers, supply the server's system ID, message server name, group name, and optionally, its destination (an entry in the `saprfc.ini` file on the local work station). For either an application server or a group of servers, supply the client number, the user ID, and password for connecting to the SAP R/3 server.
8. On the Business Objects page of the Define Warehouse Source notebook, expand the Business Objects folder and select the specific SAP business objects that you want to define as part of this warehouse source from a list of business objects defined on the selected SAP R/3 system.

It might take several minutes for the Data Warehouse Center to retrieve the list of business objects from the SAP R/3 system, depending on network traffic and workload on the R/3 system.

Defining the properties of SAP source business objects in the Data Warehouse Center

After you define the SAP source to the Data Warehouse Center, you can define properties for each source business object:

Procedure:

To define the properties of each source business object:

1. Expand the list of SAP source objects in the tree and click the **Business Objects** folder for an SAP source.
2. Right-click one of the business objects in the details view of the navigator and click **Properties**.
3. On the Source Business Object page of the Properties notebook. The description of the business object is visible and you can edit it.
4. On the Parameters page of the Properties notebook, you can see the parameters that were defined for the selected business object in the SAP R/3 system, with the data type, length, precision, and scale (if applicable). You can also see which parameters are key fields and which are mandatory.
5. If you want detailed information for the business object you must map key fields for the business object to export parameters from the business object's GetList method (part of the Business Application Programming Interface) on the Column Mapping page of the Properties notebook. Click on a key field on the right and drag the cursor to an export parameter to map the two together. You must map all key fields, but not all of the export parameters need to be mapped.

WebSphere Site Analyzer

This section explains how to load data into the Data Warehouse Center using the WebSphere Site Analyzer connector.

Loading data from a WebSphere Site Analyzer database into the Data Warehouse Center

IBM WebSphere Site Analyzer (WSA) is part of the IBM WebSphere family of Web servers and application servers. WSA helps you analyze the traffic to and from your Web site.

The Connector for the Web allows you to extract data from a WebSphere Site Analyzer database, or webmart, into a data warehouse. The Connector for the Web provides a polling step that checks whether WSA copied Web traffic data from its data imports (log files, tables, and clickstream data) to the webmart.

Loading and exporting data

After this check is successful, an SQL step could copy the Web traffic data from the webmart into a warehouse target. You can then use DB2 and DB2 Warehouse Manager for data analysis, data transformation, or data mining. You can also incorporate WebSphere Commerce data with the Web traffic data for a more complete analysis of your Web site. After defining a WSA source, you can define the Web traffic polling step from the Data Warehouse Center in the process modeler.

The Connector for the Web runs on the same platform as the warehouse agent: Windows NT, Windows 2000, or Windows XP operating systems, AIX, or the Solaris Operating Environment.

Procedure:

To extract data from a Websphere Site Analyzer database:

1. Define a WebSphere Site Analyzer source.
2. Define a Web traffic polling step.

Related tasks:

- “Defining a WebSphere Site Analyzer source in the Data Warehouse Center” on page 170
- “ Specifying database information for a Web traffic source : Data Warehouse Center help” in the *Help: Data Warehouse Center*
- “ Importing source data imports, tables, and views into a Web traffic source : Data Warehouse Center help” in the *Help: Data Warehouse Center*
- “ Defining a warehouse source based on IBM WebSphere Site Analyzer : Data Warehouse Center help” in the *Help: Data Warehouse Center*
- “ Adding information about the Web Traffic Polling step : Data Warehouse Center help” in the *Help: Data Warehouse Center*
- “Defining a Web Traffic Polling step : Data Warehouse Center help” in the *Help: Data Warehouse Center*

Defining a WebSphere Site Analyzer source in the Data Warehouse Center

Before you load data from a WebSphere Site Analyzer source into the Data Warehouse Center, define a warehouse source.

Procedure:

To define a WebSphere Site Analyzer source to the Data Warehouse Center, right-click the **Warehouse Sources** folder, and click **Define**, then click **WebSphere Site Analyzer**.

Related tasks:

- “ Defining a warehouse source based on IBM WebSphere Site Analyzer : Data Warehouse Center help” in the *Help: Data Warehouse Center*

Loading and exporting data

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 will FTP the remote flat file to a local file on the iSeries system, and then the load step will load 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:

- - Windows NT
 - Windows 2000
 - Windows XP

Moving files and tables

- AIX
- Linux
- Solaris Operating Environment
- iSeries
- z/OS

The Submit JCL jobstream program is also available for all of the operating systems mentioned above except for iSeries.

Prerequisites:

Copy file using FTP warehouse program

Before you copy files to z/OS using the Copy file using FTP program, you 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 discussed in this topic, open the Properties notebook for the step, and specify properties for the step.

Unable to access 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 glacier.stl.ibm.com. You want to transfer a file using FTP from the remote site kingkong.stl.ibm.com to the agent site, using the remote user ID vwinst2. The ~vwinst2/.netrc file must contain the following entry:

```
machine glacier.stl.ibm.com login vwinst2
```

Related tasks:

- “Importing table definitions from a Microsoft Access database (Windows NT, Windows 2000, Windows XP)” on page 72

Replication

This section describes how to set up replication in the Data Warehouse Center.

Replication in the Data Warehouse Center

Replication is a process of maintaining a defined set of data in more than one location. It 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 machine or on different machines in a distributed network.

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 having to fully load the table each time 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 Relational Connect 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 data updated and how you want transactions handled.

Moving files and tables

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 point in 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 against 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 CD table (the Changed Data table) which contains the recently changed data.

Staging table

Produces target tables that contain data from committed transactions. Also called consistent change-data tables (CCD tables), these tables contain different types 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 updates using the primary key. Condensed tables are useful for staging changes to remote locations and for summarizing hot-spot updates before they are replicated to targets.
- 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.

For a replication step, promoting to test mode will create the target table and generate the subscription set in the replication control table. 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 have been defined. You can make changes to a step only when it is in development mode.

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 both the warehouse control and warehouse target databases.
- 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.

Procedure:

To set up replication in the Data Warehouse Center, you must complete the following steps:

1. Create the replication control tables using the Replication Center.
2. Register a source table using the Replication Center.
3. Import the defined replication source table into 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 required by your warehouse step. Replication steps assume that the password file for the step will be found in the `VWS_LOGGING` (environment variable) directory, and have a file name of `applyqual.pwd` where "applyqual" is the apply qualifier for the step. This information is added to Apply program command before Apply 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 or production 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.

Moving files and tables

Related tasks:

- “Using the z/OS warehouse agent to automate DataPropagator replication apply steps” in the *DB2 Warehouse Manager Installation Guide*

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 importing a replication source be sure to mark 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 CD table which is created by DB2 Replication Center. In the CD table, the before-image columns names will 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 go back to the DB2 Replication Center and redefine the replication source there. Then you must re-import the replication source into Data Warehouse Center.

Creating replication control tables

Replication control tables must exist in both the control and target databases before you can set up replication in the Data Warehouse Center. The replication control tables are found in the ASN schema and the control table names all 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 defaults for all settings.
6. Click **Finish**.
7. Click **Close** in the Message Dialog.
8. In the Run Now or Save SQL Dialog, remove the SQL statements that insert into the IBMSNAP_REGISTER table and the statements that create the CD table.
9. Click **OK** to run the SQL script.
10. Repeat these steps to create the replication control tables in the warehouse target database.

Defining a replication step in the Data Warehouse Center

There are five types of replication in the Data Warehouse Center:

- User copy
- Point-in-time
- Base aggregate
- Change aggregate
- Staging table

Procedure:

To define a replication step, open the Properties notebook for the step that you want to define, and specify the step properties.

Related tasks:

- “Creating replication control tables” on page 178
- “Defining a base aggregate replication step : Data Warehouse Center help” in the *Help: Data Warehouse Center*
- “Defining a change aggregate replication step : Data Warehouse Center help” in the *Help: Data Warehouse Center*
- “Defining a point-in-time replication step : Data Warehouse Center help” in the *Help: Data Warehouse Center*
- “Defining a staging table replication step : Data Warehouse Center help” in the *Help: Data Warehouse Center*
- “Defining a user copy replication step: Data Warehouse Center help” in the *Help: Data Warehouse Center*

Using a replication step in a process

This topic explains how to use a replication step a Data Warehouse Center process.

Moving files and tables

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.
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. You must start the Capture program before you run the step.

To start the Capture program, type the following command in a DOS window:

```
ASNCCP CAPTURE_SERVER=source-database STARTMODE=WARMSI
```

The STARTMODE=WARMSI parameter specifies an initial COLD start that deletes any existing data in the CD 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. Make sure that the password file has been created for the replication step and its Apply qualifier.
8. 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.
9. Schedule the step.
10. Promote the step to production mode.

Replication password files

Replication password files in the current release

For replication in the current release of the Data Warehouse Center, the user must setup and maintain the replication password files using the replication program asnpwd. The replication password files are encrypted for security.

Data Warehouse Replication steps assume that the password file for a DW Replication step will be found in the VWS_LOGGING (environment variable) directory and have a file name of applyqual.pwd where "applyqual" is the Apply qualifier for the Data Warehouse step. This information is added to Apply program command before Apply is started by the DW Agent. For example:

```
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:\sqllib\logging pwdfile=dwqual.pwd
```

Replication password files in the previous release

The warehouse agent creates the replication password file automatically before the warehouse replication step is run. The password file is deleted when the step has finished. Unlike the current release, the user does not have to maintain the replication password files. The password files are not encrypted.

The password file has the file name *applyqual.pwd* where *applyqual* is the Apply qualifier for the step. The password file is created in the ASNPATH (environment variable) directory if ASNPATH is specified. If ASNPATH is not found, the password file is created in the current directory on the system where the Apply program runs.

Chapter 12. Transforming data

This chapter describes the methods of transforming data that the Data Warehouse Center provides. It 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 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” in the *Help: Data Warehouse Center*
- “Assigning key column values : Data Warehouse Center help” in the *Help: Data Warehouse Center*
- “Creating a period table : Data Warehouse Center help” in the *Help: Data Warehouse Center*

Transforming data

- “ Inverting rows and columns in a table : Data Warehouse Center help” in the *Help: Data Warehouse Center*
- “Grouping related data from selected columns into a single column : Data Warehouse Center help” in the *Help: Data Warehouse Center*

Clean Data transformer

The Clean Data Transformer can be used to perform basic scrubbing, 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 will be written to the target error table. You can also use the Clean Data transformer to clean and standardize data values after load or import, as part of a process.

Rules for the Clean Data transformer

The rules that can be applied to relational source data using the Clean Data transformer are:

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 specified 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 string¹
- Find substring, replace whole string¹
- Find substring, replace substring¹

¹Available only for CHARACTER data types.

Use the Matching Options window to specify how matches will 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 exactly 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 you have turned on the error processing option by selecting the multiple match option, Write to error table, or by enabling error processing, the entire input row is written to the error table along with the RUN_ID of the execution.

If you allow nulls for this clean type, you must put a null value in the Find column of the rules table.

Numeric clip

For numeric values only. This clean type clips numeric input values which are not within the specified range. Input values within the range are written to the output as is. 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 matches will be handled.

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. Use the Matching Options window to specify how matches are handled.

If you allow nulls for this clean type, you must put a null value in the Bound column of the rules table.

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. Carry over allows you to replace null values with a specified value. You can also reject nulls and write the rejected rows to the error table.

Convert case

Converts the characters in the source column from uppercase to lowercase or from lower case to uppercase, and inserts them into the target column. The default is to convert the characters in the source column to upper case. A rules table is not required for this clean type.

Encode invalid values

Replaces any values that are not contained in the valid values column of the rules table that you are using with the specified value. 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.

Transforming data

Additional functions

In addition to providing you with basic ETML capabilities and creating SQL for you, the Clean Data transformer also provides you with the following features to be used with the clean data rules.

Data types support: The carry over column rule will support any SQL data type to be carried over. Other rules will now have the flexibility of specifying a different replace value data type as opposed to previous versions. For example, an integer value can be replaced by a character value, instead of the same data type as in the case of 1 is replaced by 'A'.

The following table shows the data types that are supported for each clean type:

Clean type	Character	Numeric	Date/Time	Graphic	LOB
Carry over	✓	✓	✓	✓ ¹	✓ ¹
Find and replace	✓	✓	✓		
Discretize	✓	✓	✓		
Clip		✓			
Convert case	✓				
Encode invalid values	✓	✓	✓		

¹Carry over supports Graphic and LOB data types only in the cases where Insert from Select SQL statements allow this type of column.

Null support: You can specify explicitly 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 where you have a rules table, the replacement value should be compatible with the data type of the rules replace column. The replacement value overrides the rules regarding 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:

First and last matching

The rules implement first or last match criteria depending on what you specified. The first match option looks for the first match in the

relevant rules table. This is to avoid 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 find and replace, the default order is the minimum replace value when an exact match is specified, otherwise it is the minimum find and replace value pair for a substring match. The same is 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 also implies that records that do not match any rule are also put into the error table.

Create all matches

Default. You can create all the matching values into the target table. This means there would exist a 1:n mapping between an input row and its corresponding output rows. For example, this could happen when you have duplicate find values or overlapping ranges in a rules table.

Differentiator columns: The user can optionally specify a differentiator column in the input table and the rules table which should match exactly to satisfy the overall match of an input in the source table and a rule in the rules table.

Automatic Summary Tables: Optional: 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 feature is off by default.

Error processing: Records that do not match rules are written to an error table as appropriate. 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.

Transforming data

- 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 the case when you are demoting the step to development.

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:

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>
------------	-------	--

Transforming data

Numeric 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>
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.

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.

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.

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.

Transforming data

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 begin this task, 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.

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 be 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.

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.

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.

Transforming data

10. Click **OK**.

The Function Arguments - FormatDate window closes, and the expression is displayed in the Expression Builder window.

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.

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 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.

The Data Warehouse Center already provides integration with tools from Vality and Evolutionary Technologies, Inc.

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.
- Users must have a working knowledge of Trillium software.

The following table shows the software that is required for cleaning name and address data in the Data Warehouse Center.

Operating system	Required software
UNIX	Trillium Software System Version 4.0 DB2 Universal Database Warehouse Manager Version 7.2 or later warehouse agent
Windows NT, Windows 2000, Windows XP	Trillium Software System Version 4.0 DB2 Universal Database Warehouse Manager Version 7.2 or later warehouse agent For remote access, the host 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, Windows NT, Windows 2000, or Windows XP TCP/IP 3.2 or later must be installed The z/OS operating system is supported as a remote host only

Restrictions:

Transforming data

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 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

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.

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 1. Sample Trillium script file

```
INP_FNAME01 c:\tril40\us_proj\data\convinp
INP_DDL01 c:\tril40\us_proj\dict\input.ddl
```

Figure 2. 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 3. 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 4. The Trillium Batch System step definition

Related reference:

- “Trillium DDL to Data Warehouse Center metadata mapping” on page 276

Importing Trillium metadata

In addition to importing other kinds of metadata, you can also 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

Transforming data

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 double quotation marks when you type them into the Script or JCL, or Output error file fields.

Procedure:

To import Trillium metadata:

1. Create a Trillium Batch System script or JCL.
2. From the Data Warehouse Center tree, right-click **Warehouse**, and click **Import Metadata** → **Trillium** to open the Import Metadata Trillium window.
3. Specify the information for your import.

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.

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.

Parameters for the Trillium Batch System script or JCL

The following table contains the 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 RemoteHostName is localhost.

Transforming data

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 double quotation marks.• The first parameter in the string must be the name of the output file where the password will be written.
Password	<p>The valid value is the password or password program name. The password program must be local to the warehouse agent site.</p>
Program parameters	<p>The parameters for the password program.</p>
Output Error File	<p>The name of the output error file.</p>
<p>Note: The data type for all of the parameters in this table is CHARACTER.</p>	

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
```

Error return codes for name and address cleansing

The following table lists the 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.

Transforming data

16	<p>The Trillium Batch System user-defined program cannot create the log or an internal file.</p> <p>Check to see if the user has the correct authorization. Check to see if the disk is full.</p>
20	<p>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.</p> <p>Check the JESLogFile to identify the reason.</p>
48	<p>Environment variable VWS_LOGGING cannot be found or the log file cannot be created. Check the log file for more information.</p>
56	<p>Either the Windows NT or UNIX script cannot be executed, 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.</p>
500	<p>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.</p>

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.

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.

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.

Calculating statistics

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 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” in the *Help: Data Warehouse Center*
- “Obtaining variability estimates between or within groups: Data Warehouse Center help” in the *Help: Data Warehouse Center*

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” in the *Help: Data Warehouse Center*

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:

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:

Source column	Target column
A_week	

Calculating statistics

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:

Source column	Target column
B	C

Because column B is mapped as a source column, the following mapping 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:

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.

Related tasks:

- “Calculating the running subtotal for numeric values grouped by period : Data Warehouse Center help” in the *Help: Data Warehouse Center*
- “Reference for Calculate subtotals : Data Warehouse Center help” in the *Help: Data Warehouse Center*

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.

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

Calculating statistics

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.

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.

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.

Calculating statistics

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.

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.

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

This chapter describes how to use the Data Warehouse Center export and import capabilities to export and import object definitions.

Exporting metadata

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 or XML file by default. The Data Warehouse Center export capabilities are available for the following operating systems:

- Windows NT
- Windows 2000
- Windows XP
- AIX
- Solaris Operating Environment

You can export the following types of metadata:

- Tag language
- Common Warehouse Metadata
- OLAP Integration Server (Windows NT and Windows 2000 only)

Export 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 heapsize 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.

Alternatively, if you want to make a copy of your warehouse control database (such as when you want separate test and production systems), you can use the DB2 Universal Database export and load utilities to copy the data. However, both databases must have a user ID and password that matches the default warehouse user, so that users can log on to the Data Warehouse Center.

Tag language and XML files

When you export metadata to a tag language 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 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.

Note: Tag 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 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 heapsize of the warehouse control database to 8192.

You can import the following types of metadata:

- Tag language file
- Common Warehouse Metamodel
- ERwin

Exporting and importing metadata in the Data Warehouse Center

- MQSeries
- Trillium

You can use the Data Warehouse Center import capabilities to import object definitions within the following operating systems:

- Windows NT
- Windows 2000
- Windows XP
- AIX
- Solaris Operating Environment

Restrictions:

- If you move a tag language or XML file from one system to another, you must move all the associated files along 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 or XML files as you want.

Procedure:

To import metadata into the Data Warehouse Center, open the Import Metadata window for the type of metadata that you want to import and specify parameters for the import.

Related tasks:

- “Importing objects : Information Catalog Center help” in the *Help: Data Warehouse Center*
- “ Importing an ERwin tag language file : Data Warehouse Center help” in the *Help: Data Warehouse Center*
- “Importing an MQ Series file : Data Warehouse Center help” in the *Help: Data Warehouse Center*
- “ Importing source SAP R/3 business objects into a warehouse source : Data Warehouse Center help” in the *Help: Data Warehouse Center*

Related reference:

- “Metadata mappings between the Data Warehouse Center and CWM XML objects and properties” on page 276

Publishing warehouse metadata

Publishing metadata is the process of transferring metadata from the Data Warehouse Center to the Information Catalog Center.

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*
- “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*

MQSeries

This section describes how to set up MQSeries in the Data Warehouse Center.

MQSeries data

With the Data Warehouse Center you can access data from a 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 a MQSeries message queue and a DB2 XML Extender Document Access Definition (DAD) file.

Exporting and importing metadata in the Data Warehouse Center

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
- <ServiceName>.<DAD file base name>.<Warehouse target Name> step
- <ServiceName>.<DAD file base name> program template

Creating views for MQSeries messages

Create a view to access data from an MQSeries message queue. The following requirements and restrictions apply.

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.

Restrictions:

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 or choose a different database alias in the wizard.

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

Exporting and importing metadata in the Data Warehouse Center

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 250

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.
- DB2 XML Extender Version 7.2 or later.
- MQSeries Support.

Restrictions:

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:
 - Define the warehouse target and register and enable the transformers.
 - Enable the warehouse target for DB2 XML Extender.
 - 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.

Exporting and importing metadata in the Data Warehouse Center

6. In the **Warehouse Target** field, select the name of the warehouse target where the step will run. The warehouse target must already be defined.
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.

MQSeries user-defined program

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.

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

Exporting and importing metadata in the Data Warehouse Center

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.

Error return codes for the MQXMLXF user-defined program

When the step runs, the stored procedure might return error code SQLCODE -443 and SQLSTATE 38600. The following table shows some possible diagnostic texts.

Error return codes

The following table shows the diagnostic texts for error codes returned by the MQXMLXF user-defined program.

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.

Error logging for MQSeries

The Data Warehouse Center stores all diagnostic information in a log file when MQXMLXF runs. The name of the log file is mqxf<nnnnnnnn>.log, where <nnnnnnnn> is the run ID that was passed to the stored procedure. The Data Warehouse Center will create the file in the directory 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.

Exporting and importing metadata in the Data Warehouse Center

```
db2set DB2ENVLIST="AMT_DATA_PATH VWS_LOGGING"
```

The log file is deleted if the step runs successfully.

Importing ERwin metadata

The following topics describe how to use the IBM ERwin Metadata Extract program to extract metadata from an ER1 file and create a DB2 Data Warehouse Center or Information Catalog Center tag language file.

IBM ERwin Metadata Extract Program

The 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, creating all the relevant relationship tags between objects, such as between databases and tables and between tables and entities. 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 metadata extract program supports all ER1 models with relational databases, including DB2, Informix, Oracle, Sybase, ODBC data sources and Microsoft[®] SQL Server.

Starting the IBM ERwin Metadata Extract program

This procedure explains how to start the IBM ERwin Metadata Extract program.

Prerequisites:

The following software is required to run the metadata extract program:

- Windows NT 4.0 or later
- ERwin 3.5.2 with Service Pack 3 Build 466

Procedure:

There are two ways to start the IBM ERwin Metadata Extract program:

1. Use Data Warehouse Center to import ER1 file.
2. Use the command `flgerwin` to generate .tag file from the ER1 file, and then use the command `iwh2imp2` to import the .tag file into Data Warehouse Center.

Exporting and importing metadata in the Data Warehouse Center

Creating tag language files for the IBM ERwin Metadata Extract program

Use the IBM ERwin Metadata Extract program to create a Data Warehouse Center or Information Catalog Center tag language file from the command line.

Procedure:

To use the IBM ERwin Metadata Extract program to create a tag language file from the command line, use the following syntax:

Syntax for creating a tag file

```
► flgerwin—inputFile.er1—outputFile.tag —-dwc—+—-m  
—-icm—-m—-starschema  
—-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 only specify 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 metadata extract program creates a tag file which defines the warehouse targets when it is imported into the warehouse. The warehouse targets are defined in Data Warehouse Center as logical entities and might not physically exist, or might be empty.

Related reference:

- “Mapping ERwin object attributes to Data Warehouse Center tags” on page 274

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 in order 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 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 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.

Importing ERwin metadata into the Data Warehouse Center

You can import a tag language file into the Data Warehouse Center in either of two ways:

- Data Warehouse Center import function
- Command line

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.

Procedure:

Exporting and importing metadata in the Data Warehouse Center

To use the Data Warehouse Center import function to import a tag language file:

1. In the Data Warehouse Center, right-click the **Warehouse** node, and click **Import Metadata** → **ERwin**.

The Import Metadata – ERwin window, and specify the parameters of the import.

2. Specify parameters for the import.

To import a tag language file using the command line, enter the following command:

1. `iw2imp2 tag-filename log-pathname target-control-db userid password.`
2. Launch the Data Warehouse Center.
3. Right-click the Warehouse node, and click **Import Metadata** → **Tag Language**.
4. Specify the parameters for the import.

Where:

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.

Related tasks:

- “ Importing an ERwin tag language file : Data Warehouse Center help” in the *Help: Data Warehouse Center*

Changing a DB2 database definition to a source in the Data Warehouse Center

By default, the extract program generates tag files which define databases to be warehouse targets. If you want to import these databases as warehouse sources, you must change the .tag file generated by the extract program. This can be done by first running the extract program from the command line, then editing the .tag file, and importing the tag 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.

Exporting and importing metadata in the Data Warehouse Center

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) TARGETYPE(DATABASE)** to **:RELTYPE.TYPE(LINK) SOURCETYPE(SCGSR CIR) TARGETYPE(DATABASE)** for each database that you want as a source.

IBM ERwin metadata extract program errors

If you receive an error message, look for the message here, and an explanation of how to resolve the error.

Missing ER1 input file or tag output file.

The 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 will be 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 metadata extract program put the file in read-only state. Issue the command:

```
attrib -r inputFile.er1
```

in a command shell to change the state of the ER1 file to read/write.

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 ERwin, 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.

Exporting and importing metadata in the Data Warehouse Center

To register the ERwin API, run the following command from the directory where your ERwin program files are installed: **regsvr32 er2api32.dll**. You will see a message, "DllRegisterServer in er2api32.dll succeeded." You can start the extract program from the Data Warehouse Center, or by issuing the flgerwin command from a command shell.

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.

During processing, you might receive the message, "Duplicate column found. Column will not be extracted." This is an informational message 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 currently being processed.

The object type "COLUMN" identified by "DBNAME(__) OWNER(__) TABLE(__) COLUMNS(__)" is defined twice in the tag language file

When a tag language file is imported, you might receive the following message :Message: DWC13238E The object of type "COLUMN" identified by "DBNAME(__) OWNER(__) TABLE(__) COLUMNS(__)" is defined twice in the tag language file.

This is an informational message, 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.

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.

Extending the Data Warehouse Center

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.

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.

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 NT, Windows 2000, Windows XP)

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:

Extending the Data Warehouse Center

To enable a user-defined program to run under Windows NT, Windows 2000, or Windows XP:

1. Define the warehouse agent or server service as a system process that can interact with the Windows NT, Windows 2000, or Windows XP 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.

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[®] NT, Windows 2000, or Windows XP workstation. It 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;
        }
    }
}
```

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® NT, Windows 2000, or Windows XP 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
- Polling interval
- Timeout interval

Extending the Data Warehouse Center

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.

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 using 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 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 that SQL returned a 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 will display 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 will display in the Log Viewer Details window for the step.
- The SQLSTATE return code, which the Data Warehouse Center will display 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, it should continue 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 set on, the step's processing status in the Work in Progress window will be 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.

Extending the Data Warehouse Center

The following example shows the feedback file.

```
<RC> 20</RC>
<ROWS>2345</ROWS>
<MSG>The parameter type is not correct</MSG>
<COMMENT> Please 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 vary, depending on the following values set by the program:

- 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 example lists the possible combinations of these values and the results that they produce.

Conditions	Step status ¹	Values of Error RC1 and RC2
<ul style="list-style-type: none">• The user-defined program return code is 0• No feedback file exists²	Successful	RC1 = 0; RC2 = 0
<ul style="list-style-type: none">• The user-defined program return code is 0• A feedback file exists²• The value of <RC> in the feedback file is 0³• <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 exists²• The value of <RC> in the feedback file is 0³• The value of <WARNING> in the feedback file is 1	Warning	RC1 = 0; RC2 = 0

Conditions	Step status ¹	Values of Error RC1 and RC2
<ul style="list-style-type: none"> • The user-defined program return code is 0 • A feedback file exists² • The value of <RC> in the feedback file is non-0³ • <WARNING> is not set in the feedback file 	Failed	<p>RC1 = 8410 (the user-defined program failed);</p> <p>RC2 = the value of <RC> in the feedback file</p>
<ul style="list-style-type: none"> • The user-defined program return code is 0 • A feedback file exists² • The value of <RC> in the feedback file is non-0³ • The value of <WARNING> in the feedback file is 1 	Warning	<p>RC1 = 0;</p> <p>RC2 = the value of <RC> in the feedback file</p>
<ul style="list-style-type: none"> • The user-defined program return code is nonzero • No feedback file exists² 	Failed	<p>RC1 = 8410 (the user-defined program failed);</p> <p>RC2 = the code returned by the user-defined program</p>
<ul style="list-style-type: none"> • The user-defined program return code is nonzero • A feedback file exists² • The value of <RC> in the feedback file is 0³ • <WARNING> is not set in the feedback file 	Successful	<p>RC1 = 0;</p> <p>RC2 = 0</p>
<ul style="list-style-type: none"> • The user-defined program return code is nonzero • A feedback file exists² • The value of <RC> in the feedback file is 0³ • The value of <WARNING> in the feedback file is 1 	Warning	<p>RC1 = 0;</p> <p>RC2 = 0</p>

Extending the Data Warehouse Center

Conditions	Step status ¹	Values of Error RC1 and RC2
<ul style="list-style-type: none">• The user-defined program return code is nonzero• A feedback file exists²• The value of <RC> in the feedback file is non-0• <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 exists²• The value of <RC> in the feedback file is non-0• The value of <WARNING> in the feedback file is 1	Warning	RC1 = 0; RC2 = the value of <RC> in the feedback file

Notes:

1. The step processing status, as displayed in the Work in Progress window.
2. The Data Warehouse Center checks for the existence of the feedback file, regardless of whether the return code for the user-defined program is 0 or nonzero.
3. 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.

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.

Extending the Data Warehouse Center

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.

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 multidimensional cubes necessary to support OLAP customers. A *multidimensional cube* is a set of data and metadata that defines a multidimensional database.

You would populate a star schema within the Data Warehouse Center with cleansed data before you use the data to build a multidimensional 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 multidimensional cube with data using the Data Warehouse Center and the DB2 OLAP Integration Server. The table lists a task and tells you which product or component you use to perform each task. Each task is described in this chapter.

Task	Complete task from Data Warehouse Center	Complete task from DB2 OLAP Integration Server
Define warehouse targets of relational data that you will use as sources for an OLAP multidimensional cube.	X	
Define the warehouse schema.	X	

Creating a star schema from within the Data Warehouse Center

Task	Complete task from Data Warehouse Center	Complete task from DB2 OLAP Integration Server
Export the warehouse schema to the DB2 OLAP Integration Server. The warehouse schema will be used as a model in the DB2 OLAP Integration Server.	X	
Use the model (warehouse schema) to define hierarchies for the model.		X
Use the model to define a metaoutline on which your multidimensional cube will be based.		X
Create a command script that will be used to load the multidimensional cube with data.		X
Create a batch file that runs the command script.		X
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 multidimensional cube from the Data Warehouse Center are generated by the fignxolv command.	X	
Schedule the step that was created by the export process so that you can populate the multidimensional cube.	X	

Designing the warehouse schema in the Data Warehouse Center

Use the Warehouse Schema Model window to generate 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).

Creating a star schema from within the Data Warehouse Center

Procedure:

To design a warehouse schema:

1. Define the warehouse schema
2. Add tables and views to the warehouse schema
3. Autojoin 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 target table that you plan to use as a dimension table or a fact table.
- **Requirement:** When you define the warehouse target for a warehouse schema, the warehouse target name must match exactly 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 that has access to the warehouse target that contains the tables can change the tables.

Only those 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:
 - 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.

Creating a star schema from within the Data Warehouse Center

- 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.

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. From the Data Warehouse Center, right-click Warehouse, and click **Export** → **OLAP Integration Server** to open the Export Metadata - OLAP Integration Server notebook.
2. Specify the information required to export the schema.

You can view the log file that stores trace information about the export process. The 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:\sql11ib\logging, where x is the drive where the DB2 Universal Database is installed. The name of the log file is FLGNXHIS.LOG.

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.

Creating a star schema from within the Data Warehouse Center

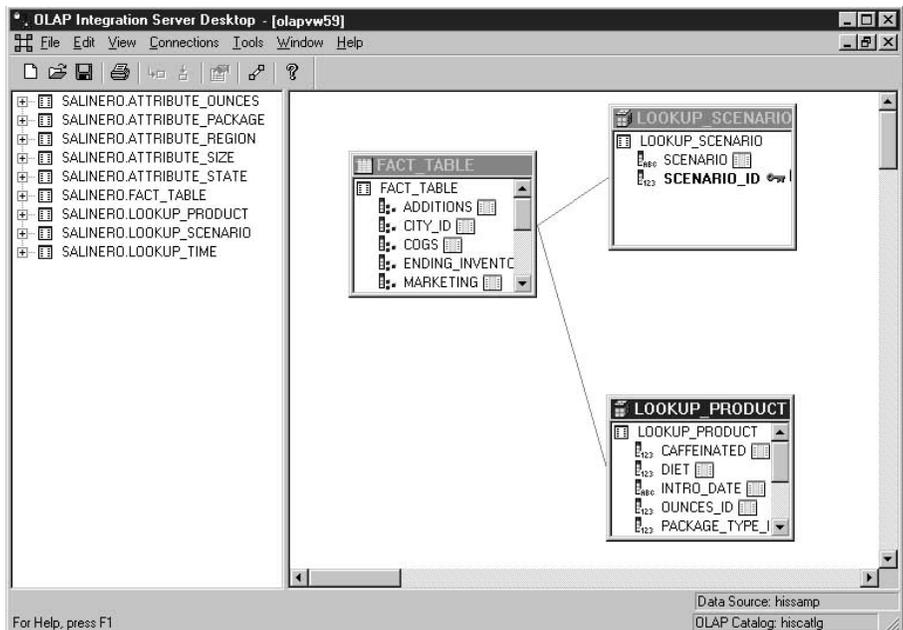
Completing the design for your multidimensional cube

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 multidimensional cube.

Procedure:

From the DB2 OLAP Integration Server, you must complete the following tasks:

1. Optional. View the warehouse schema that you exported. To view the warehouse schema that you exported, open the OLAP model (warehouse schema) using 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. The following image shows what your model looks like when you open it on the DB2 OLAP Integration Server desktop. The join relationships that you defined between the fact table and dimension tables are displayed.



2. Create an OLAP metaoutline, which is a subset of the OLAP model (warehouse schema) on which you will base your multidimensional 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*

Creating a star schema from within the Data Warehouse Center

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 multidimensional cube is defined. For example, your outline will contain the definitions of members and dimensions, members, and formulas. You will also define the script that is used to load the cube with data. Then you will define a batch file from which to invoke the script.
4. 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.

Creating an outline in the DB2 OLAP Integration Server

This section describes how to create an outline. When you have created the outline, you can associate it with a script that loads data into the multidimensional cube. 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.

See the online help for the DB2 OLAP Integration Server for detailed information on fields and controls in a window.

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.

Creating a load script to load data into the multidimensional cube in the DB2 OLAP Integration Server

After you have created an outline, you can create a load script that loads data into the multidimensional 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 do analysis on 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 As window opens.
9. Type a name and file extension for the command script file.
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 shows an example 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"
LOADALL
```

Creating a star schema from within the Data Warehouse Center

```
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 invokes the script. The batch file is used as a parameter for the Data Warehouse Center step that runs the script to load the multidimensional cube.

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.

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 DWcpw ModelName
```

HIScatalog

The name of the DB2 OLAP Integration Server metadata catalog.

Creating a star schema from within the Data Warehouse Center

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.

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 that are created and associated with the target tables in the warehouse when you export metadata from the DB2 OLAP Integration Server

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 named 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 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 invokes 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 is run, 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 (data in the multidimensional cube).

Creating a schedule to populate the multidimensional cube

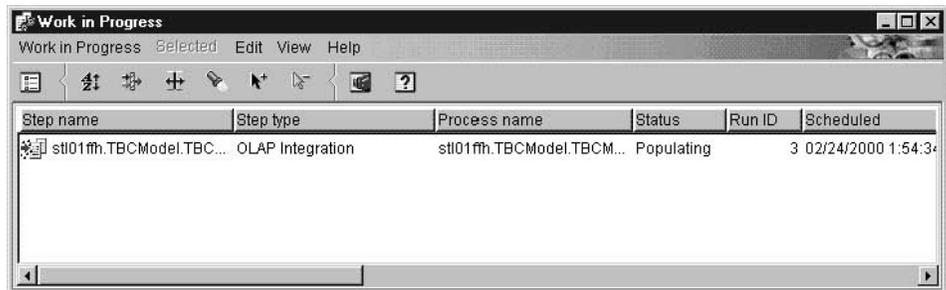
You can schedule the step that loads the multidimensional cube, and you can promote it so that it runs on a regular basis.

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. For more information about scheduling a step to run based on the success or failure of another step.
4. Right-click the step, and click **Mode**. You can click either **Test** or **Production** to run the step.

After you successfully schedule and test the step, the multidimensional cube that was built using your warehouse schema is populated. The following picture shows the Work in Progress window as a multidimensional cube is being populated.



About publishing metadata about your warehouse schema

You can use the Publish Data Warehouse Center Metadata 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:

- “Preparing to publish OLAP server metadata” in the *Information Catalog Center Administration Guide*

Creating a star schema from within the Data Warehouse Center

Related reference:

- “Metadata mappings between the Information Catalog Center and OLAP server” in the *Information Catalog Center Administration Guide*

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.

Maintaining the warehouse database

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.

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 DISCDNS. 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.

Maintaining the warehouse database

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.

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.

Maintaining the warehouse database

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 215

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” in the *Help: Data Warehouse Center*

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:

Maintaining the warehouse database

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 NT, Windows 2000, Windows XP)

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.

Managing the control database

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.
- 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.

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 you want to use as the active warehouse control database. You must stop the warehouse server before using the Warehouse Control Database Management tool.

The Data Warehouse Center will create 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 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.
2. Click **Start** → **Programs** → **IBM DB2** → **Set-up Tools > Warehouse Control Database Management** to open the Warehouse Control Database Management window.
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.
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.

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.

Managing the control database

Note: If you apply a fixpak 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.

Procedure:

To create a warehouse control database 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. After installation, you must then use the Warehouse Control Database Management window on the warehouse server workstation by following the steps for initializing a new warehouse control database. 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.

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.

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 OLAP Server warehouse programs, you must install Essbase Server or IBM DB2 OLAP Server software.

Procedure:

To use the Data Warehouse Center to load data into the OLAP Server database:

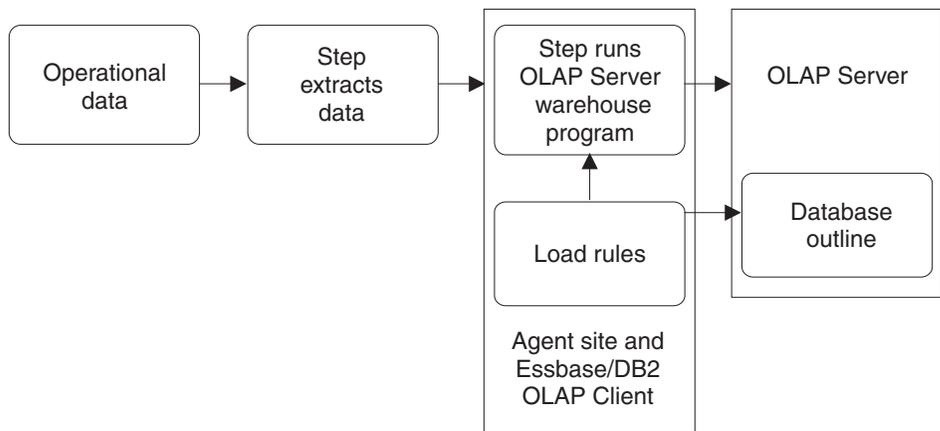
1. Using the Essbase Application Manager, create the OLAP Server application and database. Make note of the application name, the database name, the user ID, and the password. You will need this information as input to a warehouse program.
2. Using the Essbase Application Manager, define the outline 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.

Refreshing an OLAP Server database

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 warehouse program for Hyperion Essbase, such as DB2 OLAP: 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.

The following picture shows the 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.

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.

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.

To load data from a file 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.

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.

To update an OLAP Server outline form 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.

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.

Viewing run-time errors using the basic logging function

Use the basic logging function to view run-time errors.

Procedure:

To view run-time (step processing) 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. Click **Log** to open the Log Viewer window and display the run-time errors for the selected step.

Data Warehouse Center logging and trace data

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\sqllib\logging directory, so that they will not contain obsolete data.

Data Warehouse Center logging and trace data

The default value of *VWS_LOGGING* is:

Windows® NT, Windows 2000, Windows XP

x:\program files\sqllib\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

Data Warehouse Center logging and trace data

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.

Running warehouse agents as a user process (Windows NT, Windows 2000, Windows XP)

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 NT, Windows 2000, or Windows XP **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 twice.
 - 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 executing 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 executing the program and is specified as a system variable.

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.

Remember to turn the trace level back to 0 when you are finished, otherwise performance will degrade.

You can run an agent trace independently for individual steps by setting the trace level in the the step 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 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

Data Warehouse Center logging and trace data

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. If you want to do this, append 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 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
- Q SQL code

Tracing errors created by the Apply program

You can trace errors that are generated 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 don't see any data in the CD table, then the Capture program has not been started, or you have not created 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 logs are:

IWH2LOGC.LOG

When the logger is not running, processes will write messages to this file. The Data Warehouse Center server and the OLE server write to this file. The file will exist 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 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. This is important in situations where DB2[®] is not starting or the file system is full. If you hear five beeps when the logger stops, you should look in this file. The server cannot start if the logger cannot start.

IWH2SERV.LOG

The server log contains the startup message, and will grow 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
- CWM XML metadata to Data Warehouse Center metadata
- ERwin metadata to Data Warehouse Center metadata
- Trillium Software System 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. The metadata mappings shown here are used when publishing data from the OLAP Integration Server to the Information Catalog Center.

DB2 OLAP Integration Server metadata	Data Warehouse Center metadata tag language
N/A	SubjectArea – OLAP Cubes
OLAP cube name This property value is the 4-part name in the following format: server.application.database.outline	Process name
OLAP cube name This property value is the 4-part name in the following format: server.application.database.outline	Step name
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.

Metadata mappings

<p>This property is based on a query of the DB2 OLAP Integration Server catalog (OM_INFO table) based on the following columns:</p> <p>Columns MODEL_NAME and MODEL_ID get MODEL_DATA_SOURCE</p>	DatabaseName
<p>This property is based on a query of the DB2 OLAP Integration Server catalog (OA_INFO table) based on the following columns:</p> <p>MODEL_NAME and MODEL_ID get PHYSICAL_TABLE</p> <p>Only the name on the right side of the period (without the schema) is used.</p>	TablePhysicalName
<p>OLAP cube name</p> <p>This property value is the 4-part name in the following format: server.application.database.outline</p>	Cube name
N/A	StepViewVWPOutputTable
<p>OLAP cube name</p> <p>This property value is the 4-part name in the following format: server.application.database.outline</p>	Step name
N/A	VWPProgramTemplate – Build OLAP Cube
N/A	VWPProgramTemplateParameter
N/A	VWPProgramInstance
N/A	VWPProgramInstanceParameter
DB2 OLAP Integration Server bat file name	VWPProgramInstanceParameterName
N/A	VWP Group – OLAP Integration

Mapping ERwin object attributes to Data Warehouse Center tags

The following tables show how the main ERwin object attributes correspond to the Data Warehouse Center tags:

This table shows the Database - 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 database shows the Table-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-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 Metadata Extract Program” on page 219

Related tasks:

Metadata mappings

- “Creating tag language files for the IBM ERwin Metadata Extract program” on page 220

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.

DDL data types	Warehouse data type
ASCII CHARACTER	CHARACTER(<i>n</i>)
ASCII NUMERIC	
EBCDIC CHARACTER	
EBCDIC NUMERIC	
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 metadata” on page 196

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.

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
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.

Data Warehouse Center properties	Equivalent CWM XML properties
Description or Notes	Description
Administrator	ResponsibleParty

Metadata mappings

Appendix B. Using Classic Connect with the Data Warehouse Center

You can use Classic Connect with the Data Warehouse Center to access data in IMS and VSAM databases.

What is Classic Connect?

Classic Connect provides read access to nonrelational data stored in Information Management Systems (IMS) databases and Virtual Storage Access Method (VSAM) data sets on z/OS™. It provides communication, data access, and data mapping functions so you can read nonrelational data using relational queries.

This chapter contains the following sections:

- What does it do?
- Which data sources does it access
- How is it used
- What are its components

What does it do?

Classic Connect allows you to access nonrelational data by issuing a standard SQL query from a Data Warehouse Center step. You access the data just as if the data was in a DB2® database.

Which data sources does it access?

Classic Connect provides read-only relational access to IMS™ databases and VSAM data sets. It creates a logical, relational database, complete with logical tables that are mapped to actual data in IMS or VSAM databases. Specifically:

- For VSAM, each logical table corresponds to a VSAM data set. Each field of the data set maps to a column of the logical table; each record of the data set maps to a row. Classic Connect can read data from VSAM KSDS, RRDS, and ESDS data sets.
- For IMS, each logical table corresponds to one or more segments of a path in an IMS full-function database. The fields of multiple segments in a path correspond to the columns of a logical table. Each unique set of segment instances for the given path corresponds to a row in a logical table.

Using this relational structure, Classic Connect interprets relational queries that are submitted by users against IMS databases and VSAM data sets.

Using Classic Connect with the Data Warehouse Center

You can define multiple logical databases for a single data source (such as a set of VSAM data sets or a single IMS database). Multiple logical tables can be defined in a logical database.

You can define multiple logical tables for a single data entity (such as a VSAM data set or an IMS segment). For example, a single VSAM data set can have multiple logical tables defined for it, each one mapping data in a new way.

How is it used?

Use Classic Connect with the Data Warehouse Center if your data warehouse uses operational data in an IMS or VSAM database. Use Classic Connect to map the nonrelational data into a pseudo-relational format. Then use the CROSS ACCESS ODBC driver to access the pseudo-relational data. You can then define an IMS or VSAM warehouse source in the Data Warehouse Center that corresponds to the pseudo-relational data.

If you are using the z/OS warehouse agent, this Classic Connect configuration does not apply.

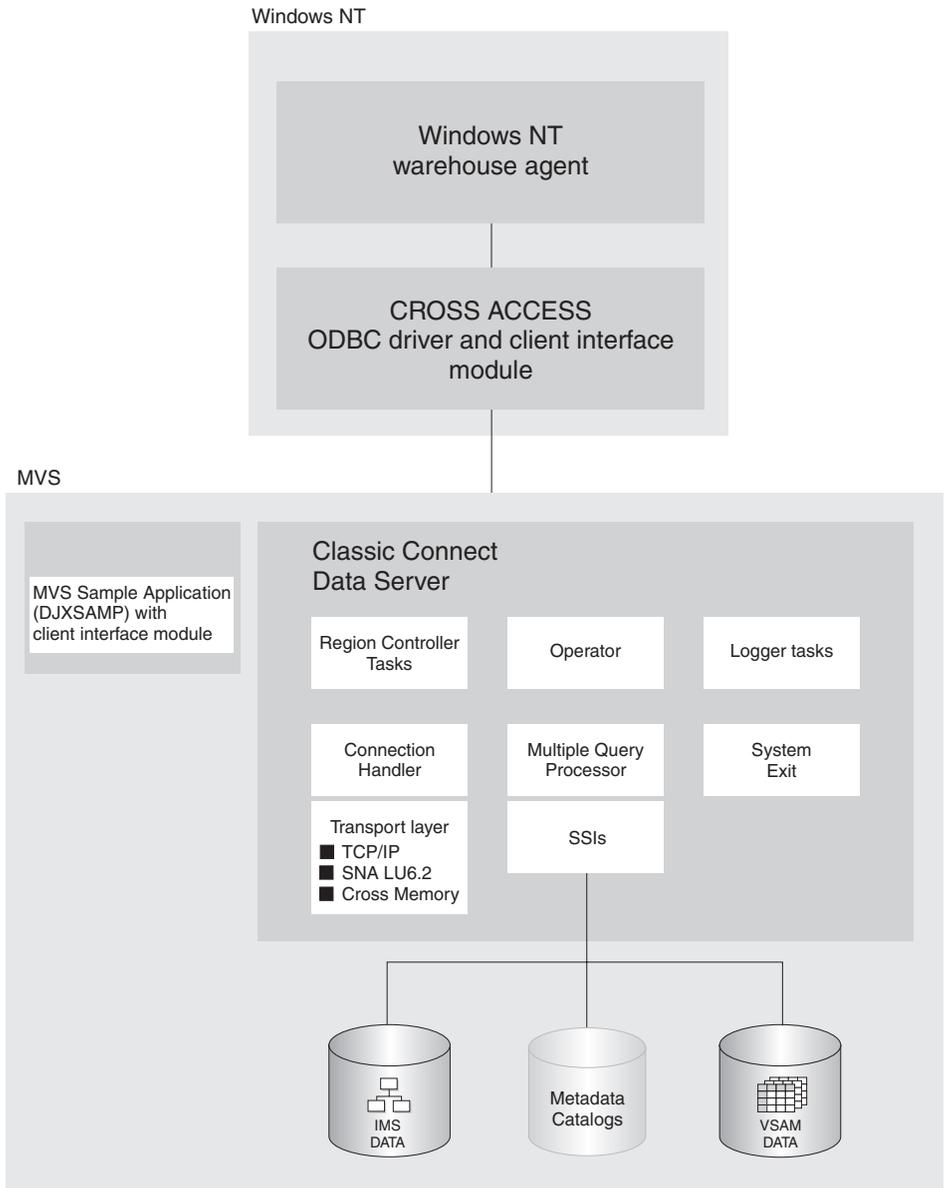
What are its components?

Using Classic Connect with the Data Warehouse Center consists of the following major components:

- Warehouse agents
- CROSS ACCESS ODBC driver
- Classic Connect data server
- Enterprise server
- Data mapper

The following diagram shows how Classic Connect and its components fit into the overall Data Warehouse Center architecture.

Using Classic Connect with the Data Warehouse Center



Warehouse agents

Warehouse agents manage the flow of data between the data sources and the target warehouses. The warehouse agents use the CROSS ACCESS ODBC Driver to communicate with Classic Connect.

Using Classic Connect with the Data Warehouse Center

CROSS ACCESS ODBC driver

The Open Database Connectivity (ODBC) interface enables applications to use Structured Query Language (SQL) statements to access data in relational and nonrelational database management systems.>

The ODBC architecture consists of four components:

- The ODBC-compliant application performs processing and calls the ODBC functions to submit SQL statements and retrieve results.
- The Driver Manager loads drivers on behalf of an application.
- The driver processes ODBC function calls, submits SQL requests to a specific data source, and returns results to the application.
- The data source consists of the data the user wants to access. The data source name is equivalent to the data source name in the CROSS ACCESS data server configuration file.

The Driver Manager and the driver act as one unit that processes ODBC function calls.

Classic Connect data server

All data access is performed by Classic Connect data servers. A data server is responsible for the following functions:

- Accepting SQL queries from the Data Warehouse Center.
- Determining the type of data to be accessed.
- Rewriting the SQL query into the native file or database access language needed. A single SQL access could translate into multiple native data requests.
- Optimizing queries based on generic SQL query rewrite and file- or database-specific optimization.
- Querying multiple data sources for JOINS.
- Translating result sets into a consistent relational format, which involves restructuring nonrelational data into columns and rows.
- Sorting result sets as needed; for example, ORDER BY.
- Issuing all client catalog queries to the Classic Connect metadata catalog.

A Classic Connect data server accepts connection requests from the CROSS ACCESS ODBC driver and the sample application on z/OS.

There are five types of services that can run in the data server:

- Region controller services, which include an MTO operator interface
- Initialization services
- Connection handler services
- Query processor services
- Logger services

Using Classic Connect with the Data Warehouse Center

Region controller services: The primary component of a data server is the region controller. The region controller is responsible for starting, stopping, and monitoring all of the other components of the data server. These different components are referred to as services. The services are implemented as individual load modules running as separate z/OS tasks within the data server address space. Services can have multiple instances, and each instance can support multiple users.

The region controller determines which services to start based on SERVICE INFO ENTRY parameter settings.

Included in the region controller service is the z/OS Master Terminal Operator (MTO) interface, which allows you to display and control the services and users that are being serviced by a data server. Using the MTO interface, you can also dynamically configure the data server.

Initialization services: Initialization services are special tasks that are used to initialize and terminate different types of interfaces to underlying database management systems or z/OS system components. Currently, three initialization services are provided:

IMS BMP/DBB initialization service

used to initialize the IMS region controller to access IMS data using a BMP/DBB interface

IMS DRA initialization service

used to initialize the Classic Connect DRA interface and to connect to an IMS DBCTL region to access IMS data using the DRA interface

WLM initialization service

used to initialize and register with the z/OS Workload Manager subsystem (using the WLM System Exit). This allows individual queries to be processed in WLM goal mode.

Connection handler services: A connection handler (CH) service task is responsible for listening for connection requests from the Data Warehouse Center. Connection requests are routed to the appropriate query processor task for subsequent processing.

Classic Connect supplies three typical transport layer modules that can be loaded by the CH task:

- TCP/IP
- SNA LU 6.2
- z/OS cross memory services.

The z/OS client application, DJXSAMP, can connect to a data server using any of these methods; however, the recommended approach for local clients is to

Using Classic Connect with the Data Warehouse Center

use z/OS cross memory services. The Data Warehouse Center can use either TCP/IP or SNA to communicate with a remote data server.

Query processor services: The query processor is the component of the data server that is responsible for translating client SQL into database- and file-specific data access requests. The query processor treats IMS and VSAM data as a single data source and is capable of processing SQL statements that access either IMS, VSAM, or both. Multiple query processors can be used to separately control configuration parameters, such as those that affect tracing and governors, to meet the needs of individual applications.

The query processor can service SELECT statements. The query processor invokes one or more subsystem interfaces (SSIs) to access the target database or file system referenced in an SQL request. The following SSIs are supported:

IMS BMP/DBB interface

Allows IMS data to be accessed through an IMS region controller. The region controller is restricted to a single PSB for the data server, limiting the number of concurrent users a data server can handle.

IMS DRA interface

Allows IMS data to be accessed using the IMS DRA interface. The DRA interface supports multiple PSBs and is the only way to support a large number of users. This is the recommended interface.

VSAM interface

Allows access to VSAM ESDS, KSDS, or RRDS files. This interface also supports use of alternate indexes.

Classic Connect supplies a RUNSTATS utility program that is used to update population statistics for the logical tables and their associated indexes and keys. This information can be used by the query processor to optimize JOINS.

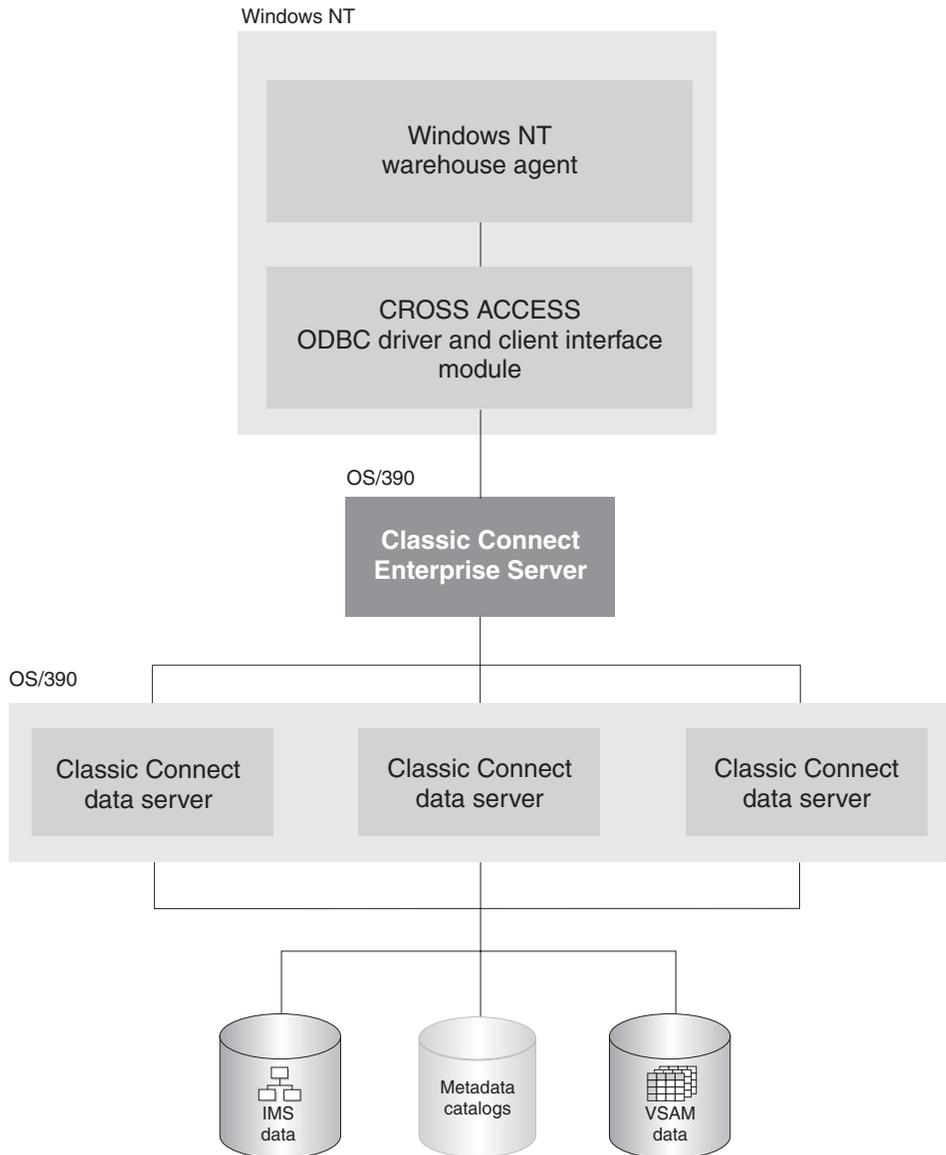
Logger service: A logger service is a task that is used for system monitoring and trouble shooting. A single logger task can be running within a data server. During normal operations, you will not need to be concerned with the logger service.

Enterprise server

The enterprise server is an optional component that you can use to manage a large number of concurrent users across multiple data sources. An enterprise server contains the same tasks that a data server uses, with the exception of the query processor and the initialization services.

The following diagram shows how the Classic Connect enterprise server fits into a Classic Connect configuration:

Using Classic Connect with the Data Warehouse Center



Like a data server, the enterprise server's connection handler is responsible for listening for client connection requests. However, when a connection request is received, the enterprise server does not forward the request to a query processor task for processing. Instead, the connection request is forwarded to a data source handler (DSH) and then to a data server for processing. The enterprise server maintains the end-to-end connection between the client application and the target data server. It is responsible for sending and receiving messages between the client application and the data server.

Using Classic Connect with the Data Warehouse Center

The enterprise server is also used to perform load balancing. Using configuration parameters, the enterprise server determines the locations of the data servers that it will be communicating with and whether those data servers are running on the same platform as the enterprise server.

The enterprise server can automatically start a local data server if there are no instances active. It can also start additional instances of a local data server when the currently active instances have reached the maximum number of concurrent users they can service, or the currently active instances are all busy.

Data mapper

The Classic Connect nonrelational data mapper is a Microsoft® Windows-based application that automates many of the tasks required to create logical table definitions for nonrelational data structures. The objective is to view a single file or portion of a file as one or more relational tables. The mapping must be accomplished while maintaining the structural integrity of the underlying database or file.

The data mapper interprets existing physical data definitions that define both the content and the structure of nonrelational data. The tool is designed to minimize administrative work, using a definition-by-default approach.

The data mapper accomplishes the creation of logical table definitions for nonrelational data structures by creating *metadata grammar* from existing nonrelational data definitions (COBOL copybooks). The metadata grammar is used as input to the Classic Connect metadata utility to create a metadata catalog that defines how the nonrelational data structure is mapped to an equivalent logical table. The metadata catalogs are used by query processor tasks to facilitate both the access and translation of the data from the nonrelational data structure into relational result sets.

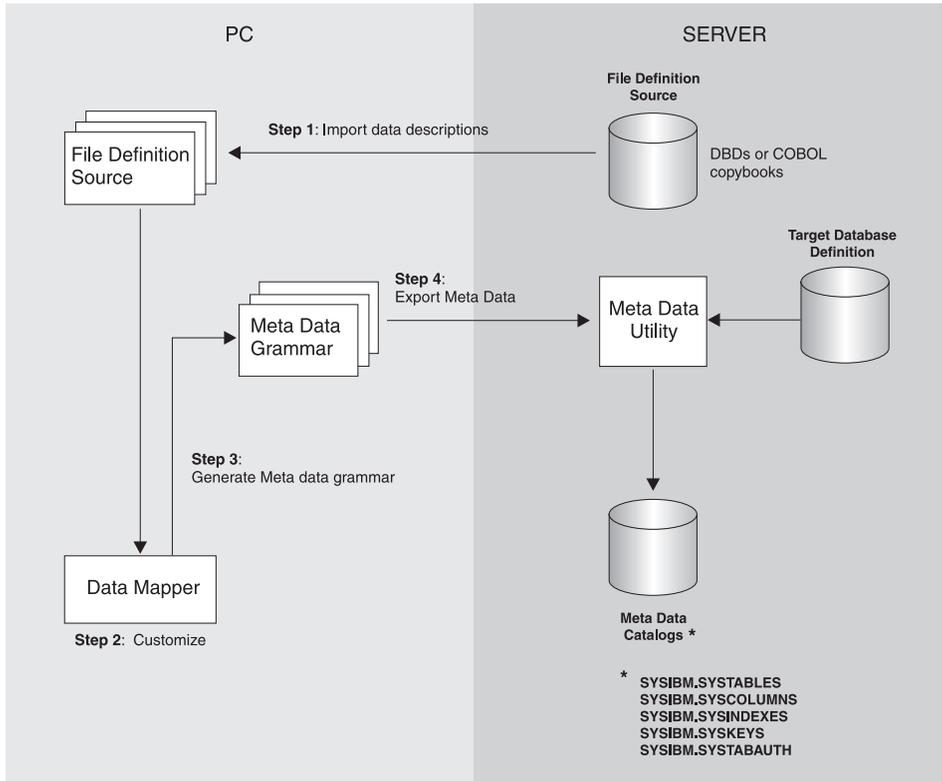
The data mapper import utilities create initial logical tables from COBOL copybooks. A visual point-and-click environment is used to refine these initial logical tables to match site- and user-specific requirements. You can utilize the initial table definitions automatically created by data mapper, or customize those definitions as needed.

Multiple logical tables can be created that map to a single physical file or database. For example, a site may choose to create multiple table definitions that all map to an employee VSAM file. One table is used by department managers who need access to information about the employees in their departments; another table is used by HR managers who have access to all employee information; another table is used by HR clerks who have access to information that is not considered confidential; and still another table is used by the employees themselves who can query information about their own

Using Classic Connect with the Data Warehouse Center

benefits structure. Customizing these table definitions to the needs of the user is not only beneficial to the end-user but recommended.

The following diagram shows the data administration workflow with data mapper.



The data mapper contains embedded FTP support to facilitate file transfer from and to the mainframe.

The steps in the Data mapper workflow diagram are described as follows:

1. Import existing descriptions of your nonrelational data into data mapper. COBOL copybooks and IMS database definitions (DBDs) can all be imported into the data mapper.

The data mapper creates default logical table definitions from the COBOL copybook information. If these default table definitions are acceptable, skip the following step and go directly to step 3.

Using Classic Connect with the Data Warehouse Center

2. Refine or customize the default table definitions as needed by the users. For example, importing the record layout for the VSAM customer master file creates the default Customer_Table. Two additional tables can also be created from the original:
 - Marketing_Customer_Table, which contains only those data items required by the marketing department
 - Service_Customer_Table, which contains only those data items required by support representatives
3. Generate the actual metadata grammar that will be used by the metadata utility on z/OS.
4. Export the logical table definitions to the mainframe where the database or file resides. These definitions are then used as input to the metadata utilities, which create the metadata catalogs.

After completing these steps, you are ready to use Classic Connect operational components with your tools and applications to access your nonrelational data.

Requirements for setting up integration between Classic Connect and the Data Warehouse Center

This topic describes the requirements for setting up integration between Classic Connect and the Data Warehouse Center.

Hardware and software requirements

The integration requires the following software:

- DB2[®] Relational Connect Classic Connect
- DB2 Universal Database[™] Version 8

Optionally, you can use the DB2 Relational Connect Classic Connect data mapper to generate metadata grammar for you.

Installing and configuring prerequisite products

Complete the tasks summarized in the following table to set up integration between Classic Connect and the Data Warehouse Center.

Task	Content
Learning about the integration	What is Classic Connect? Concepts and terminology

Using Classic Connect with the Data Warehouse Center

Installing and configuring the data server	System requirements and planning
	Installing Classic Connect on z/OS™
	Data server installation and verification procedure
	Introduction to data server setup
	Configuring communication protocols between z/OS and Windows® NT, Windows 2000, or Windows XP
Installing and configuring the client workstation	Configuring a Windows NT, Windows 2000, or Windows XP client
	Defining an agent site
Using an IMS™ or VSAM warehouse source	Mapping nonrelational data and creating queries
	Optimization
	Defining a warehouse source

Configuring communications protocols between z/OS, Windows NT, Windows 2000 and Windows XP

This section describes how to configure the communications protocols between z/OS, Windows NT, Windows 2000 and Windows XP.

Communications protocols between z/OS and Windows NT, Windows 2000, or Windows XP

Classic Connect supports the TCP/IP and SNA LU 6.2 (APPC) communication protocols to establish communication between a Visual Warehouse™ agent and Classic Connect data servers. A third protocol, cross memory, is used for local client communication on z/OS.

This chapter describes modifications you need to make to the TCP/IP and SNA communications protocols before you can configure Classic Connect and includes the following sections:

- Communications options
- Configuring the TCP/IP communications protocol
- Configuring the LU 6.2 communications protocol

Communications options

Classic Connect supports the following communications options:

- Cross memory
- SNA

Using Classic Connect with the Data Warehouse Center

- TCP/IP

Cross memory: Cross memory should be used to configure the local z/OS™ client application (DJXSAMP) to access a data server. Unlike for SNA and TCP/IP, there are no setup requirements to use the z/OS cross memory interface. This interface uses z/OS data spaces and z/OS token naming services for communications between client applications and data servers.

Each cross memory data space can support up to 400® concurrent users, although in practice this number may be lower due to resource limitations. To support more than 400 users on a data server, configure multiple connection handler services, each with a different data space name.

The following example illustrates the communications compound address field:

```
XM1/DataSpace/Queue
```

Because you don't need to modify any cross memory configuration settings, this protocol is not discussed in detail here.

SNA: SNA is a more sophisticated protocol and supports hardware compression, which can greatly reduce the amount of data actually transmitted over the wire. Unfortunately, the infrastructure requirements and setup time for using SNA are generally more expensive than using TCP/IP.

TCP/IP: A single TCP/IP connection handler service can service a maximum of 255 concurrent users. Depending on your particular TCP/IP subsystem, further limitations might apply.

Multiple sessions are created on the specified port number. The number of sessions carried over the port is the number of concurrent users to be supported plus one for the listen session the connection handler uses to accept connections from remote clients. If the TCP/IP implementation you are using requires you to specify the number of sessions that can be carried over a single port, you must ensure that the proper number of sessions have been defined. Failure to do so will result in a connection failure when a client application attempts to connect to the data server.

Configuring the TCP/IP communications protocol

This section describes the steps you must perform, both on your z/OS system and on your Windows NT, Windows 2000, or Windows XP system, to configure a TCP/IP communications interface for Classic Connect. This section also includes a TCP/IP planning template and worksheet that are designed to illustrate TCP/IP parameter relationships.

TCP/IP CIs that function with Classic Connect

There are two types of TCP/IP CIs that function with Classic Connect: IBM's TCP/IP and Berkeley Sockets. Your configuration may differ depending on which type of TCP/IP CI you are using. If your site does not use TCP/IP, proceed to Configuring the LU 6.2 communications protocol.

Both interfaces allow Classic Connect to communicate with the z/OS™ TCP/IP stack. Berkeley Sockets allows you to use the host and service name, where IBM's TCP/IP requires a numeric IP address and port number. Berkeley Sockets can use a local host file instead of calling DNS. However, both Berkeley Sockets and IBM's TCP/IP require a TCP/IP address space name.

Configuring TCP/IP on z/OS for Classic Connect

Classic Connect's TCP/IP is compatible with both IBM's and Interlink's Berkeley Socket TCP/IP. This section describes how to configure Classic Connect using IBM's TCP/IP.

Berkeley sockets is supported by IBM and Interlink. The Berkeley Sockets version requires an additional parameter in the DJXDSCF member called TASK PARAMETER, which identifies the Interlink subsystem name and identifies the location of the configuration data sets for IBM. Within the configuration data sets, users must specify the name of the started-task procedure used to start the TCP/IP address space name and can also specify the TCP/IP DNS IP addresses. If no environment variables are passed, then the default value TCPIP is used for both the address space name and as the high-level qualifier (hlq) of the standard configuration files:

- hlq.TCPIP.DATA
- hlq.ETC.HOSTS
- hlq.ETC.PROTOCOLS
- hlq.ETC.SERVICES
- hlq.ETC.RESOLV.CONF

Classic Connect uses a search order to locate the data sets, regardless of whether Classic Connect sets the hlq or not.

Procedure:

To configure TCP/IP on Windows NT, Windows 2000, or Windows XP systems:

1. Obtain either the hostname or IP address of the z/OS system.

Using a hostname requires the availability of a configured local HOSTS file or a domain name server. If a domain name server is involved, then there is some overhead required to resolve the HOST name to the correct IP

Using Classic Connect with the Data Warehouse Center

address. However, it is recommended that you use hostnames in remote client configuration files for readability and ease of future configuration changes.

Using hostnames also makes it easier to change IP addresses if the environment changes. If hostnames are used, frequently the data server/remote clients will not have to be reconfigured. Classic Connect can be brought down, and the network administrator can change the IP address for a hostname in the z/OS and client domain name server(s). When the data server is restarted it will automatically listen on the new IP address for connection requests from remote clients. When a remote client connects to the data server it will automatically use the new IP address that has been assigned to the hostname without a change to the Classic Connect configuration files.

For IBM's TCP/IP, determine the IP address or hostname of the host computer on which Classic Connect is being installed. If you are running your z/OS TCP/IP on off-loaded devices, specify the address of the TCP/IP stack on the z/OS image, not the address of an off-load gateway's IP stack.

2. Obtain a unique port (socket) number greater than 1024 for each data server that will be accessed from a client.

The port number cannot be the same as any port that is already defined for use by any other application, including other Classic Connect data servers on the same z/OS system. Using a non-unique port number causes the data server to fail at start up. To determine if a port number has already been assigned to another application, issue the following command from the Spool Display and Search Facility (SDSF) log:

```
TSO NETSTAT SOCKETS
```

Because some sites restrict the use of certain port numbers to specific applications, you should also contact your network administrator to determine if the port number you've selected is unique and valid.

Optionally, you can substitute the service name assigned to the port number defined to your system.

Service names, addresses, and tuning values for IBM's TCP/IP are contained in a series of data sets:

- hlq.TCPIP.DATA
- hlq.ETC.HOSTS
- hlq.ETC.PROTOCOLS
- hlq.ETC.SERVICES
- hlq.ETC.RESOLV.CONF

Using Classic Connect with the Data Warehouse Center

where "hlq" represents the high-level qualifier of these data sets. You can either accept the default high-level qualifier, TCPIP, or you can define a high-level qualifier specifically for Classic Connect.

When you have determined these values, use the TCP/IP communications template and worksheet to complete the z/OS configuration of your TCP/IP communications.

Configuring TCP/IP on Windows NT, Windows 2000, or Windows XP systems

You must configure your Windows NT, Windows 2000, or Windows XP machine to locate the data server on z/OS.

Procedure:

1. Resolve the host address on the client.

If you are using the IP address in the client configuration file, you can skip this step.

The client workstation must know the address of the host server to which it is attempting to connect. There are two ways to resolve the address of the host:

- By using a name server on your network. This is the recommended approach. See your TCP/IP documentation for information about configuring TCP/IP to use a name server.

If you are already using a name server on your network, proceed to step 2.

- By specifying the host address in the local HOSTS file. On a Windows NT, Windows 2000, or Windows XP client, the HOSTS file is located in the %SYSTEMROOT%\SYSTEM32\DRIVERS\ETC directory.

Add an entry to the HOSTS file on the client for the server's hostname as follows:

```
9.112.46.200 stplex4a # host address for Classic Connect
```

where 9.112.46.200 is the IP address, and stplex4a is the HOSTNAME. If the server resides in the same Internet domain as the client, this name can be a flat hostname. If the server is not in the same domain, the name must be a fully-specified domain name, such as stplex4a.stl.ibm.com, where stl.ibm.com is an example of a domain name.

Notes:

- a. You must end the last line with a comment (# comment) or press Enter at the end of that line to insert an end-of-line character.

Using Classic Connect with the Data Warehouse Center

- b. You should refer to the documentation from your TCP/IP product for specific information about resolving host addresses.
2. Update the SERVICES file on the client.

If you are using the port number in the client configuration file, you can skip this step.

The following information must be added to the SERVICES file on the client for TCP/IP support:

```
ccdatser 3333 # CC data server on stplex4a
```

The SERVICES file is located in the %SYSTEMROOT%\SYSTEM32\DRIVERS\ETC directory.

TCP/IP communications template and worksheet

The left side of the following picture provides you with an example set of TCP/IP values for a z/OS™ configuration; these values will be used during data server and client configuration in a later step. Use the right side of the figure as a template in which to enter your own values.

TCP/IP-MVS	
MVS hostname or IP address	not used *
subsystem name	9.112.46.200
port number or service name for data server	tcp
	3333

* This example uses an IP address rather than a hostname

TCP/IP-MVS	
MVS hostname or IP address	
subsystem name	
port number or service name for data server	

LU 6.2 communications protocol

This section describes the values you must determine and steps you must perform, both on your z/OS™ system and on your Windows® NT, Windows 2000, or Windows XP system, to configure LU 6.2 (SNA/APPC) communications for Classic Connect.

Requirement:

For APPC connectivity between Classic Connect and DB2® Relational Connect for Windows NT, Windows 2000, or Windows XP, you need Microsoft® SNA Server Version 3.0 with service pack 3 or later.

The information in this section is specific to Microsoft SNA Server Version 3.0. For more information about configuring Microsoft SNA Server profiles, see the appropriate product documentation. This section also includes a communications template and worksheet that are designed to clarify LU 6.2

Using Classic Connect with the Data Warehouse Center

parameter relationships on z/OS and Windows NT, Windows 2000, and Windows XP and assist you with your LU 6.2 configuration.

Configuring LU 6.2 on z/OS: If you will be using LU 6.2 to access Classic Connect from DB2 Relational Connect, you need to configure VTAM® table definitions on your z/OS system, which include:

- A Mode table entry
- An Application ID

The Application ID must be unique for a data server. Using a non-unique value causes the data server to fail at start up.

Unlike TCP/IP, you can specify the packet size for data traveling through the transport layer of an SNA network. However, this decision should be made by network administrators because it involves the consideration of complex routes and machine/node capabilities. In general, the larger the bandwidth of the communications media, or pipe, the larger the RU size should be.

Configuring LU 6.2 on a Windows NT, Windows 2000, or Windows XP system: This section explains the values you need to configure to use the SNA LU 6.2 protocol with your Windows NT® client.

For each Windows NT, Windows 2000, or Windows XP system, configure the following values:

- SNA Server Properties Profile
- Connections for the SNA Server

This example assumes you have installed a SNA DLC 802.2 link service. See your network administrator to obtain local and remote node information.

- Local APPC LU Profile

The LU name and network must match the local node values in the Connections for the SNA Server profile. The LU must be an *Independent* LU type.

- Remote APPC LU Profile

There must be one of these profiles for each Classic Connect data server or enterprise server that will be accessed. It must support parallel sessions. It will match the Application ID configured in VTAM table definitions on z/OS.

- APPC Mode

See the mode, CX62R4K in the LU 6.2 Configuration Template. It has a maximum RU size of 4096.

- CPIC Symbolic Name

Using Classic Connect with the Data Warehouse Center

There must be one of these for each Classic Connect data server or enterprise server that will be accessed. The TP Name referenced in this profile must be unique to the Classic Connect data server or enterprise server on a given z/OS system.

After you have entered these values, save the configuration, and stop and restart your SNA Server. When the SNA Server and the Connection (in this example, OTTER and SNA z/OS respectively), are 'Active,' the connection is ready to test with an application.

APPC communications template and worksheet: The following figure, LU 6.2 Configuration Template, which provides you with an example set of VTAM and SNA values, is included for reference. Use the figure, LU 6.2 Configuration Worksheet, which is a duplicate of figure LU 6.2 Configuration Template without the supplied values, as a worksheet in which you can enter the VTAM and SNA values specific to your LU 6.2 configuration. You will need the values you enter in this worksheet to complete configuration steps in subsequent chapters.

Using Classic Connect with the Data Warehouse Center

SNA Server

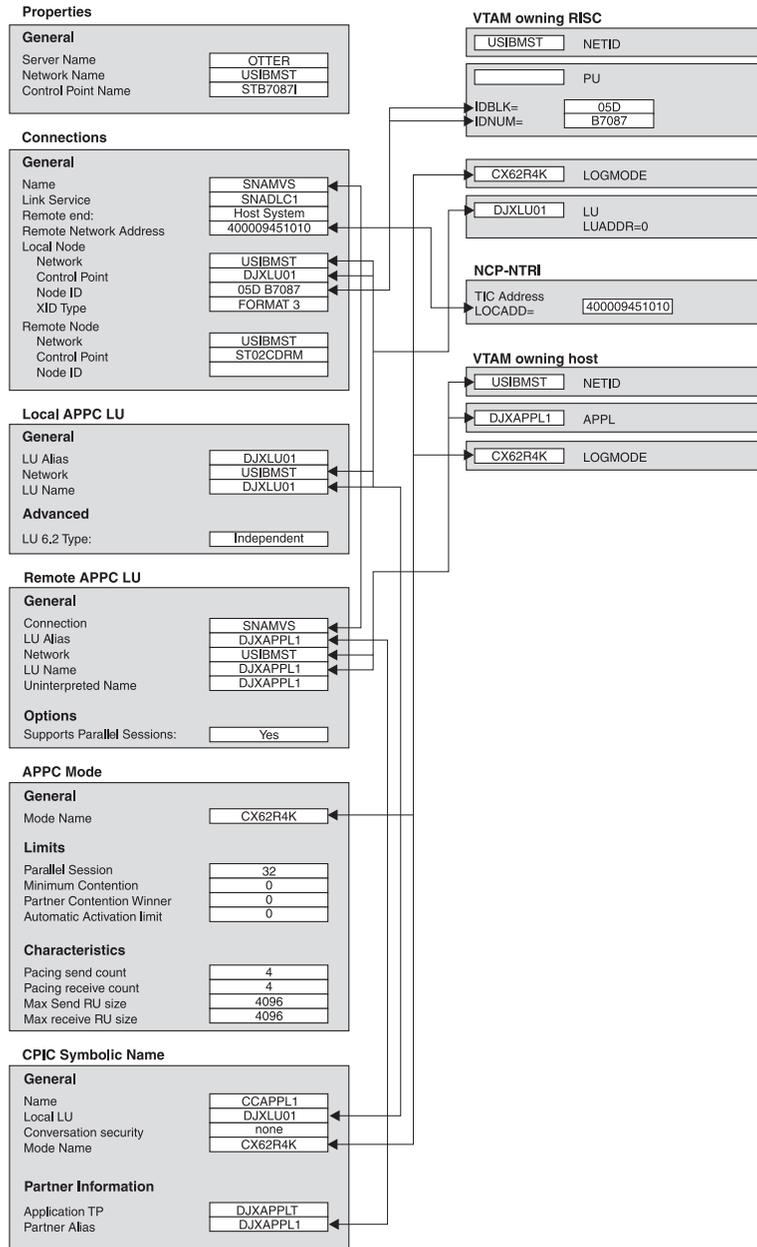


Figure 5. LU 6.2 Configuration Template

Using Classic Connect with the Data Warehouse Center

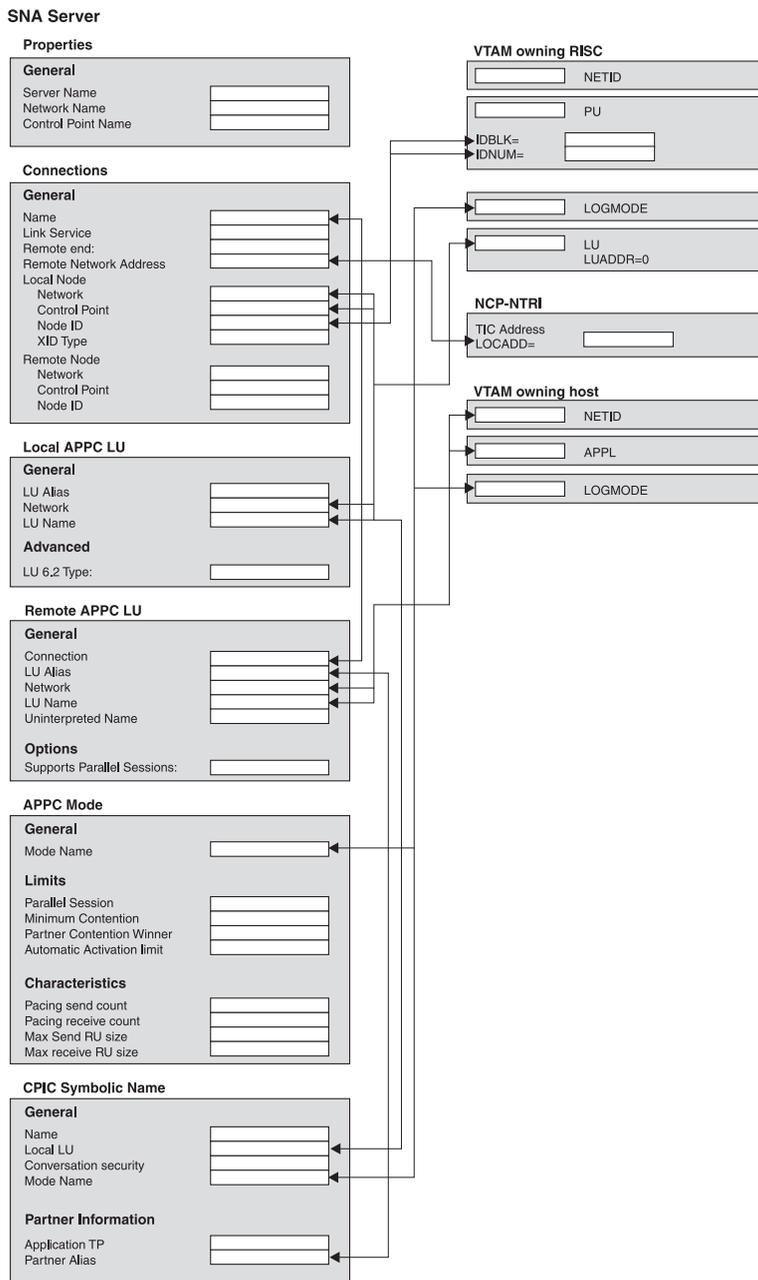


Figure 6. LU 6.2 Configuration Worksheet

Configuring a Windows NT, Windows 2000, Windows XP client

This section describes how to install the Classic Connect Drivers component, and the CROSS ACCESS ODBC driver, and how to use the CROSS ACCESS ODBC driver to configure data sources.

Installing the Classic Connect Drivers component (Windows NT, Windows 2000, Windows XP)

Follow the steps in this task to install the Classic Connect ODBC driver.

Procedure:

Installing the Classic Connect ODBC driver:

1. Insert the DB2 Warehouse Manager CD-ROM into your CD-ROM drive. The auto-run feature automatically starts the DB2 Setup Launchpad.
2. Click **Install Products** from the launchpad.
3. Ensure that **DB2 Warehouse Manager** is selected, then click **Next**.
4. Proceed as prompted by the DB2 Setup Wizard.
5. Select the **Classic Connect Drivers**, then click **Next**.
6. Proceed as prompted by the DB2 Setup Wizard. Online help is available to guide you through the remaining steps.

Installing the CROSS ACCESS ODBC driver

This topic describes how to install the CROSS ACCESS ODBC driver.

Procedure:

Install the CROSS ACCESS ODBC driver by performing a custom install of the DB2 Warehouse Manager Version 7, and selecting the Classic Connect Drivers component. The driver is not installed as part of a typical installation of the DB2 Warehouse Manager.

The CROSS ACCESS ODBC driver will be installed in the ODBC32 subdirectory of the SQLLIB directory. After the installation is complete, you must manually add the path for the driver (for example, C:\Program Files\SQLLIB\ODBC32) to the PATH system environment variable. If you have another version of the CROSS ACCESS ODBC driver already installed, place the ... \SQLLIB\ODBC32\ path before the path for the other version. The operating system will use the first directory in the path that contains the CROSS ACCESS ODBC driver.

Configuring data sources

This section describes how to configure data sources.

Using Classic Connect with the Data Warehouse Center

Data source configuration

CROSS ACCESS ODBC data sources are registered and configured using the ODBC Administrator. Configuration parameters unique to each data source are maintained through this utility.

You can define many data sources on a single system. For example, a single IMS system can have a data source called `MARKETING_INFO` and a data source called `CUSTOMER_INFO`. Each data source name should provide a unique description of the data.

Configuration prerequisites: The following information must be available before you attempt to configure the ODBC driver. If you are missing any of this information, see your system administrator.

- The name of the CROSS ACCESS data source to define in the ODBC Administrator
- If you are using TCP/IP:
 - The IP address for the host system where the data server runs.
 - The port number assigned to the TCP/IP connection handler in the TASK INFO ENTRY parameter of the data server.
- If you are using LU 6.2:
 - The name of the outbound side information record (SIR Outbound) defined in the SNA Server.
 - The data compression level of the host components.

Before you configure the ODBC driver, be sure that the Windows client is set up for the connection handler that you want to use, either TCP/IP or LU 6.2.

For APPC connectivity between Classic Connect and DB2 Relational Connect for Windows NT, you need Microsoft SNA Server Version 3 service pack 3 or later.

Specifying a data source

The data sources that are defined for all the currently installed ODBC drivers are listed in the ODBC Data Source Administrator window. From this window, you can:

- Add and configure data sources.
- Modify the configuration of data sources.
- Delete data sources.

Procedure:

To open the ODBC Data Source Administrator window:

1. Click **Start** on your desktop and click **Settings**.

Using Classic Connect with the Data Warehouse Center

2. Click **Control Panel**.
3. Open the ODBC icon. The ODBC Data Source Administrator window opens.

This window displays a list of data sources and drivers on the System DSN page.

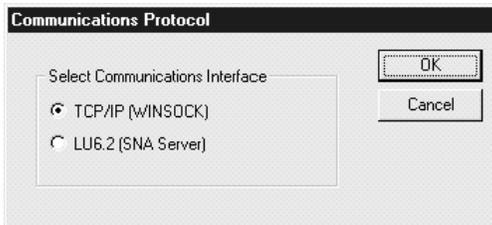
Adding and configuring a data source

This topic describes how to add and configure a data source.

Procedure:

To add and configure a data source:

1. Open the ODBC Data Source Administrator window.
2. On the System DSN page, click **Add**. The Create New Data Source window opens.
3. Click CrossAccess32.
4. Click **Finish**. The CROSS ACCESS Communications Protocol window opens.
5. Select a communications interface to use with the data source that you are configuring.



6. Click **OK**. The CROSS ACCESS ODBC Data Source Configuration window opens.

In this window, you can enter parameters for new data sources or modify parameters for existing data sources. Many of the parameters must match the values specified in the server configuration. If you do not know the settings for these parameters, contact the Classic Connect system administrator.

The parameters that you enter in this window vary depending on whether you are using the TCP/IP or LU 6.2 communications interface.

Using Classic Connect with the Data Warehouse Center

Setting up TCP/IP communications

Configuring TCP/IP communications

Use the CROSS ACCESS ODBC Data Source Configuration window to:

- Name the data source.
- Configure the TCP/IP communications setting.
- Specify the necessary authorizations.

The following figure shows the CROSS ACCESS ODBC Data Source Configuration window for TCP/IP.

The screenshot shows a dialog box titled "CROSS ACCESS ODBC Data Source Configuration". It has two tabs: "General" and "Catalog", with "General" selected. The "Data Source Name" field contains "DJXSAMP" and the "Description" field contains "Classic Connect". Below these is a "Communications Configuration" section with "Host IP Address" set to "xxx.xxx.xxx.xxx" and "Host Port Number" set to "1035". The "Authorizations" section has three checkboxes: "OS Login ID Required" (checked), "DB User ID Required" (unchecked), and "File ID Required" (unchecked). At the bottom of the dialog, there are three buttons: "OK", "Cancel", and "Apply".

Procedure:

To configure TCP/IP communications:

1. Type the name of the data source in the **Data Source Name** field. This name must match Field 2 of the QUERY PROCESSOR SERVICE INFO ENTRY of the data server configuration file.
2. Enter a brief description of the data source in the **Description** field.
3. Type the IP address of the data server in the **Host IP Address** field. This field specifies the IP address, in host-name or dotted-decimal notation, of the host where the data server is installed.

Using Classic Connect with the Data Warehouse Center

4. Type the port number (socket) assigned to the host component TCP/IP communications in the **Host Port Number** field. This number must match Field 10 of the TCP/IP SERVICE INFO ENTRY of the data server configuration file.
5. Select one or more of the following check boxes:
 - **OS Login ID Required.** Select this box to be prompted for a user ID and password to log in to the operating system.
 - **DB User ID Required.** Select this box to be prompted for a user ID and password to log in to the database system, for example, DB2 or Sybase.
 - **File ID Required.** Select this box to be prompted for the file ID and password required to access the database. The file ID and password is required for certain databases, like Model 204.
6. Specify whether the data source has update capabilities. The default is read-only access.

Setting the database catalog options

Use the Catalog page to perform the following tasks:

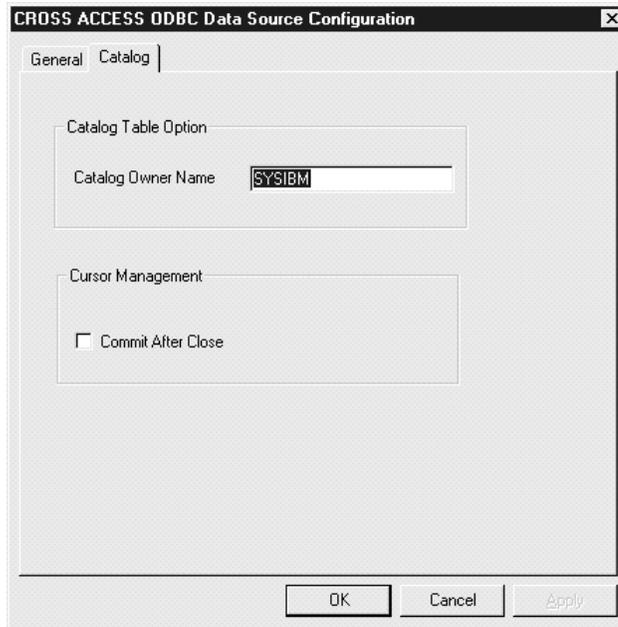
- Specify the catalog table option.
- Specify cursor management.

Procedure:

To set the database catalog options:

1. Click the **Catalog** tab in the CROSS ACCESS ODBC Data Source Configuration window.

Using Classic Connect with the Data Warehouse Center



2. Type the name of the database catalog owner in the **Catalog Owner Name** field.
3. Select the **Commit After Close** check box if you want the ODBC driver to automatically issue a COMMIT call after a CLOSE CURSOR call is issued by the application. On certain database systems, a resource lock will occur for the duration of an open cursor. These locks can be released only by a COMMIT call and a CLOSE CURSOR call.
If you leave this box clear, the cursors are freed without issuing a COMMIT call.
4. Click **OK**.

The TCP/IP communications information is saved.

Setting up LU 6.2 communications

Configuring LU 6.2 communications

Use the CROSS ACCESS ODBC Data Source Configuration window to:

- Identify the data source.
- Configure LU 6.2 communications settings.
- Specify necessary authorizations.

Using Classic Connect with the Data Warehouse Center

The following figure shows the CROSS ACCESS ODBC Data Source Configuration window for LU 6.2.

The screenshot shows the 'CROSS ACCESS ODBC Data Source Configuration' dialog box. It has two tabs: 'General' and 'Catalog'. The 'General' tab is active. It contains the following fields and options:

- Data Source Name**: A text input field.
- Description**: A text input field.
- Communications Configuration**: A section containing a **Side Information Record** text input field.
- Authorizations**: A section containing three checkboxes:
 - OS Login ID Required
 - DB User ID Required
 - File ID Required
- Read Only

At the bottom of the dialog are three buttons: **OK**, **Cancel**, and **Apply**.

Procedure:

To configure LU 6.2 communications:

1. Type the name of the data source in the **Data Source Name** field. This name must match Field 2 of the QUERY PROCESSOR SERVICE INFO ENTRY of the data server configuration file.
2. Enter a brief description of the data source in the **Description** field.
3. Type the side information record (SIR) name in the **Side Information Record** field.

The SIR name refers to a side information record (also called a CPIC symbolic name in the figure LU 6.2 Configuration Worksheet) defined in the SNA server. This SIR must include the configuration parameters that represent the data server.

4. Select one or more of the following check boxes:
 - **OS Login ID Required**. Select this box to be prompted for a user ID and password to log in to the operating system.
 - **DB User ID Required**. Select this box to be prompted for a user ID and password to log in to the database system, for example, DB2 or Sybase.

Using Classic Connect with the Data Warehouse Center

- **File ID Required.** Select this box to be prompted for the file ID and password required to access the database. The file ID and password is required for certain databases, like Model 204.
5. Clear the **Read Only** check box to indicate that the data source has update capabilities. The default is read-only access.

Setting the database catalog options for LU 6.2 communications

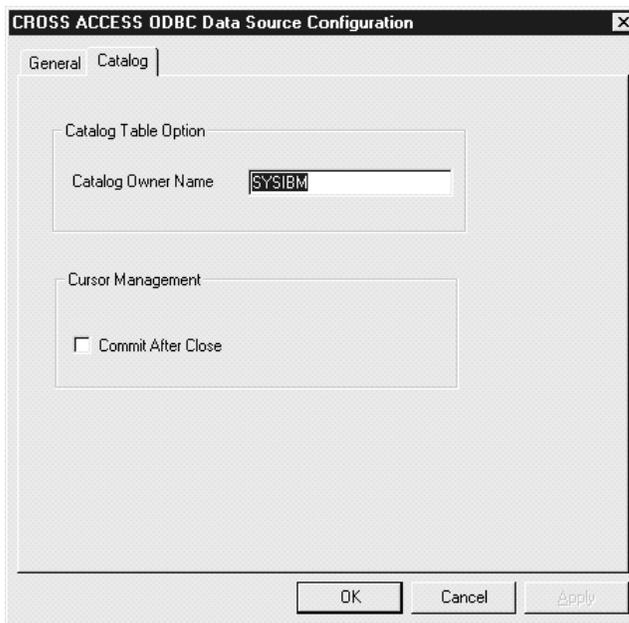
Use the Catalog page to perform the following tasks:

- Specify the catalog table option.
- Specify cursor management.

Procedure:

To set the database catalog options:

1. Click the **Catalog** tab in the Configuration window.



2. Type the name of the database catalog owner in the **Catalog Owner Name** field.
3. Select the **Commit After Close** check box if you want the ODBC driver to automatically issue a COMMIT call after a CLOSE CURSOR call is issued by the application. On certain database systems, a resource lock will occur for the duration of an open cursor. These locks can be released only by a COMMIT call and a CLOSE CURSOR call.

Using Classic Connect with the Data Warehouse Center

If you leave this box clear, the cursors are freed without issuing a COMMIT call.

4. Click **OK**.

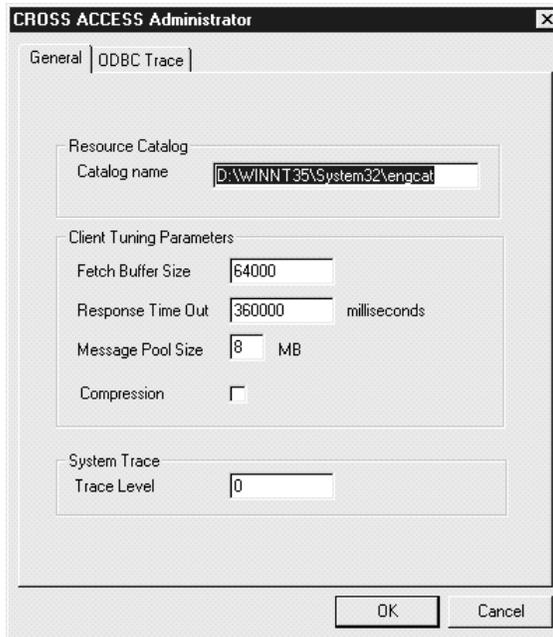
The LU 6.2 communications information is saved.

After you complete the procedure, you need to create the related topics for this task.

Configuring ODBC drivers

The CROSS ACCESS ODBC driver maintains a set of configuration parameters common to all CROSS ACCESS data sources. Configuration of these parameters are performed in the CROSS ACCESS Administrator window. The following steps show how to configure the ODBC driver parameters.

The following figure shows the General page of the CROSS ACCESS Administrator window.



Procedure:

1. From the General page of the CROSS ACCESS Administrator window, type the full path name of the language catalog in the **Catalog name** field. This value is required.

Using Classic Connect with the Data Warehouse Center

The language catalog contains messages in a specific language and is pointed to by a file contained within the CROSS ACCESS configuration files.

2. Optionally, type the size of a CROSS ACCESS fetch buffer in the **Fetch Buffer Size** field.

This value tunes message blocking by controlling the amount of data that is returned in a single fetch request. The system packs as many rows of data as possible into a fetch buffer of the size specified. For example, if the fetch buffer is set to 10,000 bytes and each row is 2,000 bytes, the system can pack 5 rows per fetch request. If a single row is returned but does not fit into the specified buffer, the fetch buffer internally increases to fit the single row of data. To turn off message blocking, set this parameter to 1. The value must be from 1 to 64,000, inclusive. The default value is 10,000.

3. Optionally, type the maximum amount of time that this service will wait for an expected response before terminating a connection in the **Response Time Out** field.

You can specify the following time intervals:

- nMS = number of milliseconds
- nS = number of seconds
- nM = number of minutes

Specify a value between 0 and 1000MS, 60S, or 60M. The default value is 6M.

4. Type the size of the memory used for all memory allocation in the **Message Pool Size** field. This value is required.

Specify the number in bytes. The actual workable maximum value should be set to 2 MB less than the heap size. If the value specified is less than 1 MB, 1 MB is used. If the amount of storage that can be obtained is less than the value specified, the maximum amount available is obtained. The maximum permitted value is 2,097,152,000 bytes (2 GB). The default value is 1,048,575 bytes (1 GB).

5. Optionally, specify whether data compression is turned on or off for data transport between all tasks (internal and external) and between an initiator and the ODBC drivers. Type one of the following values in the **Compression** field:

- 1 Data compression is turned on.
- 2 Data compression is turned off.

The default value is 2 (off).

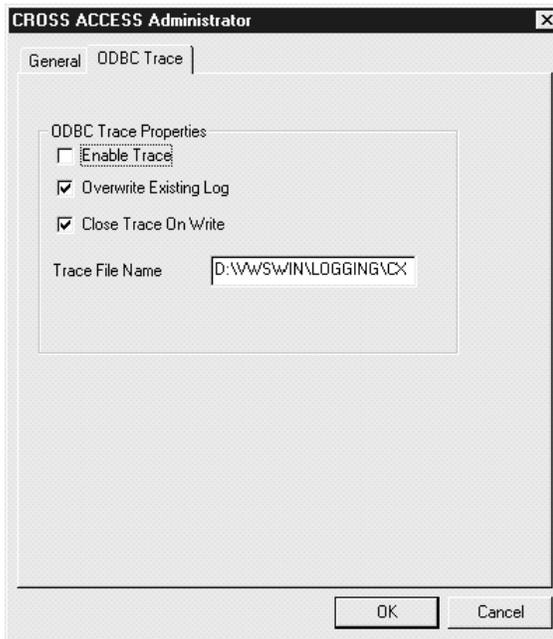
6. Optionally, type a value that corresponds to the amount of information that the ODBC driver writes to the trace log in the **Trace Level** field. The value must be an integer between 0 and 4, where:

Using Classic Connect with the Data Warehouse Center

- 0 No tracing information is logged.
- 1 Minimum tracing information is logged.
- 4 Maximum tracing information is logged.

This trace is different from the ODBC trace; it is specific to the ODBC driver used by Visual Warehouse.

7. Click the **ODBC Trace** tab in the CROSS ACCESS Administrator window. The following figure shows the ODBC Trace page of the CROSS ACCESS Administrator window.



8. Select the **Enable Trace** box to generate an ODBC trace.
9. Select the **Overwrite Existing Log** box to overwrite an existing log trace.
10. Select the **Close Trace on Write** box if you want the driver to close the trace log after each message is written.
11. Specify the name of the trace file in the **Trace File Name** field.
If the directory is not indicated, the trace file will be created in the subdirectory of the Program Files directory that corresponds to the tool that issued the query to the ODBC data source.
12. Click **OK**.

Using Classic Connect with the Data Warehouse Center

Appendix C. Defining values for warehouse deprecated programs and transformers

Deprecated programs are located in the Warehouse Deprecated Programs folder in the Process Modeler. You should not use the programs in this folder to create new steps. Instead, use the data warehouse programs and transformers for all step types. This appendix provides information about the following deprecated programs:

- Visual Warehouse 5.2 programs
- DWC 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:

Defining values for warehouse deprecated programs and transformers

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.
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.

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 data 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 EEE (VWPLDPR) for DB2 UDB Extended Enterprise Edition.
- The user ID and password for the warehouse target.
- The target table that is defined for the step.

Defining values for warehouse deprecated programs and transformers

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 tablespace. Any private tablespace that you create will be used by default for all new tables that do not specify a tablespace. If processing fails, DB2 might put the whole tablespace in hold status, making the tablespace inaccessible. To avoid this hold problem, create a second private tablespace for steps that do not use the load programs.

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:

Defining values for warehouse deprecated programs and transformers

```
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 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 data 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 Load flat file into DB2 EEE (VWPLDPR) for DB2 UDB Extended Enterprise Edition.
- The user ID and password for the warehouse target.
- The target table that is defined for the step.

Defining values for warehouse deprecated programs and transformers

These parameters are predefined. You do not specify values for these parameters.

Recommendation: Create the target table in its own private DB2 tablespace. Any private tablespace that you create will be used for all new tables that do not specify a tablespace. If processing fails, DB2 might put the whole tablespace in hold status, making the tablespace inaccessible. To avoid this hold problem, create a second private tablespace 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.

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 tablespace:

```
CREATE TABLESPACE tablespace-name MANAGED BY SYSTEM USING ('d:/directory')
```

Defining values for warehouse deprecated programs and transformers

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:

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"

Defining values for warehouse deprecated programs and transformers

Dateisiso	Dateisiso causes all date data values to be exported in ISO format.
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.

Defining values for warehouse deprecated programs and transformers

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:

Defining values for warehouse deprecated programs and transformers

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 Extended Edition (VWPLDPR) program (AIX)

Use this step to run the Visual Warehouse 5.2 Load flat file into DB2 Extended Enterprise Edition (AIX only) (VWPLDPR) program in the Data Warehouse Center. This step loads data from a delimited flat file into a DB2 Universal Database Extended Enterprise Edition database, replacing existing data. 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 AIX.

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 node.
4. Runs remote load on all nodes.

If the load step fails on any node, the VWPLDPR program does the following:

1. Builds an empty load data file for each node.
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 nodes involved in the database load. The shared file system must be mounted on the same directory on all nodes. 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.

Defining values for warehouse deprecated programs and transformers

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 tablespace. Any private tablespace that you create will be used for all new tables that do not specify a tablespace. If processing fails, DB2 might put the whole tablespace in hold status, making the tablespace inaccessible. To avoid this hold problem, create a second private tablespace 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 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.

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.

Defining values for warehouse deprecated programs and transformers

- 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 DWC 7.2 Clean Data transformer

This section explains how to define values for the DWC 7.2 Clean Data transformer.

DWC 7.2 Clean Data transformer

Use the DWC 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 DWC 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 Modeler window.

Use the DWC 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 DWC 7.2 Clean Data transformer to perform the following tasks:

- 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

Defining values for warehouse deprecated programs and transformers

You can use the DWC 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 DWC 7.2 Clean Data transformer, you must create a rules table for your clean type. A rules table designates the values that the DWC 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.

Rules tables for DWC 7.2 Clean Data transformer

This topic covers rules tables for the DWC 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 Modeler 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.

For example, Column 1 and Column 2 in a rules table have the values shown here:

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 DWC 7.2 Clean Data transformer

Defining values for warehouse deprecated programs and transformers

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 DWC 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:

Clean type	Minimum number of columns in rules table	Purpose of columns
Find and replace	2 - 3	<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>
Discretize	3 - 4	<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>

Defining values for warehouse deprecated programs and transformers

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 DWC 7.2 Clean Data transformer

You can use the DWC 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 Modeler window.

Procedure:

To define a DWC 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.

Appendix D. 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.

The environment variable:	Is added to, 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

Data Warehouse Center registry updates

The following values are added to the Windows NT, Windows 2000, or Windows XP registry in the following key:

HKEY_LOCAL_MACHINE\Software\IBM\DB2\DataWarehouseCenter\ServiceParms

Value name	Value data
Database name	<Control DB name>
Log Directory	<disk:\dir\>
Password	<password>
Qualifier	<table qualifier>
Userid	<DB2 user ID>

Data Warehouse Center environment structure

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
```

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.

Licenses 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	LAN Distance
AISPO	MVS
AIX	MVS/ESA
AIXwindows	MVS/XA
AnyNet	Net.Data
APPN	NetView
AS/400	OS/390
BookManager	OS/400
C Set++	PowerPC
C/370	pSeries
CICS	QBIC
Database 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/400
DB2 Extenders	SQL/DS
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView
DRDA	Tivoli
eServer	VisualAge
Extended Services	VM/ESA
FFST	VSE/ESA
First Failure Support Technology	VTAM
IBM	WebExplorer
IMS	WebSphere
IMS/ESA	WIN-OS/2
iSeries	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

- accessing
 - data sources 32
- adding
 - data sources 301
- agent site
 - configuration 17
 - default 2
 - defining 17
 - description 2
- agents
 - nonrelational data 279
 - warehouse 2
- aggregating columns 192
- analysis of variance 203
- ANOVA transformer 203

B

- backing up
 - Data Warehouse Center 257
- backward regression 210
- Base Aggregate step 179
- basic statistics 204
- Berkeley sockets 291
- business objects
 - SAP
 - defining 169
- BVBESTATUS table
 - and DB2 Relational Connect 108
 - creating 105

C

- Calculate Statistics transformer 204
- Calculate Subtotals transformer 205
- calculated columns
 - in warehouses 152
- calculation scripts 261
- CASE statement 146
- catalog name 307
- Change Aggregate step 179
- changing your configuration 258
- Chi-square transformer 206
- Classic Connect
 - data server 279
 - Data Warehouse Center step 279
 - nonrelational data mapper 279
 - warehouse agents 279
- Clean Data transformer 184
- clean types 321

- click stream, importing data to Data Warehouse Center 169
- Client Access/400 103
- client connection requests 279
- close trace on write 307
- code transformation 146
- communications
 - options 289
- communications compound address field 289
- compression 307
- configuration files
 - Data Warehouse Center 326
- configuration prerequisites 300
- configurations
 - agent site 17
- configuring
 - data sources 300
 - Data Warehouse Center 260
 - IMS 78
 - Informix
 - AIX 79
 - Solaris Operating Environment 79
 - Windows NT, Windows 2000, Windows XP 59
 - local z/OS client 289
 - LU 6.2
 - communications 304
 - Windows NT 294
 - z/OS 294
 - Microsoft Access 69
 - Microsoft Excel 75
 - Microsoft SQL Server
 - AIX 82
 - Linux 82
 - Solaris Operating Environment 82
 - ODBC drivers 307
 - prerequisite products 288
 - TCP/IP
 - communications 302
 - Windows 2000 293
 - Windows NT 293
 - Windows XP 293
 - z/OS 291
 - VSAM 78
- configuring Data Warehouse Center
 - changing 258

- connecting
 - to source 33
 - DB2 family 33
 - DB2 for VM 35
 - DB2 for VSE 35
 - DB2 Universal Database for iSeries 35
 - DB2 Universal Database for z/OS 35
 - to warehouse
 - DB2 Common Server 100
 - DB2 Enterprise Server Edition 100
 - DB2 for z/OS 104
- connection handler 279
- connectivity
 - establishing
 - iSeries agent 37
 - zSeries agent 39
 - requirements
 - between remote databases 36
 - between the warehouse server and the warehouse agent 10
- connectors 2
- control database
 - initializing 258
 - installing a new one 258
- controlling configuration
 - parameters 279
- correlation coefficient 207
- Correlation transformer 207
- covariance 207
- creating
 - target tables with DB2 Relational Connect 110
- CROSS ACCESS ODBC driver 279
- cross memory 289

D

- data
 - filtering 150
 - inserting 143
 - selecting 143
- data cleansing, name and address 194
- Data export with ODBC to file warehouse program 156
- data mapper 279

- data modeling
 - IBM ERwin Metadata Extract Program 219
- data source handler 279
- Data Transaction Services support
 - Data Warehouse Center 234
- Data Warehouse Center
 - backing up 257
 - configuration files 326
 - environment variables 325
 - logger, starting 8
 - metadata mappings with the DB2 OLAP Integration Server 273
 - OLE DB Support 234
 - security, defining 20
 - server, starting 8
 - starting a step from outside 136
 - status table
 - and DB2 Relational Connect 108
 - creating 105
 - token 227
- Data Warehouse Center programs
 - user-defined 227
- databases
 - cataloging
 - iSeries warehouse agent 38
- DB2
 - accessing as source 33
- DB2 Common Server
 - connecting to warehouse 100
 - privileges 99
- DB2 Connect
 - gateway site 35
- DB2 Enterprise Server Edition
 - connecting to warehouse 100
 - See DB2 Enterprise Server Edition 100
 - privileges 107
 - warehouse target 107
- DB2 for iSeries
 - Data Warehouse Center load utilities
 - modstring parameters 161
 - privileges 100
- DB2 for iSeries Data Load Insert
 - warehouse program 158
- DB2 for iSeries Data Load Replace
 - warehouse program 160
- DB2 for VM
 - accessing as source 33
 - connecting to source 35
 - privileges 34
 - setting up the DB2 Connect gateway site 35
- DB2 for VSE
 - accessing as source 33
 - connecting to source 35
 - Grant to public property 105
 - privileges 34
 - setting up the DB2 Connect gateway site 35
- DB2 for z/OS
 - connecting to warehouse 104
 - Grant to public property 105
 - privileges 104
 - warehouse target 105
- DB2 for z/OS Load warehouse
 - program 165
- DB2 OLAP Integration Server
 - exporting metadata to Data Warehouse Center 244
 - metadata mappings with the Data Warehouse center 273
 - multidimensional database
 - loading from the Data Warehouse Center on a schedule 247
 - loading with data 242
- DB2 Relational Connect
 - accessing source and target data 85, 108
 - accessing with the zSeries warehouse agent 39
 - creating target tables 110
 - defining
 - source 85, 108
 - warehouse 108
 - moving a target table 111
 - server mapping example 86
 - updating an existing table in a remote database 112
- DB2 Universal Database export
 - warehouse program 155
- DB2 Universal Database for iSeries
 - accessing as source 33
 - connecting to source 35
 - privileges 34
 - setting up the DB2 Connect gateway site 35
- DB2 Universal Database for z/OS
 - accessing as source 33
 - connecting to source 35
 - privileges 34
 - setting up the DB2 Connect gateway site 35
- DB2 Universal Database load
 - warehouse program 157
- default agent site 2
- defining
 - agent site 17
 - privileges
 - DB2 Common Server 99
 - DB2 Enterprise Server Edition 107
 - DB2 family 32
 - DB2 for iSeries 100
 - DB2 for VM 34
 - DB2 for VSE 34
 - DB2 for z/OS 104
 - DB2 Universal Database for iSeries 34
 - DB2 Universal Database for z/OS 34
 - program group 226
 - security 20
 - source
 - for use with DB2 Relational Connect 85, 108
- SQL
 - SQL 143
- step 118
 - Base Aggregate 179
 - Change Aggregate 179
 - Clean Data 184
 - Data export with ODBC to file 156
 - DB2 for iSeries Data Load Insert 158
 - DB2 for z/OS Load 165
 - DB2 Universal Database export 155
 - DB2 Universal Database load 157
 - DWC 7.2 Clean Data 321
 - Generate Key Table 190
 - Generate Period Table 191
 - Invert Data 191
 - Pivot data 192
 - Point-in-Time 179
 - Staging Table 179
 - User Copy 179
- warehouse
 - for use with DB2 Relational Connect 108
 - warehouse target
 - DB2 for EEE 107
 - DB2 for z/OS 105
 - in Data Warehouse Center 115
- defining source
 - replication 178
- defining step
 - User Copy 179

- developing steps 134
- development mode 134
- dimension tables 149
- Distributed Relational Database
 - Architecture (DRDA) 33
- DJXSAMP 279
- DSNUTILS 118
- DTS
 - Data Warehouse Center 234
- DWC 7.2 Clean Data
 - transformer 321
- E**
- enable trace 307
- enterprise server
 - definition 279
 - implementation 279
- environment variables 325
 - updating 325
- error handling
 - IBM ERwin metadata extract
 - program 223
- errors
 - logging 269
- establishing connectivity 37
- examples
 - ANOVA transformer 203
 - DB2 Relational Connect server
 - mapping 86
 - external trigger program 137
 - left outer join 147
 - mapping source data to target
 - table 132
 - odbc.ini file entry
 - AIX, Solaris Operating
 - Environment, Linux 95
 - SQL DATE function 152
 - star join 149
 - star schema 149
 - SYSTEM DD statement 199
 - tnsnames.ora file entry
 - AIX, Linux, Solaris Operating
 - Environment 81
 - WHERE clause 150
- exponentially smoothed moving
 - average 209
- exporting
 - DB2 OLAP Integration Server to
 - Data Warehouse Center 246
- exporting data
 - using Data Warehouse Center
 - utilities 155
- external trigger program
 - example 137
 - return code 138
- external trigger program (*continued*)
 - running steps 136
 - status 138
- F**
- fact tables 149
- feedback file 230
- fetch buffer size 307
- file type modifiers
 - Visual Warehouse 5.2
 - programs 316
- files, configuration 326
- filtering data 150
- Fisher-F distributions 203
- forecasting 209
- foreign keys 149
- FormatDate transformer 193
- full outer join 147
- full-model regression 210
- G**
- Generate Key Table
 - transformer 190
- Generate Period Table
 - transformer 191
- goodness-of-fit test 206
- H**
- hostname 291
- I**
- import metadata
 - initializing control database 212
- importing
 - data
 - with SAP R/3 167
 - object definitions 212
 - tag language file into DWC 221
 - WebSphere Site Analyzer
 - data 169
- IMS BMP/DBB
 - initialization service 279
 - interface 279
- IMS DRA
 - initialization service 279
 - interface 279
- IMS logical table 279
- incremental commit 144
- Informix 9.2
 - configuring the client
 - Windows NT, Windows 2000,
 - Windows XP 59
 - configuring the server (with
 - client) 59
- initialization
 - definition 258
- initialization (*continued*)
 - parameters 258
 - types 258
- initialization services 279
- initializing
 - a second control database 258
- inserting data 143
- installing
 - prerequisite products 288
- Invert Data transformer 191
- IP address
 - configuring 291
- IWH2LOG.LOG 271
- IWH2LOGC.LOG 271
- IWH2SERV.LOG 271
- J**
- joins
 - full outer join 147
 - left outer join 147
 - outer join 147
 - right outer join 147
 - star join 149
- K**
- kernel host name 260
- key columns 190
- key values 190
- keys
 - foreign 149
 - primary 149
- L**
- left outer join 147
- load balancing 279
- load rules 261
- load utility
 - Data Warehouse Center,
 - modstring parameters 161
- loading
 - data
 - using Data Warehouse Center
 - load utilities 155
- logger service 279
- logical databases 279
- logical tables 279
- logs
 - MQSeries 218
- LU 6.2 configuration worksheet 294
- LU worksheets 294
- M**
- mapping
 - nonrelational data 279
 - steps to data sources 132
- message pool size 307

- message type 269
- messaging, MQSeries 215
- metadata
 - exporting 211
 - exporting to a tag language file 212
- metadata grammar 279
- metadata mappings
 - DB2 OLAP Integration Server and the Data Warehouse center 273
- Microsoft SQL Server
 - configuring for Windows NT, Windows 2000, Windows XP 68
 - ODBC driver, configuring on Windows NT, Windows 2000, Windows XP 68
- moving a target table from DB2 Relational Connect to remote database 111
- Moving Average 209
- moving data
 - replicating 175
- MQSeries
 - creating views 215
 - error log file 218
 - importing 216
 - installation requirements 216
 - user-defined program 217
 - using with the Data Warehouse Center 214
 - XML metadata and 216
- MTO interface 279
- multidimensional cube
 - loading with data 242

N

- name and address cleansing, Trillium 194
- Network File System (NFS)
 - using in the Data Warehouse Center 96
- non-IBM sources
 - supported versions and release levels 43
- nonparametric tests 206
- nonrelational data 279
- nonrelational data mapper 279
- notebook
 - Program
 - Agent sites page 226
 - Parameters page 227

O

- Object REXX for Windows 227
- ODBC (open database connectivity) driver 279
- OLAP server
 - Calc with calc rules (ESSCALC2) warehouse program 262
 - Default calc (ESSCALC1) warehouse program 262
- OLE DB Support 234
- outer join 147
- outline 261
- overwrite existing log 307

P

- p-value 203
- P-value 207
- parameter substitution 227
- password files
 - Data Warehouse Center replication 180
- passwords
 - control database 260
- Pearson product-moment correlation coefficient 207
- period table 191
- Pivot Data transformer 192
- point-in-time replication steps 179
- port numbers
 - TCP/IP
 - Classic Connect for z/OS 291
- primary keys
 - definition 116
 - dimension tables 149
- printing replication steps 118
- privileges
 - defining 32
 - sources 32
 - warehouse
 - DB2 Common Server 99
 - DB2 Enterprise Server Edition 107
 - DB2 for iSeries 100
 - DB2 for VM 34
 - DB2 for VSE 34
 - DB2 for z/OS 104
 - DB2 Universal Database for iSeries 34
 - DB2 Universal Database for z/OS 34
 - granting 99
- program groups 226
- Program notebook
 - Agent sites page 226

- Program notebook (*continued*)
 - Parameters page 227
 - program step, replication 2
 - pseudo-relational data 279

Q

- queries
 - processor 279

R

- recovery
 - backing up Data Warehouse Center 257
- region controller 279
- registry
 - updating variables 325
- regression
 - backward 210
 - full-model 210
 - transformer 210
- relational database
 - queries 279
- remote
 - file servers, accessing from the Data Warehouse Center 91
 - replicating tables 175
 - replication
 - Data Warehouse Center password files 180
 - replication step, using in a process 179
 - response time out 307
 - right outer join 147
 - rolling sum 209
 - running subtotal 205

S

- SAP
 - business objects
 - defining 169
 - R/3 system
 - loading data 167
 - source
 - defining 168
 - steps 118
 - security
 - Data Warehouse Center 20
 - selecting
 - data 143
 - supported DB2 data sources 25
 - server schedule mode 260
 - settings
 - database catalog option 303, 306
 - log directory 271
 - simple moving average 209
 - SNA protocol 289

- source columns, mapping to target columns 132
 - sources
 - DB2 Connect gateway site 35
 - DB2 family products 32
 - DB2 for VM 34
 - DB2 for VSE 34
 - DB2 types 25
 - DB2 UDB for iSeries 34
 - DB2 UDB for z/OS 34
 - defining 40
 - IMS 78
 - Informix, AIX 79
 - Informix, Solaris Operating Environment 79
 - Microsoft Access 69
 - Microsoft Excel 75
 - Microsoft SQL Server, AIX 82
 - Microsoft SQL Server, Linux 82
 - Microsoft SQL Server, Solaris Operating Environment 82
 - non-DB2 83
 - replication 175, 178
 - VSAM 78
 - warehouse 17, 98
 - WebSphere Site Analyzer 170
 - SQL (Structured Query Language)
 - CASE statement 146
 - JOIN clause 146
 - OUTER JOIN keyword 147
 - SUM function 151
 - WHERE clause 150
 - SQL step, description 2
 - Staging Table replication step 179
 - standard configuration files 291
 - star joins 149
 - star schema
 - description 149
 - starting
 - Data Warehouse Center
 - logger 8
 - server 8
 - warehouse agent daemon
 - iSeries 11
 - Windows NT, Windows 2000, or Windows XP 10
 - statistical transformers
 - ANOVA transformer 203
 - Calculate Statistics transformer 204
 - Calculate Subtotals transformer 205
 - Chi-Square transformer 206
 - Correlation transformer 207
 - Moving Average 209
 - statistical transformers (*continued*)
 - Regression transformer 210
 - statistics, calculating 204
 - steps
 - adding 118
 - ANOVA transformer 203
 - Base Aggregate 179
 - Calculate Statistics transformer 204
 - Calculate Subtotals transformer 205
 - Change Aggregate 179
 - Chi-square transformer 206
 - Clean Data 184
 - code transformation 146
 - Correlation transformer 207
 - Data export with ODBC to file 156
 - DB2 for iSeries Data Load Insert 158
 - DB2 for iSeries Data Load Replace 160
 - DB2 for z/OS Load 165
 - DB2 UDB export 155
 - DB2 UDB load 157
 - developing 134
 - DWC 7.2 Clean Data 321
 - Generate Key Table 190
 - Generate Period Table 191
 - in user-defined programs 226
 - Invert Data 191
 - mapping source columns to target columns 132
 - mode
 - description 134
 - development 134
 - test 134
 - Moving Average 209
 - OLAP Server: Calc with calc rules (ESSCALC2) warehouse program 262
 - OLAP Server: Default calc (ESSCALC1) warehouse program 262
 - Pivot data 192
 - Point-in-Time 179
 - printing 118
 - Regression transformer 210
 - running 134
 - SAP 118
 - SQL, selecting and inserting data 143
 - Staging Table 179
 - starting from outside the Data Warehouse Center 136
 - steps (*continued*)
 - status and user-defined program feedback 230
 - testing 134
 - used to filter source data 150
 - User Copy 179
 - Web traffic 118
 - subject areas
 - defined 2
 - warehouse 2
 - subsystem
 - interfaces 279
 - subtotal, calculating 205
 - system processes 268
- ## T
- T-value 207
 - tag language files
 - ERwin 220
 - XML 212
 - target tables
 - creating with DB2 Relational Connect 110
 - moving 111
 - updating existing table in a remote database 112
 - targets
 - agent site 17
 - columns, mapping to source columns 132
 - warehouse 114
 - TCP/IP
 - configuration
 - example 294
 - information 326
 - worksheet 294
 - protocol 289
 - test environments
 - Data Warehouse
 - steps 134
 - test mode 134
 - time-series analysis 209
 - tokens
 - for parameters 227
 - traces
 - Data Warehouse Center
 - components 266
 - directory 260
 - file name 307
 - levels 260, 307
 - transformations
 - code 146
 - transformers
 - FormatDate transformer 193
 - logging 269

- transformers (*continued*)
 - setting up 118
 - steps 2
- Trillium Software System
 - cleansing data with 194
 - error handling 200
- U**
- UDP (user datagram protocol),
 - MQSeries 217
- updates
 - environment variables 325
 - tables in a remote database 112
- user copy table
 - step, replication 179
- user process 268
- user-defined programs
 - agent sites for 226
 - and step status 230
 - changing agent to user
 - process 268
 - description 225
 - feedback 230
 - MQSeries 217
 - Object REXX for Windows 227
 - parameters 228
 - passing pre-defined parameters
 - to 229
 - return code 230
 - steps 2
 - writing 225, 227
- V**
- variables
 - environment
 - Data Warehouse Center 325
- verifying
 - communication 14
- VSAM
 - interface 279
 - logical table 279
- W**
- warehouse agent daemon
 - iSeries
 - starting 11
 - verifying it has started 11
 - stopping 16
 - stopping the zSeries warehouse
 - agent daemon 16
 - Windows NT, Windows 2000, or
 - Windows XP
 - starting 10
 - z/OS
 - starting in the foreground 12
- warehouse agent daemon (*continued*)
 - zSeries
 - starting in the
 - background 13
 - starting in the foreground 12
- warehouse agents
 - description 2
 - local 2
 - remote 2
 - sites 2
- warehouse privileges
 - DB2 Enterprise Server
 - Edition 107
 - DB2 for iSeries 100
 - DB2 for z/OS 104
- warehouse process
 - description 2
 - promoting 139
 - task flow 140
- warehouse programs
 - Data export with ODBC to
 - file 156
 - DB2 for iSeries Data Load
 - Insert 158
 - DB2 for iSeries Data Load
 - Replace 160
 - DB2 for z/OS Load 165
 - DB2 Universal Database
 - export 155
 - DB2 Universal Database
 - load 157
 - logging 269
 - OLAP Server: Calc with calc
 - rules (ESSCALC2) 262
 - OLAP Server: Default calc
 - (ESSCALC1) 262
 - parameters 227
 - steps 2
 - use in steps 226
- warehouse schemas
 - adding tables and views 239
 - defining 239
 - exporting to the OLAP
 - Integration Server 240
 - in DB2 OLAP Integration
 - Server 241
 - publishing metadata about 247
- warehouse sources
 - defined 2
 - defining 40
 - description 2
 - Linux 25
 - SAP
 - defining 168
- warehouse sources (*continued*)
 - supported versions and release
 - levels 25
 - viewing data in 39
 - WebSphere Site Analyzer
 - defining 170
- warehouse steps
 - description 2
 - program 2
 - SQL 2
 - transformer 2
 - user-defined program 2
- warehouse targets
 - defining 115
 - description 2, 113
- warehouse transformers
 - Clean Data 184
 - DWC 7.2 Clean Data 321
 - Generate Key Table 190
 - Generate Period Table 191
 - Invert Data 191
 - Pivot Data 192
- warehousing
 - DB2 for iSeries, DB2 Connect
 - gateway site 101
 - mapping to source data 132
 - objects 2
 - overview 1
 - tasks 2
- Web traffic polling, replication
 - step 118
- WebSphere Site Analyzer
 - loading data into the Data
 - Warehouse Center 169
 - warehouse sources 170
- WHERE clause
 - description 150
- WLM initialization service 279
- writing user-defined programs 225
- X**
- XTClient syntax 137
- Z**
- z/OS
 - client application 279

Contacting IBM

In the United States, call one of the following numbers to contact IBM:

- 1-800-237-5511 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 www.ibm.com/planetwide

Product information

Information regarding DB2 Universal Database products is available by telephone or by the World Wide Web at www.ibm.com/software/data/db2/udb

This site contains the latest information on the technical library, ordering books, client 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



Part Number: CT16DNA

Printed in U.S.A.

SC27-1123-00



(1P) P/N: CT16DNA



Spine information:



IBM® DB2 Universal Database™ Data Warehouse Center Admin Guide

Version 8