

IBM® DB2® Universal Database



Data Warehouse Center Administration Guide

Version 7

IBM® DB2® Universal Database



Data Warehouse Center Administration Guide

Version 7

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 445.

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.

Order publications through your IBM representative or the IBM branch office serving your locality or by calling 1-800-879-2755 in the United States or 1-800-IBM-4YOU in Canada.

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, 2001. All rights reserved.**

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

Contents

About this book	ix
Who should read this book	ix
Prerequisite publications	ix
Chapter 1. About data warehousing.	1
What is data warehousing?	1
Data warehousing in DB2 Universal Database	2
Subject areas	2
Warehouse sources	2
Warehouse targets	2
Warehouse agents and agent sites	2
Processes and steps	3
Warehousing tasks	5
Chapter 2. Setting up your warehouse.	7
Starting the Data Warehouse Center	7
Starting the warehouse server and the logger	7
Starting a warehouse agent daemon	8
Stopping the warehouse agent daemon	11
Starting the Data Warehouse Center administrative interface	12
Defining agent sites	13
Agent site configurations.	14
Connectivity requirements for the warehouse server and the warehouse agent	16
Configuring TCP/IP on OS/390	16
Updating your environment variables on OS/390	17
Defining an agent site to the Data Warehouse Center	17
Data Warehouse Center security	19
Defining a warehouse user	22
Defining a warehouse group	24
Chapter 3. Setting up warehouse sources 27	
Selecting Data Warehouse Center data sources	27
Windows NT or Windows 2000	29
AIX	36
Solaris Operating Environment.	41
OS/2	45
Setting up access to DB2 Universal Database source databases	49
Defining privileges to DB2 Universal Database source databases	49
Establishing connectivity to DB2 Universal Database source databases	49
Setting up access to DB2 DRDA source databases	50
Prerequisite products	50
Defining privileges to DB2 DRDA source databases	50
Setting up the DB2 Connect gateway site	51
Establishing connectivity to DB2 DRDA source databases	51
Connectivity requirements for remote databases from the AS/400 agent	52
Establishing connectivity to local and remote databases from the AS/400 agent	52
Connectivity requirements for remote databases from the OS/390 agent	54
Using DataJoiner	54
Defining a DB2 warehouse source.	54
Accessing remote files.	58
Accessing files through a Windows NT or Windows 2000 file server.	58
Accessing files with NFS.	59
Accessing files with FTP	59
Accessing data files with Copy file using FTP.	60
Defining a file source	60
Accessing non-DB2 database warehouse sources.	63
Setting up non-DB2 database warehouse sources on Windows NT	63
Setting up non-DB2 database warehouse sources on AIX or the Solaris Operating Environment.	87
Setting up non-DB2 database warehouse sources on OS/2	93
Defining a non-DB2 database warehouse source in the Data Warehouse Center	97
Opening the Warehouse Source notebook	98
Adding information about your warehouse source	98
Specifying an agent site	98
Specifying database information	98
Importing source tables and views	99
Authorizing access for warehouse groups	100
Specify the retry count	101

Defining warehouse sources for use with DataJoiner	101	Linking a step to sources and targets . . .	145
Chapter 4. Setting up access to a warehouse.	105	Defining the basic values of a step subtype	149
Setting up a DB2 Universal Database warehouse	105	Running warehouse steps	153
Defining privileges to DB2 Universal Database warehouses	105	Developing warehouse steps	153
Establishing connectivity to DB2 Universal Database warehouses	105	Testing warehouse steps	153
Setting up a DB2 for AS/400 warehouse	106	Scheduling warehouse processes	157
Using DB2 Connect	106	Starting a step from outside the Data Warehouse Center	162
Using CA/400	107	Chapter 6. Moving data	167
Setting up a DB2 for OS/390 warehouse	109	Selecting and inserting data	167
Defining privileges to DB2 for OS/390 warehouses	109	Defining the step	167
Establishing connectivity to the DB2 for OS/390 warehouse database	109	Incremental commit	170
Creating the Data Warehouse Center status table	110	Loading and exporting data	171
Defining the warehouse to the Data Warehouse Center.	110	Exporting data	171
Setting up a DB2 EEE warehouse	110	Loading data	174
Defining privileges to the DB2 EEE database	110	Manipulating files	189
Establishing connectivity to the DB2 EEE database	111	Defining values for the Copy File using FTP (VWPRCPY) program	189
Defining the DB2 EEE database to the Data Warehouse Center	111	Defining values for a Run FTP Command File program (VWPFTP)	190
Defining warehouses for use with DataJoiner	111	Defining values for a Submit OS/390 JCL jobstream (VWPMVS) program	192
Creating target tables with DataJoiner	113	Setting up replication in the Data Warehouse Center	193
Moving the target table	114	Replicating tables	194
Creating or updating a table in a remote database	114	Defining a replication source in the Data Warehouse Center	196
Defining a warehouse target	115	Defining a user copy, point-in-time, or base aggregate replication step	197
Defining warehouse target properties	116	Defining a change aggregate replication step	200
Defining a primary key	121	Defining a staging table replication step	203
Defining foreign keys	122	Extracting data from an SAP R/3 system	206
Creating the Data Warehouse Center status table in the warehouse database	123	Defining an SAP source	206
Chapter 5. Defining and running processes.	125	Defining the properties of each source business object	207
Defining a subject area	126	Defining an SAP step	207
Defining a process	126	Extracting data from a WebSphere Site Analyzer database	208
Opening the process	127	Defining a WebSphere Site Analyzer source	209
Adding sources and targets to a process	127	Defining a Web traffic polling step	210
Adding steps to the process	128	Chapter 7. Transforming data	211
Step subtypes	131	Joining source columns	211
		Transforming codes	214
		Adding nulls to joins	214
		Generating star joins	216

Filtering data	217	ERwin to Information Catalog Manager mapping.	278
Adding calculated columns	218		
Transforming target tables	220		
Cleaning data	220		
Generating key columns	227		
Generating period data	229		
Inverting data	233		
Pivoting data	235		
Changing the format of a date field.	238		
Chapter 8. Calculating statistics	241		
ANOVA transformer.	241		
Calculate Statistics transformer	243		
Calculate Subtotals transformer	245		
Chi-square transformer	248		
Correlation transformer.	251		
Moving Average transformer	254		
Regression transformer	259		
Chapter 9. Cleansing name and address data	261		
Requirements	261		
Trillium Software System components	262		
Using the Trillium Batch System with the Data Warehouse Center.	262		
Importing Trillium metadata	263		
Mapping the metadata	265		
Restrictions	266		
Writing a Trillium Batch System JCL file	266		
Defining a Trillium Batch System step	267		
Using the Trillium Batch System user-defined program	267		
Error handling.	268		
Error return codes	269		
Log file	270		
Chapter 10. Data modeling	271		
What is the IBM ERwin Metadata Extract Program?	271		
Software requirements	271		
Program files	272		
Creating tag language files.	272		
Importing a tag language file into the Data Warehouse Center	274		
Importing a tag language file into the Information Catalog Manager.	275		
Troubleshooting	276		
ERwin to DB2 Data Warehouse Center mapping.	277		
		Chapter 11. Refreshing a OLAP Server database	281
		Defining values for the OLAP Server: Default calc (ESSCALC1) warehouse program	282
		Defining values for the OLAP Server: Calc with calc rules (ESSCALC2) warehouse program	283
		Defining values for the OLAP Server: Free text data load (ESSDATA1) warehouse program	284
		Defining values for the OLAP Server: Load data from file with load rules (ESSDATA2) warehouse program	285
		Defining values for an OLAP Server: Load data from SQL table with load rules (ESSDATA3) warehouse program	287
		Defining values for an OLAP Server: Load data from a file without using load rules (ESSDATA4) warehouse program	288
		Defining values for an OLAP Server: Update outline from file (ESSOTL1) warehouse program	289
		Defining values for an OLAP Server: Update outline from SQL table (ESSOTL2) program	290
		Chapter 12. Maintaining the warehouse database	293
		Reorganizing data	293
		Defining values for a DB2 Universal Database REORG program.	293
		Defining values for a DB2 for OS/390 REORG program	294
		Defining values for a DB2 for OS/390 Utility program	296
		Updating system catalog statistics	301
		Defining values for a DB2 UDB RUNSTATS program.	301
		Defining values for a DB2 OS/390 RUNSTATS program.	302
		Using MQSeries with the Data Warehouse Center	304
		Creating views for MQSeries messages	304
		Importing MQSeries messages and XML metadata.	305

Chapter 13. Extending the Data Warehouse Center with user-defined programs	309
Defining a user-defined program.	309
Defining a program group	309
Specifying the program to use	310
Specifying the agent site to use	311
Specifying parameters	312
Using a user-defined program step	313
Testing steps that use user-defined programs	315
Writing your own program for use with the Data Warehouse Center	315
Passing parameters	316
Returning status information	317
Chapter 14. Maintaining the Data Warehouse Center	323
Backing up the Data Warehouse Center	323
Expanding your warehouse	324
Exporting and importing Data Warehouse Center metadata	324
Exporting metadata to another Data Warehouse Center	325
Importing metadata	327
Publishing Data Warehouse Center metadata	331
Planning to publish metadata	331
Selecting metadata to publish	332
Updating published metadata.	335
Scheduling regular updates of published metadata.	336
Viewing log files for the publication updates	336
Propagating deleted objects to the information catalog	336
Changing the active warehouse control database	337
Initializing a warehouse control database during installation	338
Configuring the Data Warehouse Center	339
Changing the Data Warehouse Center properties	339
Chapter 15. Creating a star schema from within the Data Warehouse Center	341
Designing the warehouse schema in the Data Warehouse Center	343
Defining the warehouse schema	343
Adding tables and views to the warehouse schema	344
Autojoining tables	345
Adding join relationships between non-keyed columns	345
Exporting a warehouse schema to the DB2 OLAP Integration Server	346
Working with a warehouse schema in the DB2 OLAP Integration Server.	348
Creating an outline and loading the data for the multidimensional cube in the DB2 OLAP Integration Server	350
Exporting the metadata to the Data Warehouse Center	352
Loading the multidimensional cube from the Data Warehouse Center	353
Creating a schedule to populate the multidimensional cube	354
Publishing metadata about your warehouse schema	355
Appendix A. Data Warehouse Center logging and trace data	357
The basic logging function.	357
Component trace data	358
Warehouse programs and transformers	359
Start error trace files	360
Appendix B. Metadata mappings	363
Metadata Mappings between the Information Catalog Manager and the Data Warehouse Center	363
Metadata mappings between the Information Catalog Manager and OLAP server	373
Metadata mappings between the DB2 OLAP Integration Server and the Data Warehouse Center	375
Appendix C. How the Data Warehouse Center migrates Version 5.2 definitions	377
Mapping between Visual Warehouse and Data Warehouse Center objects	377
Subjects	377
Warehouses.	377
Business views.	378
Visual Warehouse programs	378
Security changes	379
Appendix D. Defining values for Visual Warehouse 5.2 compatible programs	381

Defining values for a Visual Warehouse 5.2 DB2 UDB Data Export (VWPEXPT1) program	381	Configuring communications protocols between OS/390 and Windows NT	414
Defining values for a Visual Warehouse DB2 UDB Data Load Insert (VWPLOADI) program	382	Communications options	414
Defining values for a Visual Warehouse DB2 UDB Data Load Replace (VWPLOADR) program	385	Configuring the TCP/IP communications protocol	415
Defining values for a Visual Warehouse 5.2 DB2 UDB REORG (VWPREORG) program	388	Configuring the LU 6.2 communications protocol	419
Define values for a Visual Warehouse 5.2 DB2 UDB RUNSTATS (VWPSTATS) program	388	Configuring a Windows NT client	424
Defining values for a Visual Warehouse 5.2 Load flat file into DB2 UDB EEE (VWPLDPR) program (AIX only)	389	Installing the Classic Connect Drivers component	424
		Installing the CROSS ACCESS ODBC driver	424
		Configuring data sources	424
		Migrating from the Visual Warehouse Host Adapters to Classic Connect	435
Appendix E. The data warehousing sample	393	Appendix G. Data Warehouse Center environment structure	437
Installing the sample.	393	Data Warehouse Center environment variables	437
Creating the sample databases	393	Data Warehouse Center registry updates	437
Creating the warehouse database	394	Data Warehouse Center configuration files	438
Using Sample Contents	395		
Viewing the sample data	395	Appendix H. Microsoft OLE DB and Data Transaction Services support	439
Viewing and modifying the sample metadata	396	Software requirements	439
Promoting the steps	400	Restrictions	439
Running the steps.	401	Creating views for OLE DB table functions	440
Viewing the sample warehouse data	401	Creating views for DTS packages	440
Appendix F. Using Classic Connect with the Data Warehouse Center.	403	Appendix I. Creating an i2 step	441
What is Classic Connect?	403	Importing metadata for i2 sources	441
What does it do?	403	Creating the i2 steps	442
Which data sources does it access?	403		
How is it used?	404	Notices	445
What are its components?	404	Trademarks	448
Setting up the environment	412		
Hardware and software requirements	412	Index	451
Installing and configuring prerequisite products	412	Contacting IBM	459
		Product information	459

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.

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* for your operating system 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.

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

- The Windows NT® 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. For more detailed information about warehousing, and for information about using the Data Warehouse Center, see the Data Warehouse Center online help.

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:

- They 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.

Data warehousing in DB2 Universal Database

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 nearly any relational or nonrelational source (table, view, or file) 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[®], AS/400[®], OS/2[®], OS/390[®], and Windows NT 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 on Windows NT that the Data Warehouse Center defines during initialization of the warehouse control database.

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

For more information, see “Defining processing options” on page 151.

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. For more information about steps, see the “Chapter 5. Defining and running processes” on page 125 and the Data Warehouse Center online help.

SQL steps

An 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.

Program steps

There are several types of program steps: DB2 for AS/400 programs, DB2 for OS/390 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.

Warehousing 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.
- If you have the DB2 Warehouse Manager package, 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. Users of the warehouse can search the catalog 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.

You can learn more about these tasks and others by using the *Business Intelligence Tutorial*, viewing the *DB2 Universal Database Quick Tour*, reading the Data Warehouse Center online help, or reading this book.

Chapter 2. Setting up your warehouse

To set up your warehouse, you need to log on to the Data Warehouse Center and specify the agent sites that the Data Warehouse Center will use when it accesses the source and target databases. You also need to set up security for the objects that you will define for your warehouse. Finally, you need to set up subject areas that will contain the information about the processes that are required to populate your warehouse.

This chapter describes the steps you need to take to set up your warehouse, including:

- “Starting the Data Warehouse Center” on page 7.
- “Defining agent sites” on page 13.
- “Data Warehouse Center security” on page 19.

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. The Data Warehouse Center adjusts with you.

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. The warehouse agent can start automatically or manually. You open the Data Warehouse Center administrative interface manually from the DB2 Control Center.

Starting the warehouse server and the logger

The warehouse server and the warehouse logger run as Windows NT Services. To start them, 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, 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

Setting up your warehouse

restart the warehouse server and logger after you stop and restart the DB2 server to restore connections.

To start the warehouse server and logger manually, use one of the following options:

- From the Windows NT desktop:
 1. Click **Start** → **Settings** → **Control Panel** → **Services**.
 2. Scroll down the list until you find **Warehouse Server**. Click **Start**, and click **OK**.
- Enter the following command from a DOS command prompt:
NET START VWKERNEL

Starting a warehouse agent daemon

The warehouse server can function as 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 or Solaris Operating Environment warehouse agent, the warehouse agent daemon starts automatically.

If you are using a remote Windows NT warehouse agent or an OS/2 warehouse agent, you can select whether to start the warehouse agent daemon automatically or manually during installation.

If you are using an AS/400 or OS/390 warehouse agent, you must start it manually.

For more information about installing warehouse agents, see the *DB2 Warehouse Manager Installation Guide*.

Manually starting the Windows NT warehouse agent daemon

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

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

Manually starting the OS/2 warehouse agent daemon

To manually start the OS/2 warehouse agent, type the following command at an OS/2 command prompt:

```
START /B "IBM Visual Warehouse-NT OS/2 Daemon" d:\directory\VWDAEMON.EXE
```

where `d:\directory` is the drive and directory in which you installed the warehouse agent and warehouse agent daemon.

Manually starting the AS/400 warehouse agent daemon

After you install the AS/400 warehouse agent, you need to start the warehouse agent daemon.

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

To start the agent daemon, enter STRVWD at an AS/400 command prompt. The STRVWD command starts QIWH/IWHVWD (the warehouse agent daemon) in the QIWH subsystem. This causes all warehouse agent processes that are started by the warehouse agent daemon to start in the QIWH subsystem.

To verify that the warehouse agent daemon started:

1. Enter WRKACTJOB at an AS/400 command prompt.
2. Look for the jobs VWD and IWH4MSGQ. If these jobs are displayed, the warehouse agent daemon is started.

The warehouse agent daemon runs as a background job.

Occasionally, you might want to verify that the AS/400 warehouse agent daemon that you started is still running. For example, you use the AS/400 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.

To verify that the AS/400 warehouse agent daemon is active:

1. At an AS/400 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 not displayed, the warehouse agent is inactive.

Starting the OS/390 warehouse agent daemon

After you finish configuring your system for the OS/390 warehouse agent, you need to start the warehouse agent daemon. Both the OS/390 agent and OS/390 agent daemon run on the UNIX[®] System Services (USS) platform.

To start the warehouse agent daemon in the foreground:

1. Telnet to USS on OS/390 through the OS/390 host name and USS port.
2. Enter `vwd` on the command line to start the agent daemon in the foreground.

Setting up your warehouse

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

To verify from a UNIX shell that the warehouse agent daemon is running, enter `ps -e | grep vwd` on a UNIX shell command line.

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

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

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

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

To verify from an OS/390 console that the warehouse agent daemon is running, enter `D OMVS,A=ALL` at the OS/390 command prompt.

If the warehouse agent daemon is running, a task with the string `vwd` will be 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.

To verify that one site recognizes the fully qualified host name of the other site, use the **ping** command from 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

Occasionally, you might need to stop the warehouse agent daemon, such as when you need to change the environment variables for the warehouse agent and warehouse agent daemon on Windows NT.

Stopping the Windows NT warehouse agent daemon

To stop the Windows NT warehouse agent daemon:

1. Click **Start** → **Settings** → **Control Panel** → **Services**.
2. Select **Warehouse Agent Daemon**.
3. Click **Stop**.

The warehouse agent daemon stops.

4. Click **OK**.

Stopping the AIX or Solaris Operating Environment warehouse agent daemons

The only circumstance under which you need to stop an AIX or Solaris Operating Environment warehouse agent daemon is if you want to change the environment variables for the warehouse agents and its corresponding warehouse agent daemon.

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

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

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

The process ID is displayed.

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

```
kill processid
```

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

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

Stopping the AS/400 warehouse agent daemon

Occasionally, you might need to stop the AS/400 warehouse agent daemon.

Setting up your warehouse

To stop the warehouse agent daemon, enter ENDVWD at an AS/400 command prompt.

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 OS/390 warehouse agent daemon

If you need to stop the agent daemon:

1. Determine the daemon process ID.

You can determine the daemon process ID by entering `ps -ef | grep vwd` or `D OMVS,A=ALL` on the command line.

2. Stop the daemon.

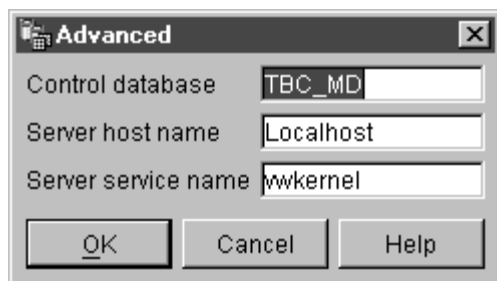
You can stop the daemon process by entering `kill [-9] pid` on the command line, where `pid` is the daemon process ID.

Starting the Data Warehouse Center administrative interface

To start the Data Warehouse Center administrative interface:

1. Start the DB2 Control Center:
 - On Windows NT, click **Start** → **Programs** → **IBM DB2** → **Control Center**.
 - On AIX or the Solaris Operating Environment, type the following command:

```
db2jstrt 6790
db2cc 6790b
```
2. Click **Tools** → **Data Warehouse Center** in the DB2 Control Center window. The Data Warehouse Center Logon window opens.
3. Click **Advanced** if you are logging on for the first time.
The Advanced window opens.
4. In the **Control database** field, type the ODBC system data set name (system DSN) for, the name of the warehouse control database.
5. In the **Server host name** field, type the TCP/IP host name for the workstation where the warehouse server is installed.



6. Click **OK**.

The Advanced window closes.

The next time that you log on, the Data Warehouse Center will use the settings that you specified in the Advanced window.

7. In the **User ID** field of the Logon window, type a valid Data Warehouse Center user ID.

If you are logging on for the first time, type the default warehouse user ID.

8. In the **Password** field, type the password for the user ID.



9. Click **OK**.

The Data Warehouse Center Logon window closes.

Defining agent sites

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

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 an OS/390 warehouse agent, configure TCP/IP, set environment variables, and set up authorization for the warehouse agent daemon.
4. Start the agent daemon. (See "Starting a warehouse agent daemon" on page 8.)
5. Verify communication between the warehouse server and the warehouse agent.
6. Define the agent site to the Data Warehouse Center.

Setting up your warehouse

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 Figure 1.

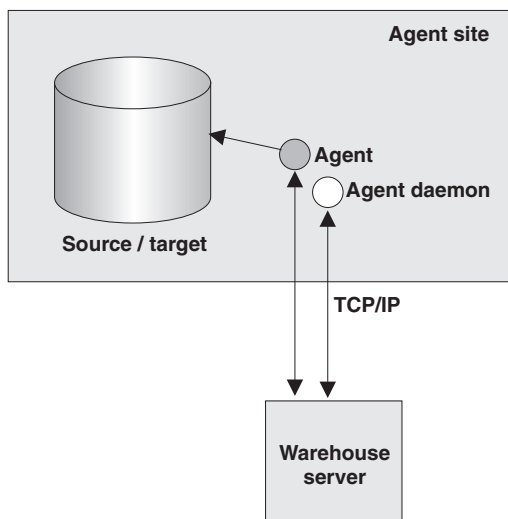


Figure 1. The warehouse agent, warehouse source, and warehouse target are located on the same system.

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

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, as shown in Figure 2 on page 15.

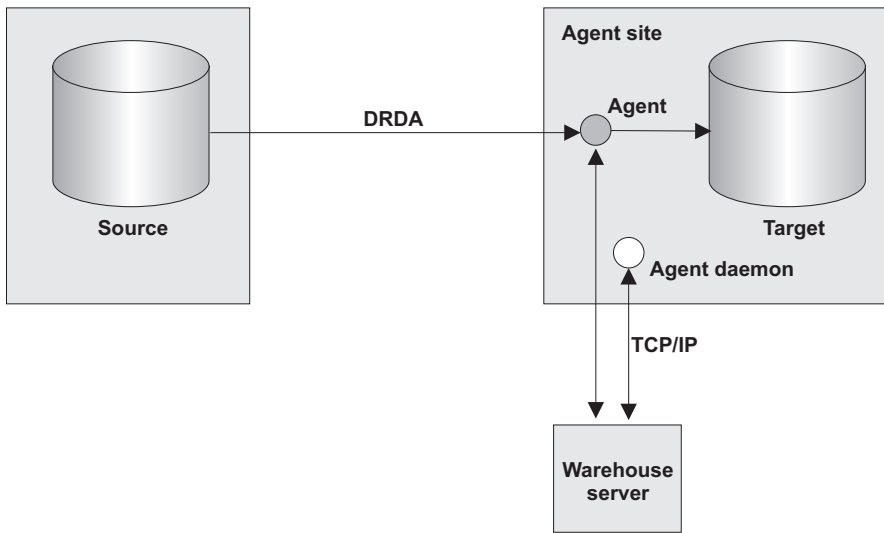


Figure 2. The warehouse agent and warehouse target are located on the same system.

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 Figure 3 on page 16.

Setting up your warehouse

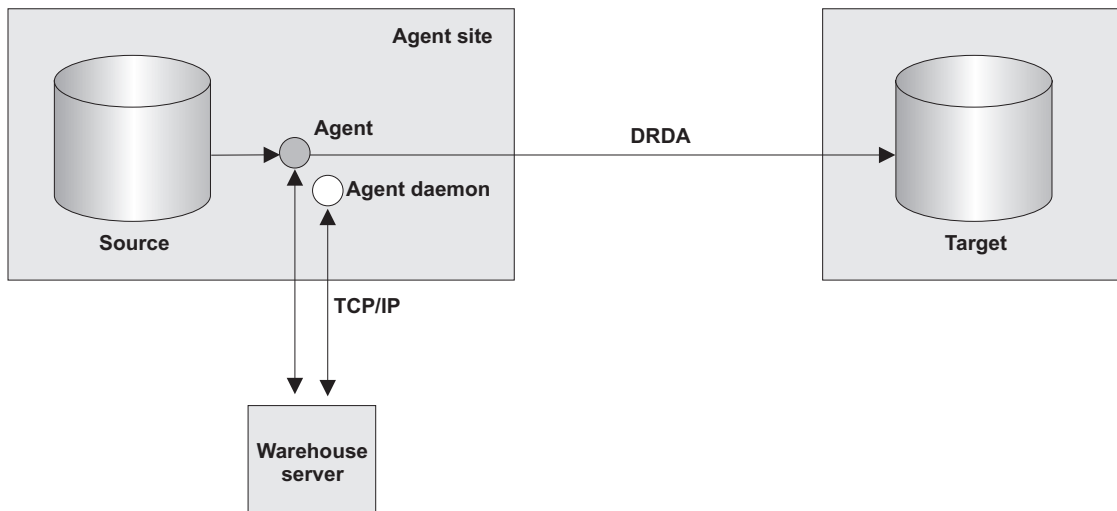


Figure 3. The warehouse agent and warehouse source are located on the same system.

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. For information about defining warehouse security, see “Data Warehouse Center security” on page 19.

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 take place, 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. If you need to change a port number, see *DB2 Universal Database Troubleshooting Guide*.

Configuring TCP/IP on OS/390

To configure TCP/IP on OS/390:

1. Go to the `/etc/services` file or the `'TCPIP.ETC.SERVICES'` file in the OS/390 environment.
2. Add the following services to the file:

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

Updating your environment variables on OS/390

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	Notes
<code>export VWS_LOGGING=/u/mydir/logs/</code>	<code>/u/mydir/logs/</code> is the directory where you want to put the agent logs.
<code>export DSNAOINI='DBA1.INSTALIB(DSNAOINI)'</code>	For more information on INI files, see the <i>DB2 UDB for OS/390 ODBC Guide and Reference</i> .
<code>export STEPLIB='DSN710.SDSNLOAD'</code>	<code>DSN710.SDSNLOAD</code> is the DB2 load library.

Add the following variable to your INI file:

Table 1.

Variable	Notes
<code>export DSNAOTRC='DBA1.DSNAOTRC'</code>	Use only if you want to receive CLI traces.

Defining an agent site to the Data Warehouse Center

Use the Agent Sites notebook to define an agent site to the Data Warehouse Center.

If you are using the local agent that is installed with the warehouse server, you do not have to define the local agent. It is defined automatically as the default agent site.

If you want your agent to access a user object, such as a LAN drive, you should specify the Data Warehouse Center to run as a user process instead of as a system process.

No authority level is required to define an agent site to the Data Warehouse Center.

Setting up your warehouse

To define an agent site to the Data Warehouse Center:

1. Open the Agent Sites notebook:
 - a. From the Data Warehouse Center tree, expand the **Administration** folder.
 - b. Right-click the **Agent Sites** folder and click **Define**. The Agent Sites notebook opens.

The screenshot shows the 'Define Agent Site' dialog box with the following fields and values:

- Name:** San Jose
- Administrator:** Adrienne
- Description:** This agent site has access to Marketing data in San Jose.
- Notes:** (Empty)
- Hostname:** 9.999.99.99
- Operating system:** Windows NT
- Agent module name:** (Empty)
- Operating system logon:**
 - User ID:** adrienne
 - Password:** (Masked with asterisks)
 - Verify password:** (Masked with asterisks)

Buttons: OK, Cancel, Help

2. In the **Name** field, type a name for the agent site that you are defining to the Data Warehouse Center. This name can be up to 80 characters in length.
3. Optional: In the **Administrator** field, type the name of the warehouse administrator who is responsible for this agent site definition.
4. Optional: In the **Description** field, type a business description for your agent definition. This description can be a maximum of 254 characters.
5. Optional: In the **Notes** field, type detailed information that might be helpful to users who can access the Agent Sites notebook.
6. In the **Hostname** field, type the IP address for the system or workstation where the agent is installed, or type the machine host name. Type the IP address in the form of n.nnn.nn.nnn. In the **Operating System** field, select the operating system on which your warehouse agent runs.

7. Optional: In the **Agent module name** field, you can change the name of the warehouse agent program at the direction of IBM Software Support. The default value is IWH2AGNT.
8. In the **User ID** field, type a valid user ID for the agent site, if one is required. If you don't specify a user ID, the agent assumes the characteristics of the warehouse server (for the default agent only) or the agent daemon. For example, if the warehouse agent daemon runs as a system process, the warehouse agent does, too. If you specify a user ID, the warehouse agent inherits the characteristics of the user specified by the user ID.
9. If you supplied a user ID, type the password that is associated with the user ID in the **Password** field. Type the password again in the **Verify Password** field.
10. If you have already defined warehouse sources and targets, click the **Warehouse Sources and Targets** tab. In the **Available warehouse sources and targets** list, select a warehouse source that you want your agent to access, and click >. To select multiple warehouse sources, hold down the Ctrl key, and click the sources that you want to add. Then, click >. If you want to add all of the items in the list, click >>.
11. Optional: If your warehouse agent will run warehouse programs, transformers, or user-defined programs, click the **Programs** tab. In the **Available Programs** list, click a program or transformer that your warehouse agent will run, then click >. To select multiple programs, hold down the Ctrl key, and click the programs that you want to add. Then, click >. If you want to add all of the items in the **Available Programs** list, click >>.
12. Click **OK** to define your agent site to the Data Warehouse Center.

If you are installing a warehouse program on the agent site, you must define the program to the Data Warehouse Center before you can use it. For more information, see "Defining a user-defined program" on page 309.

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 is the users' authorization to 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.

Setting up your warehouse

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*.

Tip: The default warehouse user requires a different type of database and operating system authorization for each operating system that the warehouse control database supports. For more information, see the *DB2 Warehouse Manager Installation Guide*.

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 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.

For a summary of the relationship between Data Warehouse Center objects and the group membership that is required to define or edit the object, see "Data Warehouse Center security" in the online help.

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.

Setting up your warehouse

Figure 4 shows the relationship between warehouse users, warehouse groups, and the user IDs and passwords for the warehouse databases:

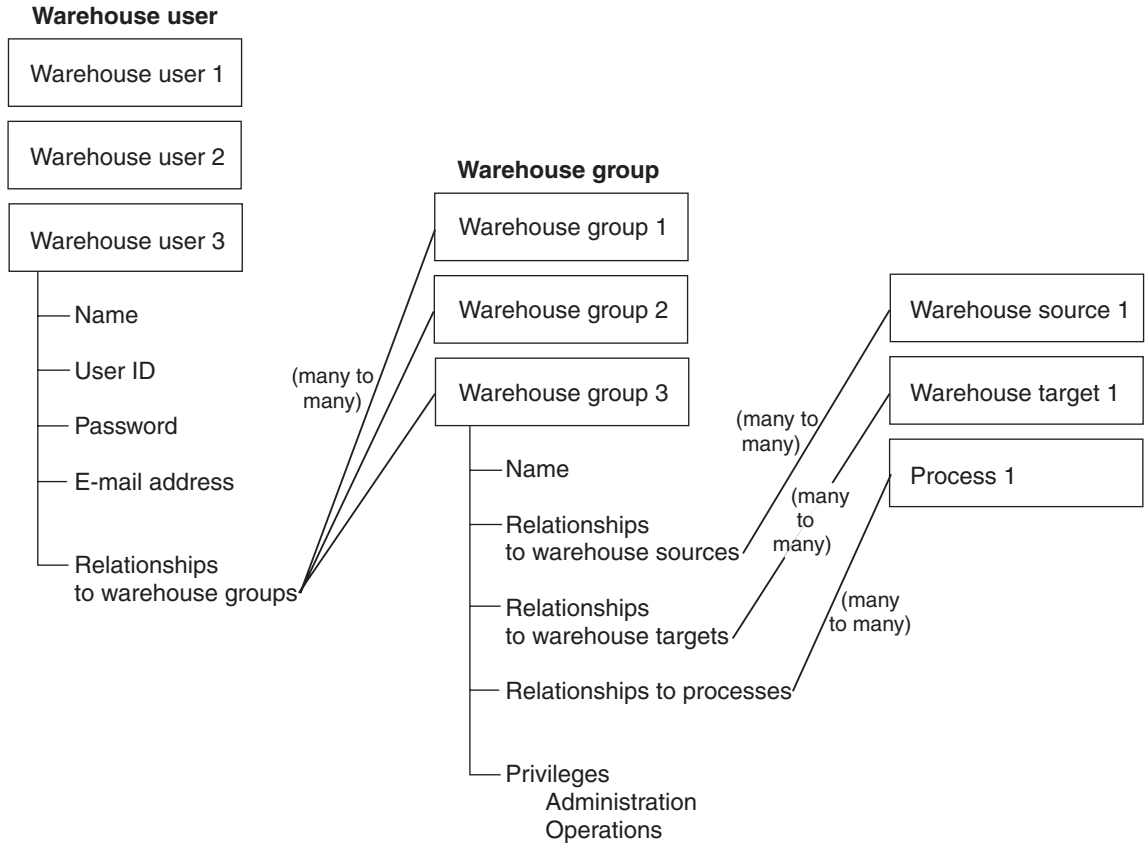


Figure 4. Relationship between warehouse users, warehouse groups, and the user IDs and passwords for the warehouse databases

Defining a warehouse user

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.

To define a warehouse user:

1. In the left side of the main Data Warehouse Center window, click the **Administration** folder.
2. Expand the **Warehouse Users and Groups** tree.
3. Right-click the **Warehouse Users** folder, and click **Define**.

The Define Warehouse User notebook opens.

The screenshot shows the 'Define Warehouse User' dialog box. The 'Warehouse User' tab is selected, and the 'Security' sub-tab is active. The 'Name' field contains 'Sample user', 'Administrator' contains 'Adrienne', and 'Description' contains 'This is a user of the Data Warehouse Center sample.'. The 'Data Warehouse Center logon' section has 'User ID' set to 'db2admin', and both 'Password' and 'Verify password' fields are masked with asterisks. The 'E-mail notification' section has an empty 'E-mail address' field. The 'Active user' checkbox is checked. At the bottom, there are 'OK', 'Cancel', and 'Help' buttons.

4. In the **Name** field, type the business name of the user.
The name identifies the user ID within the Data Warehouse Center. This name can be up to 80 characters, including spaces.
5. In the **Administrator** field, type the contact for this user.
6. In the **Description** field, type a short description of the user.

Tip: You can use the **Description** and **Notes** fields to provide metadata about the definitions for your warehouse. You can then publish this metadata in an information catalog for the warehouse. Users of the warehouse can search the metadata to find the warehouse that contains the information they need to query.

7. In the **User ID** field, type the new user ID.
The user ID must be no longer than 60 characters and cannot contain spaces, dashes, or special characters (such as @, #, \$, %, >, +, =). It can contain the underscore character.

Setting up your warehouse

8. In the **Password** field, type the password. Then, type the password again in the **Verify Password** field.

Passwords must be a minimum of six characters and cannot contain spaces, dashes, or special characters.

Tip: You can change your password on this page of the Define Warehouse User notebook.

9. Verify that the **Active User** check box is selected.

Tip: You can clear this check box to temporarily revoke a user's access to the Data Warehouse Center, without deleting the user definition.

10. Click **OK** to save the warehouse user and close the notebook.

Defining a warehouse group

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.

To define the warehouse group:

1. From the main Data Warehouse Center window, right-click the **Warehouse Groups** folder, and click **Define**.

The Warehouse Groups notebook opens.

The screenshot shows the 'Define Warehouse Group' dialog box with the following content:

- Warehouse Group** | Warehouse Users | Warehouse Sources and Targets | Processes
- Name:** Sample Warehouse Group
- Administrator:** Adrienne
- Description:** This is a warehouse group that contains users of the sample.
- Notes:** (Empty text area)
- Available privileges:** (Empty list box)
- Selected privileges:**

Name	Description
Administration	Authority to det
Operations	Authority to acc
- Buttons: >, >>, <<, <
- Bottom buttons: OK, Cancel, Help

2. In the **Name** field, type the name for the new warehouse group.
3. Optional: In the **Administrator** field, type the contact for this new warehouse group.
4. Optional: In the **Description** field, type a short description of the new warehouse group.
5. Optional: In the **Notes** field, type any additional information that an administrator might need to know about this warehouse group.
6. From the **Available privileges** list, select the privileges that you want to assign to the warehouse group, and click >. The privileges that you select are moved to the **Selected privileges** list.

If you want to assign all of the privileges in the **Available privileges** list, click >>.

You can select from the following privileges:

Administration

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.

Setting up your warehouse

Operations

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

7. On the Warehouse Users page, select the warehouse users that you want to include in the warehouse group from the **Available users** list, and click >. The warehouse users that you select are moved to the **Selected users** list.

If you want to include all existing warehouse users from the **Available users** list, click >>.

8. On the Warehouse Sources and Targets page, select the warehouse sources and warehouse targets to which the warehouse group will have access from the **Available warehouse source and targets** list, and click >. The warehouse sources and targets that you select are moved to the **Selected warehouse source and targets** list.

If you want the warehouse group to have access to all warehouse sources and targets in the **Available warehouse source and targets** list, click >>.

Tip: You can authorize warehouse groups to access warehouse sources and warehouse targets from the Define Warehouse Group notebook, or the Define Warehouse Source, or Define Warehouse Target notebook.

9. On the Processes page, select the processes to which the warehouse group will have access from the **Available processes** list, and click >. The processes that you select are moved to the **Selected processes** list.

If you want the warehouse group to have access to all processes in the **Available processes** list, click >>.

Tip: You can authorize warehouse groups to access processes from either the Define Warehouse Group notebook or the Define Process notebook.

10. Click **OK** to save the warehouse user group and close the notebook.

Chapter 3. Setting up 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 have decided to use.
- Define the sources to the Data Warehouse Center as warehouse sources.

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

Selecting Data Warehouse Center data sources

The Data Warehouse Center supports a wide variety of relational and nonrelational data sources. See Table 2 and Table 3 for a list of supported sources.

Table 2. Version and release levels of supported IBM warehouse sources

Source database	Version/Release
IMS	5.1
DB2 Universal Database for Windows NT	5.2 - 7.1
DB2 Universal Database Enterprise Extended Edition	5.2 - 7.1
DB2 Universal Database for OS/2	5.2 - 7.1
DB2 Universal Database for AS/400	3.7 - 5.1
DB2 Universal Database for AIX	5.2 - 7.1
DB2 Universal Database for Solaris Operating Environment	5.2 - 7.1
DB2 Universal Database for OS/390	5.1 - 7.1
IBM DataJoiner [®]	2.1.1
DB2 for VM	5.3.4 or later
DB2 for VSE	7.1

Table 3. Version and release levels of supported non-IBM warehouse sources

Source database	Windows NT	AIX
Informix	7.2.2 - 8.2.1	7.2.4 - 9.2.0
Oracle	7.3.2 - 8.1.5	8.1.5

Setting up warehouse sources

Table 3. Version and release levels of supported non-IBM warehouse sources (continued)

Source database	Windows NT	AIX
Microsoft [®] SQL Server	7.0	
Microsoft Excel	97	
Microsoft Access	97	
Sybase	11.5	11.9.2

You can populate your Data Warehouse Center warehouse with data from the following databases and files:

- Any DB2 family database
- Oracle
- Sybase
- Informix
- Microsoft SQL Server
- IBM DataJoiner

For more information about using DataJoiner with Data Warehouse Center, see “Defining warehouse sources for use with DataJoiner” on page 101.

- Multiple Virtual Storage (OS/390), Virtual Machine (VM), and local area network (LAN) files
- IMS and Virtual Storage Access Method (VSAM) (with Data Joiner Classic Connect)

For information about the versions and releases of these products, and the products’ prerequisites, see the *DB2 Warehouse Manager Installation Guide*.

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

The Data Warehouse Center also includes several Merant ODBC drivers that you can use to access non-IBM data. For information about installing these drivers, see *DB2 Universal Database Quick Beginnings* for your operating system.

Certain warehouse agents support certain sources. Table 4 on page 29 summarizes which warehouse agents support which sources.

Table 4. Warehouse agent support for sources

Data Source	Windows NT or Windows 2000 agent	AIX agent	Solaris Operating Environment agent	IBM Operating System/2 (OS/2) agent	AS/400	OS/390
DB2 family	✓	✓	✓	✓	✓	✓
DataJoiner	✓	✓				
Oracle	✓	✓	✓	✓		
Sybase	✓	✓	✓	✓		
Microsoft SQL Server	✓	✓	✓			
Informix	✓	✓	✓			
Generic ODBC driver	✓	✓	✓	✓		
Local file	✓	✓	✓	✓	✓	✓ ²
Remote file	✓	✓	✓	✓	✓	
IMS	✓	✓	✓ ¹	✓ ¹		✓ ³
VSAM	✓	✓	✓ ¹	✓ ¹		✓ ³

1. To access IMS or VSAM data when DataJoiner is on a remote workstation, catalog the node on which DataJoiner resides and catalog the DataJoiner database on the agent site.
2. Flat files are not accessible as ODBC sources for OS/390, but they can be used as a source for warehouse utilities.

Windows NT or Windows 2000

Table 5 on page 30 lists the supported data sources on Windows NT or Windows 2000 and explains what you need to do to connect to those sources.

Setting up warehouse sources

Table 5. Connectivity requirements for supported data sources on Windows NT or Windows 2000

Source database or file:	Prerequisite products for Windows NT or Windows 2000:	How to connect:
DB2 Universal Database databases	Database access program: None Source / agent connection: ODBC Client enabler program: DB2 Universal Database Version 7 server or a DB2 client	<ol style="list-style-type: none">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.
DB2 DRDA databases	Database access program: None Source / agent connection: ODBC Client enabler program: DB2 Connect [®]	<ol style="list-style-type: none">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.

Table 5. Connectivity requirements for supported data sources on Windows NT or Windows 2000 (continued)

Source database or file:	Prerequisite products for Windows NT or Windows 2000:	How to connect:
IMS (OS/390)	<p>Database access program: If you are not using the OS/390 agent, you must use one of the following programs:</p> <ul style="list-style-type: none"> • The CROSS ACCESS ODBC driver and DataJoiner Classic Connect • DataJoiner and DataJoiner Classic Connect <p>If you are using the OS/390 agent, Classic Connect</p> <p>Source / agent connection: If you are using the CROSS ACCESS ODBC driver, ODBC</p> <p>If you are using DataJoiner, TCP/IP or APPC</p> <p>Client enabler program: None</p>	<p>If you are using the CROSS ACCESS ODBC driver</p> <ol style="list-style-type: none"> 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. <p>If you are using DataJoiner, do the following procedure from the DataJoiner workstation:</p> <ol style="list-style-type: none"> 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. <p>From the agent site:</p> <ol style="list-style-type: none"> 1. Catalog the node on which DataJoiner resides. 2. Catalog the DataJoiner database.

Setting up warehouse sources

Table 5. Connectivity requirements for supported data sources on Windows NT or Windows 2000 (continued)

Source database or file:	Prerequisite products for Windows NT or Windows 2000:	How to connect:
VSAM (OS/390)	<p>Database access program: If you are not using the OS/390 agent, you must use one of the following programs:</p> <ul style="list-style-type: none"> • The CROSS ACCESS ODBC driver and DataJoiner Classic Connect • DataJoiner and DataJoiner Classic Connect <p>If you are using the OS/390 agent, Classic Connect</p> <p>Source / agent connection: If you are using the CROSS ACCESS ODBC driver, ODBC</p> <p>If you are using DataJoiner, TCP/IP or APPC</p> <p>Client enabler program: None</p>	<p>If you are using the CROSS ACCESS ODBC driver</p> <ol style="list-style-type: none"> 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. <p>If you are using DataJoiner, do the following procedure from the DataJoiner workstation:</p> <ol style="list-style-type: none"> 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. <p>From the agent site:</p> <ol style="list-style-type: none"> 1. Catalog the node on which DataJoiner resides. 2. Catalog the DataJoiner database.
OS/390 file	<p>Database access program: FTP or NFS</p> <p>Source / agent connection: TCP/IP (FTP or NFS)</p> <p>Client enabler program: None</p>	<p>Establish a link from the agent site to the host.</p> <p>For more information about accessing files using NFS, see "Accessing files with NFS" on page 59.</p> <p>For more information about accessing files using FTP, see "Accessing files with FTP" on page 59.</p>

Table 5. Connectivity requirements for supported data sources on Windows NT or Windows 2000 (continued)

Source database or file:	Prerequisite products for Windows NT or Windows 2000:	How to connect:
VM file	<p>Database access program: FTP or NFS</p> <p>Source / agent connection: TCP/IP (FTP or NFS)</p> <p>Client enabler program: None</p>	<p>Establish a link from the agent site to the host.</p> <p>For more information about accessing files using NFS, see "Accessing files with NFS" on page 59.</p> <p>For more information about accessing files using FTP, see "Accessing files with FTP" on page 59.</p>
Sybase	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: Sybase Open Client-Library 10.0.4 or later and the appropriate Sybase Net-Library.</p> <p>For Intel® systems, Sybase Open Client Library Version 10.03 (or later) and the appropriate Sybase Net-Library</p> <p>For Alpha systems, Sybase Open Client Library Version 11.01 (or later) and the appropriate Sybase Net-Library</p>	<ol style="list-style-type: none"> 1. Install the Open Client on the agent site. 2. Catalog the remote database according to the client enabler instructions. 3. Identify the user ID and password that has access to the source database.

Setting up warehouse sources

Table 5. Connectivity requirements for supported data sources on Windows NT or Windows 2000 (continued)

Source database or file:	Prerequisite products for Windows NT or Windows 2000:	How to connect:
Oracle	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: Oracle SQL*Net V2</p> <p>Database client requirements</p> <p>For Oracle Version 7: The appropriate DLLs for the current version of SQL*Net, plus OCIW32.DLL. For example, SQL*Net 2.3 requires ORA73.DLL, CORE35.DLL, NLSRTL32.DLL, CORE350.DLL and OCIW32.DLL.</p> <p>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.</p> <p>On Intel systems, install the appropriate DLLs for the Oracle Net8 Client (such as Ora804.DLL, PLS804.DLL and OCI.DLL) on your path.</p>	<ol style="list-style-type: none"> 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.
Informix	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: For Informix 5, 6, and 7.x, i-connect 7.2 For Informix 7.x and 9.x, i-connect 9.x</p> <p>Database client requirements Informix-Connect for Windows Platforms 2.x or Informix-Client Software Developer's Kit for Windows Platforms 2.x</p>	<ol style="list-style-type: none"> 1. Install i-connect on the agent site. 2. Catalog the remote database according to the client enabler instructions. 3. Identify the user ID and password that has access to the source database.

Table 5. Connectivity requirements for supported data sources on Windows NT or Windows 2000 (continued)

Source database or file:	Prerequisite products for Windows NT or Windows 2000:	How to connect:
Microsoft SQL Server	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: For access to a Version 6.0 DBMS, Microsoft SQL Server DB-Library and Net-Library Version 6.0</p> <p>For access to a Version 7.0 DBMS, Microsoft SQL Server DB-Library and Net-Library Version 7.0</p>	<ol style="list-style-type: none"> 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.
Local file	<p>Database access program: None</p> <p>Source / agent connection: TCP/IP</p> <p>Client enabler program: None</p>	<p>If required, supply a pre-access command, a post-access command, or both.</p>
Remote file	<p>Database access program: None</p> <p>Source / agent connection: TCP/IP</p> <p>Client enabler program: None</p>	<ol style="list-style-type: none"> 1. Obtain the FTP server or mount string. 2. Identify the user ID and password with access to the source files. <p>For more information about accessing files using NFS, see "Accessing files with NFS" on page 59.</p> <p>For more information about accessing files using FTP, see "Accessing files with FTP" on page 59.</p>
Microsoft Access	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: None</p>	<ol style="list-style-type: none"> 1. Use the generic ODBC connect string. 2. See the Microsoft Access help topics for a mapping of the ANSI SQL data types supported by Microsoft Access. <p>For detailed instructions about creating a warehouse that uses Microsoft Access as a source database, see "Microsoft Access" on page 80.</p>

Setting up warehouse sources

Table 5. Connectivity requirements for supported data sources on Windows NT or Windows 2000 (continued)

Source database or file:	Prerequisite products for Windows NT or Windows 2000:	How to connect:
Microsoft Excel	Database access program: None Source / agent connection: ODBC Client enabler program: None	<ol style="list-style-type: none">1. Use the generic ODBC connect string.2. See the Microsoft Excel help topics for a mapping of the ANSI SQL data types supported by Microsoft Excel. <p>For detailed instructions about creating a warehouse that uses Microsoft Excel as a source database, see "Microsoft Excel" on page 84.</p>

Notes:

¹ If you are using the Sybase client on the Windows 95 or Windows 98 operating systems, you must include the following statement as the last statement in the autoexec.bat file:

```
CALL C:\SYBASE\BIN\WSYBSET.BAT
```

² See the *Warehouse Manager Installation Guide* for more information about the OS/390 agent.

AIX

Table 6 on page 37 lists the supported data sources on AIX and explains what you need to do to connect to those sources.

There are two versions of the AIX warehouse agent: one for ODBC access and one for CLI access. The non-DB2 databases listed in the following table require that you install the ODBC version of the warehouse agent.

To verify which version of the warehouse agent that you have installed, type the following command:

```
ls -l /usr/bin/IWH2AGNT
```

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

If you are using a Data Warehouse Center AIX agent that is linked to access Merant 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/lpp/db2_07_01/lib/db2_36.o
```

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

```
[SAMPLE] Driver=/usr/lpp/db2_07_01/lib/db2_36.o
Description=DB2 ODBC Database
Database=SAMPLE
```


For more information about installing and configuring the warehouse agent, see the *DB2 Warehouse Manager Installation Guide*.

Table 6. Connectivity requirements for supported data sources on AIX

Source database or file:	Prerequisite products for AIX:	How to connect:
DB2 Universal Database databases	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: DB2 UDB Version 7 server or a DB2 client</p>	<ol style="list-style-type: none"> 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.
DB2 DRDA databases	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: DB2 Connect</p>	<ol style="list-style-type: none"> 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.
IMS (OS/390)	<p>Database access program: If you are not using the OS/390 agent, DataJoiner and DataJoiner Classic Connect If you are using the OS/390 agent, Classic Connect</p> <p>Source / agent connection: TCP/IP or APPC</p> <p>Client enabler program: None</p>	<p>From the DataJoiner workstation:</p> <ol style="list-style-type: none"> 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. <p>From the agent site:</p> <ol style="list-style-type: none"> 1. Catalog the node on which DataJoiner resides. 2. Catalog the DataJoiner database.

Setting up warehouse sources

Table 6. Connectivity requirements for supported data sources on AIX (continued)

Source database or file:	Prerequisite products for AIX:	How to connect:
VSAM (OS/390)	<p>Database access program: If you are not using the OS/390 agent, DataJoiner and DataJoiner Classic Connect</p> <p>If you are using the OS/390 agent, Classic Connect</p> <p>Source / agent connection: TCP/IP or APPC</p> <p>Client enabler program: None</p>	<p>From the DataJoiner workstation:</p> <ol style="list-style-type: none"> 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. <p>From the agent site:</p> <ol style="list-style-type: none"> 1. Catalog the node on which DataJoiner resides. 2. Catalog the DataJoiner database.
OS/390 file	<p>Database access program: FTP or NFS</p> <p>Source / agent connection: TCP/IP (FTP or NFS)</p> <p>Client enabler program: None</p>	<p>Establish a link from the agent site to the host.</p> <p>For more information about accessing files using NFS, see "Accessing files with NFS" on page 59.</p> <p>For more information about accessing files using FTP, see "Accessing files with FTP" on page 59.</p>
VM file	<p>Database access program: FTP or NFS</p> <p>Source / agent connection: TCP/IP (FTP or NFS)</p> <p>Client enabler program: None</p>	<p>Establish a link from the agent site to the host.</p> <p>For more information about accessing files using NFS, see "Accessing files with NFS" on page 59.</p> <p>For more information about accessing files using FTP, see "Accessing files with FTP" on page 59.</p>

Table 6. Connectivity requirements for supported data sources on AIX (continued)

Source database or file:	Prerequisite products for AIX:	How to connect:
Sybase	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: Intersolv Version 3.6 Driver Manager and Sybase driver</p> <p>Database client requirements In a non-DCE environment: (ibsyb15 ODBC driver): libct library</p> <p>In a DCE environment (ibsyb1115 ODBC driver): Sybase 11.1 client library libct_r</p>	<ol style="list-style-type: none"> 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. 3. Install the Open Client.
Oracle	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: Intersolv Version 3.6 Driver Manager and Oracle driver</p> <p>Database client requirements For Oracle Version 7: Oracle7 SQL*Net and Oracle7 SQL*Net shared library (built by the genclntsh script)</p> <p>For Oracle Version 8: Oracle8 Net8 and the Oracle8 SQL*Net shared library (built by the genclntsh8 script)</p>	<ol style="list-style-type: none"> 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 warehouse sources

Table 6. Connectivity requirements for supported data sources on AIX (continued)

Source database or file:	Prerequisite products for AIX:	How to connect:
Informix	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: Intersolv Version 3.6 Driver Manager and Informix driver</p> <p>Database client requirements Informix-Connect and ESQL/C version 9.1.4 or later</p>	<ol style="list-style-type: none"> 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.
Microsoft SQL Server	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: Data Warehouse Center ODBC driver manager¹</p>	<ol style="list-style-type: none"> 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.
Local file	<p>Database access program: None</p> <p>Source / agent connection: TCP/IP</p> <p>Client enabler program: None</p>	<p>If required, supply a pre-access command, a post-access command, or both.</p>
Remote file	<p>Database access program: None</p> <p>Source / agent connection: TCP/IP</p> <p>Client enabler program: None</p>	<ol style="list-style-type: none"> 1. Obtain the FTP server or mount string. 2. Identify the user ID and password with access to the source files. <p>For more information about accessing files using NFS, see “Accessing files with NFS” on page 59.</p> <p>For more information about accessing files using FTP, see “Accessing files with FTP” on page 59.</p>

Table 6. Connectivity requirements for supported data sources on AIX (continued)

Source database or file:	Prerequisite products for AIX:	How to connect:
Notes:		
¹ The Data Warehouse Center ODBC drivers and driver manager are the DataDirect Connect ODBC drivers and driver manager provided by Merant.		
² The Open Client is required for connecting to Sybase sources on Windows NT or Windows 2000 operating systems.		
³ See the <i>Warehouse Manager Installation Guide</i> for more information about the OS/390 agent.		

Solaris Operating Environment

Table 7 on page 42 lists the supported data sources on the Solaris Operating Environment and explains what you need to do to connect to those sources.

There are two versions of the Solaris Operating Environment warehouse agent: one for ODBC access and one for CLI access. The non-DB2 databases listed in the following table require that you install the ODBC version of the warehouse agent.

To verify which version of the warehouse agent that you have installed, type the following command:

```
ls -l /usr/bin/IWH2AGNT
```

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

If you are using a Data Warehouse Center agent for the Solaris Operating Environment that is linked to access Merant 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/V7.1/lib/libdb2_36.so
```

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

```
[SAMPLE] Driver=/opt/IBMdb2/V7.1/lib/libdb2_36.so
Description=DB2 ODBC Database
Database=SAMPLE
```

For more information about installing and configuring the warehouse agent, see the *DB2 Warehouse Manager Installation Guide*.

Setting up warehouse sources

Table 7. Connectivity requirements for supported data sources on the Solaris Operating Environment

Source database or file:	Prerequisite products for the Solaris Operating Environment:	How to connect:
DB2 Universal Database	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: DB2 UDB Version 7 server or a DB2 client</p>	<ol style="list-style-type: none"> 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.
DB2 DRDA databases	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: DB2 Connect</p>	<ol style="list-style-type: none"> 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.
IMS (OS/390)	<p>Database access program: If you are not using the OS/390 agent, DataJoiner and DataJoiner Classic Connect If you are using the OS/390 agent, Classic Connect</p> <p>Source / agent connection: ODBC (to DataJoiner database)</p> <p>Client enabler program: None</p>	<p>From the DataJoiner workstation:</p> <ol style="list-style-type: none"> 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. <p>From the agent site:</p> <ol style="list-style-type: none"> 1. Catalog the node on which DataJoiner resides. 2. Catalog the DataJoiner database.
VSAM (OS/390)	<p>Database access program: If you are not using the OS/390 agent DataJoiner and DataJoiner Classic Connect If you are using the OS/390 agent, Classic Connect</p> <p>Source / agent connection: ODBC (to DataJoiner database)</p> <p>Client enabler program: None</p>	<p>From the agent site:</p> <ol style="list-style-type: none"> 1. Catalog the node on which DataJoiner resides. 2. Catalog the DataJoiner database.

Table 7. Connectivity requirements for supported data sources on the Solaris Operating Environment (continued)

Source database or file:	Prerequisite products for the Solaris Operating Environment:	How to connect:
OS/390 file	<p>Database access program: FTP or NFS</p> <p>Source / agent connection: TCP/IP (FTP or NFS)</p> <p>Client enabler program: None</p>	<p>Establish a link from the agent site to the host.</p> <p>For more information about accessing files using NFS, see “Accessing files with NFS” on page 59.</p> <p>For more information about accessing files using FTP, see “Accessing files with FTP” on page 59.</p>
VM file	<p>Database access program: FTP or NFS</p> <p>Source / agent connection: TCP/IP (FTP or NFS)</p> <p>Client enabler program: None</p>	<p>Establish a link from the agent site to the host.</p> <p>For more information about accessing files using NFS, see “Accessing files with NFS” on page 59.</p> <p>For more information about accessing files using FTP, see “Accessing files with FTP” on page 59.</p>
Sybase	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: For Sybase 11, Intersolv Version 3.6 Driver Manager and Sybase driver.</p> <p>Database client requirements In a DCE environment (ibsyb1115 ODBC driver): Sybase 11.1 client library libct_r. In a non-DCE environment (ibsyb15 ODBC driver): libct library</p>	<ol style="list-style-type: none"> 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. 3. Install the Open Client.

Setting up warehouse sources

Table 7. Connectivity requirements for supported data sources on the Solaris Operating Environment (continued)

Source database or file:	Prerequisite products for the Solaris Operating Environment:	How to connect:
Oracle	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: For Oracle Version 7.3.2, Intersolv Version 3.6 Driver Manager and Oracle driver.</p> <p>Database client requirements For Oracle Version 7: Oracle7 SQL*Net and Oracle7 SQL*Net shared library (built by the genclntsh script). For Oracle Version 8: Oracle8 Net8 and the Oracle8 SQL*Net shared library (built by the genclntsh8 script).</p>	<ol style="list-style-type: none"> 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.
Informix	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: Intersolv Version 3.6 Driver Manager and Informix driver.</p> <p>Database client requirements Informix-Connect and ESQL/C version 9.1.3 or later.</p>	<ol style="list-style-type: none"> 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.
Microsoft SQL Server	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: Data Warehouse Center ODBC Driver Manager¹</p>	<ol style="list-style-type: none"> 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.

Table 7. Connectivity requirements for supported data sources on the Solaris Operating Environment (continued)

Source database or file:	Prerequisite products for the Solaris Operating Environment:	How to connect:
Local file	<p>Database access program: None</p> <p>Source / agent connection: TCP/IP</p> <p>Client enabler program: None</p>	If required, supply a pre-access command, a post-access command, or both.
Remote file	<p>Database access program: None</p> <p>Source / agent connection: TCP/IP</p> <p>Client enabler program: None</p>	<ol style="list-style-type: none"> 1. Obtain the FTP server or mount string. 2. Identify the user ID and password with access to the source files. <p>For more information about accessing files using NFS, see "Accessing files with NFS" on page 59.</p> <p>For more information about accessing files using FTP, see "Accessing files with FTP" on page 59.</p>

Notes:

¹ The Data Warehouse Center ODBC drivers and driver manager are the DataDirect Connect ODBC drivers and driver manager provided by Merant.

² The Open Client is required for connecting to Sybase sources on Windows NT or Windows 2000 operating systems.

³ See the *Warehouse Manager Installation Guide* for more information about the OS/390 agent.

OS/2

Table 8 on page 46 lists the supported data sources on OS/2 and explains what you need to do to connect to those sources.

Setting up warehouse sources

Table 8. Connectivity requirements for supported data sources on OS/2

Source database or file:	Prerequisite products for OS/2:	How to connect:
DB2 Universal Database databases	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: DB2 UDB Version 7 server or a DB2 client</p>	<ol style="list-style-type: none"> 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.
DB2 DRDA databases	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: DB2 Connect</p>	<ol style="list-style-type: none"> 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.
LAN BLOB file	<p>Database access program: None</p> <p>Source / agent connection: TCP/IP</p> <p>Client enabler program: None</p>	<ol style="list-style-type: none"> 1. Obtain the FTP server or mount string. 2. Identify the user ID and password with access to the source files. <p>For more information about accessing files using NFS, see "Accessing files with NFS" on page 59.</p> <p>For more information about accessing files using FTP, see "Accessing files with FTP" on page 59.</p>
IMS (OS/390)	<p>Database access program: If you are not using the OS/390 agent, DataJoiner and DataJoiner Classic Connect If you are using the OS/390 agent, Classic Connect</p> <p>Source / agent connection: ODBC (to DataJoiner database)</p> <p>Client enabler program: None</p>	<p>If you are using DataJoiner, do the following procedure from the agent site:</p> <ol style="list-style-type: none"> 1. Catalog the node on which DataJoiner resides. 2. Catalog the DataJoiner database.

Table 8. Connectivity requirements for supported data sources on OS/2 (continued)

Source database or file:	Prerequisite products for OS/2:	How to connect:
VSAM (OS/390)	<p>Database access program: DataJoiner and DataJoiner Classic Connect</p> <p>Source / agent connection: ODBC (to DataJoiner database)</p> <p>Client enabler program: None</p>	<p>From the agent site:</p> <ol style="list-style-type: none"> 1. Catalog the node on which DataJoiner resides. 2. Catalog the DataJoiner database.
OS/390 file	<p>Database access program: FTP or NFS</p> <p>Source / agent connection: TCP/IP (FTP or NFS)</p> <p>Client enabler program: None</p>	<p>Establish a link from the agent site to the host.</p> <p>For more information about accessing files using NFS, see “Accessing files with NFS” on page 59.</p> <p>For more information about accessing files using FTP, see “Accessing files with FTP” on page 59.</p>
VM file	<p>Database access program: FTP or NFS</p> <p>Source / agent connection: TCP/IP (FTP or NFS)</p> <p>Client enabler program: None</p>	<p>Establish a link from the agent site to the host.</p> <p>For more information about accessing files using NFS, see “Accessing files with NFS” on page 59.</p> <p>For more information about accessing files using FTP, see “Accessing files with FTP” on page 59.</p>
Sybase	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: Sybase Open Client Library Version 10 for OS/2, Intersolv Version 3.6 Driver Manager, and Sybase driver</p>	<ol style="list-style-type: none"> 1. Install the Open Client on the agent site. 2. Catalog the remote database according to the client enabler instructions. 3. Identify the user ID and password that has access to the source database.

Setting up warehouse sources

Table 8. Connectivity requirements for supported data sources on OS/2 (continued)

Source database or file:	Prerequisite products for OS/2:	How to connect:
Oracle	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: Oracle SQL*Net V2.1.4 for OS/2, Intersolv Version 3.6 Driver Manager and Oracle driver</p>	<ol style="list-style-type: none"> 1. Install SQL*NET on the agent site. 2. Catalog the remote database according to the client enabler instructions. 3. Identify the user ID and password with access to the source database.
Informix	<p>Database access program: None</p> <p>Source / agent connection: ODBC</p> <p>Client enabler program: Informix-Net for OS/2, Intersolv Version 3.6 Driver Manager and Informix driver</p>	<ol style="list-style-type: none"> 1. Install Informix-Net on the agent site. 2. Catalog the remote database according to the client enabler instructions. 3. Identify the user ID and password that has access to the source database.
Local file	<p>Database access program: None</p> <p>Source / agent connection: TCP/IP</p> <p>Client enabler program: None</p>	<p>If required, supply a pre-access command, a post-access command, or both.</p>
Remote file	<p>Database access program: None</p> <p>Source / agent connection: TCP/IP</p> <p>Client enabler program: None</p>	<ol style="list-style-type: none"> 1. Obtain the FTP server or mount string. 2. Identify the user ID and password with access to the source files. <p>For more information about accessing files using NFS, see "Accessing files with NFS" on page 59.</p> <p>For more information about accessing files using FTP, see "Accessing files with FTP" on page 59.</p>

Notes:

¹ See the *Warehouse Manager Installation Guide* for more information about the OS/390 agent.

Setting up access to DB2 Universal Database source databases

You can use any DB2 Universal Database database as a source database for your warehouse. For more information about using DB2 Universal Database, see *DB2 Universal Database Installation and Configuration Supplement*.

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 source databases

The system administrator of the source system must set up a user ID with the following privileges:

- BINDADD (database-level privilege)
- CONNECT (database-level privilege)

Additionally, the following SYSIBM system tables require explicit SELECT privilege:

- SYSIBM.SYSTABLES
- SYSIBM.SYSCOLUMNS
- SYSIBM.SYSDBAUTH
- SYSIBM.SYSTABAUTH
- SYSIBM.SYSINDEXES
- SYSIBM.SYSRELS
- SYSIBM.SYSTABCONST

Any tables that you want to access also require explicit SELECT privilege.

Establishing connectivity to DB2 Universal Database source databases

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

1. Set up communications to the database if the database is remote.
2. Catalog the node if the database is remote.
3. Catalog the database.
4. Register the database as a ODBC system DSN if you are using the Windows NT, Windows 2000, or OS/2 warehouse agent, or the version of the AIX or Solaris Operating Environment warehouse agent that uses ODBC. If you are using the AIX, OS/390, or Solaris Operating Environment 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 warehouse sources

Setting up access to DB2 DRDA source databases

To access DB2 Distributed Relational Database Architecture™ (DRDA®) source databases using a gateway:

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.

To access DB2 Distributed Relational Database Architecture™ (DRDA®) source databases through a direct connection, establish connectivity to the source databases. See “Establishing connectivity to DB2 DRDA source databases” on page 51 for step-by-step instructions.

Prerequisite products

If you want to access data from the one of the following source databases, you need a gateway site. Configure the site for DRDA:

- DB2 Universal Database for AS/400®
- DB2 Universal Database for OS/390
- DB2 for VM
- DB2 for VSE

For more information about DRDA, see the *Distributed Relational Database Architecture Connectivity Guide*.

For information about DB2 Connect, see the following books:

- *DB2 Connect Quick Beginnings* for your product edition
- *DB2 Connect User's Guide*
- *Installation and Configuration Supplement*

Defining privileges to DB2 DRDA source databases

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 OS/390, 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 AS/400, the user ID must have CHANGE authority or higher on the NULLID collection

Setting up the DB2 Connect gateway site

After your user ID is set up with the required privileges, perform the following tasks at the gateway site:

1. For Windows NT or Windows 2000, 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.

Establishing connectivity to DB2 DRDA source databases

You need to establish connectivity to the 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 as described in the *DB2 Connect User's Guide*. Each type of client requires only one bind.

Setting up warehouse sources

Tip: On Windows 32-bit systems, you can use the DB2 UDB Client Configuration Assistant to complete this task. See *IBM DB2 Universal Database: Quick Beginnings for Windows* for information about registering databases with the DB2 UDB Client Configuration Assistant.

Connectivity requirements for remote databases from the AS/400 agent

You can access remote databases via the AS/400 agent through Systems Network Architecture (SNA) 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 AS/400
- DB2 Universal Database for OS/390

You can connect from the AS/400 agent to a remote database when the following conditions are met:

- The SNA connection to the remote database is correct.
- The remote database is cataloged in the AS/400 Relational Database Directory.

Tip: You should be able to 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 AS/400 agent.
- You can query the remote database from the AS/400 interactive SQL facility (STRSQL).

For more information about using DRDA through SNA to connect to different types of databases, see the Redbook, *Distributed Relational Database Cross Platform Connectivity and Application*. You can also view this book online at <http://www.redbooks.ibm.com>.

For more information about using DRDA over TCP/IP see the *Warehouse Manager Installation Guide*

Establishing connectivity to local and remote databases from the AS/400 agent

You must catalog the names of local and remote database that you plan to use as a warehouse source or target in the AS/400 Relational Database directory on your agent site. You must also catalog these database names on the remote workstation that your agent accesses.

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

Setting up warehouse sources

agent will access. Likewise, 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.

For example, 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 AS/400 agent. The database named Expenses is located on the remote workstation that the agent will access. Table 9 describes how Fred should catalog each database on each workstation.

Table 9. How to catalog local and remote database names

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

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

To add a database name entry to the AS/400 Relational Database directory, enter the following command at an AS/400 command prompt:

```
ADDRDBDIRE databasename locationname
```

where `databasename` is the name of your AS/400 database, and `locationname` is the location name of your AS/400 workstation. You must specify whether the database is local or remote.

Ensure that you 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.

You can also use the **WRKRDBDIRE** command to view, add, change, and remove remote Relational Database directory entries. To use this command, enter the command at an AS/400 command prompt. A list of currently defined remote database names is displayed. A set of options is displayed at the top of the window.

Setting up warehouse sources

For more information, see the AS/400 online help for each of these commands.

Connectivity requirements for remote databases from the OS/390 agent

You can access remote databases from the OS/390 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 OS/390
- DB2 Universal Database for AS/400
- Oracle, Sybase, Informix, and other similar database products, when DataJoiner[®] is used

For more information about using DRDA to connect to different types of databases, see the *IBM Distributed Relational Database Architecture Reference*.

The OS/390 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. For more information on using Classic Connect with IMS and VSAM databases, see “Appendix F. Using Classic Connect with the Data Warehouse Center” on page 403.

Using DataJoiner

When you use the OS/390 agent to access DataJoiner Version 2 and other DataJoiner data sources, Systems Network Architecture (SNA) LU 6.2 is required as the communications protocol. Although DataJoiner supports TCP/IP as an application requester in Version 2.1 and Version 2.1.1, it does not have an application server. Also, DataJoiner can not be used as a target for OS/390, because DataJoiner does not support two-phase commit from DRDA, a requirement of DB2 Universal Database for OS/390.

For more information about using DRDA through SNA to connect to different types of databases, see the IBM redbook, *Distributed Relational Database Cross Platform Connectivity*. You can view this book online at <http://www.redbooks.ibm.com>.

Defining a DB2 warehouse source

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.

The Data Warehouse Center supports source 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 does not support 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

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. For more information, see the online help.

To help you 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.

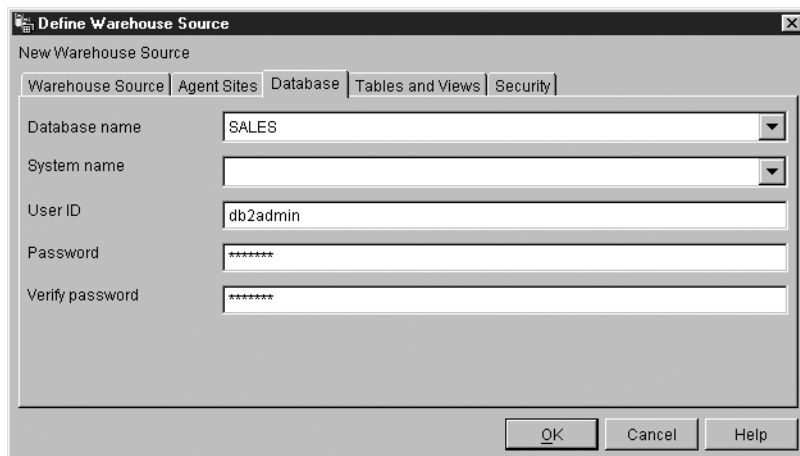
When you define a warehouse source for a DB2 for VM database, which is accessed through a DRDA gateway, the following restrictions apply to the use of CLOB and BLOB data types:

- You cannot use the Sample Contents function to view data of CLOB and BLOB data types.
- You cannot use columns of CLOB and BLOB data types with an SQL step. This is a known restriction on the DB2 for VM Version 5.2 server in which LOB objects cannot be transmitted using DRDA to a DB2 Version 7 client.

To define a DB2 Universal Database warehouse source:

Setting up warehouse sources

1. Right-click the **Warehouse Sources** folder.
2. Click **Define** → **DB2 Family**, then click the version of DB2 Universal Database for your operating system (such as **DB2 UDB for Windows NT**) from the menu.
The Define Warehouse Source notebook opens.
3. In the **Name** field, type the business name for the warehouse source.
You will use this name to refer to your warehouse source throughout the Data Warehouse Center.
4. In the **Administrator** field, type the contact for the warehouse source.
5. In the **Description** field, type a short description of the data.
6. Click the **Agent sites** page. The sites that you can select are displayed in the **Available Agent Sites** list.
7. From the **Available agent sites** list, select the name of the site to which you want to give access to your warehouse source and click **>**. The agent site is added to the **Selected agent sites** list. The site can now access the warehouse source.
To include all of the items in the **Available agent sites** list, click **>>**.
8. Click the **Database** tab.



The screenshot shows a dialog box titled "Define Warehouse Source" with a subtitle "New Warehouse Source". It has five tabs: "Warehouse Source", "Agent Sites", "Database", "Tables and Views", and "Security". The "Database" tab is selected. The dialog contains the following fields:

- Database name:** A dropdown menu with "SALES" selected.
- System name:** A dropdown menu.
- User ID:** A text field containing "db2admin".
- Password:** A text field containing "*****".
- Verify password:** A text field containing "*****".

At the bottom right, there are three buttons: "OK", "Cancel", and "Help".

9. In the **Database name** field, type the name of the physical database. If you are defining a warehouse source on OS/2, type the data source name in uppercase letters. If you are defining an OS/390 data source, this is the location name of your database.
10. In the **User ID** field, type a user ID that has access to the database.
11. In the **Password** field, type the password for the user ID that will access the database.
12. In the **Verify Password** field, type the password again.

13. Click the **Agent sites** page. The sites that you can select are displayed in the **Available Agent Sites** list.
14. From the **Available agent sites** list, select the name of the site to which you want to give access to your warehouse source, and click >. The agent site is added to the **Selected agent sites** list. The site can now access the warehouse source.
To include all of the items in the **Available agent sites** list, click >>.
15. Click the **Tables and views** tab.
Because the tables are in a DB2 database, you can import the table definitions from DB2 Universal Database rather than defining them manually.
On OS/400® Version 4 Release 2 and Version 4 Release 3, you must select the **View** folder to import system tables.
16. Expand the **Tables** folder.
The Filter window opens.
17. Click **OK**.
The Data Warehouse Center displays a progress window. The import might take a while.
If the warehouse source 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 source has three agent sites selected: Default Agent, AIX Agent, and OS/390™ Agent. The warehouse server uses the agent site named AIX Agent for the import.
After the import finishes, the Data Warehouse Center lists the imported objects in the **Available tables and views** list.
18. From the **Available tables and views** list, select the tables and views that you want to include in your warehouse source and click >.
The tables and views that you selected are moved to the **Selected tables and views** list.
19. Click the **Security** tab.
20. Select a warehouse group to grant users in the group the ability to create steps that use this warehouse source.
21. Click >.
The warehouse group moves to the **Selected security groups** list.
Accept the rest of the values in the notebook. For more information about the values, see "Warehouse Source" in the online help.
22. Click **OK** to save your changes and close the Define Warehouse Sources notebook.
23. Click **Warehouse** → **Properties** to open the Properties notebook for the Data Warehouse Center.

Setting up warehouse sources

24. Click the **Processing Options** tab.
25. In the **Count** field, type the value for the number of times that you want to retry the extract, or use the up and down arrows on the button to the right of the field to scroll through a list of values that you can select.
26. Click **OK** to save your changes and close the Properties notebook for the Data Warehouse Center.

Accessing remote files

There are several ways that a step can access files on a remote workstation. This section lists some of the access methods that you can set up for your step.

Accessing files through a Windows NT or Windows 2000 file server

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 or Windows 2000 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.

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 VISWHSE 3.6 32-bit Textfile (*.*)).

Accessing files with NFS

Another way to access files from an agent site is to use 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, as described in “Accessing files through a Windows NT or Windows 2000 file server” on page 58.

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 VISWHSE 3.6 32-bit Textfile (*.*)).

Accessing files with FTP

You can use FTP to access data files on a remote workstation. As a guideline, use FTP if the file 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 Copy file using FTP instead of FTP (see “Accessing data files with Copy file using FTP” on page 60).

If you plan to use the FTP commands GET or PUT to transfer files from an OS/390 host to another remote host, you need to change the account information in the JCL template for your OS/390 system. For information about changing the Data Warehouse Center template for FTP support on OS/390 systems, see *Warehouse Manager Installation Guide*.

To use FTP to access a data file:

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.

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.

Setting up warehouse sources

If you are having problems accessing a remote file on a secure UNIX® or UNIX System Services system, verify that you have write access to the \$HOME directory of the user specified in the agent site definition.

For example, the hostname 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
```

Accessing data files with Copy file using FTP

You can use Copy file using FTP to access data files on a remote workstation. Use Copy file using FTP if the file 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 Copy file using FTP.

To use Copy file using FTP 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:
 - a. Define the first step to use the Copy file using FTP warehouse program.
Use this step to copy the file to the agent site.
 - b. 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.

Defining a file source

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

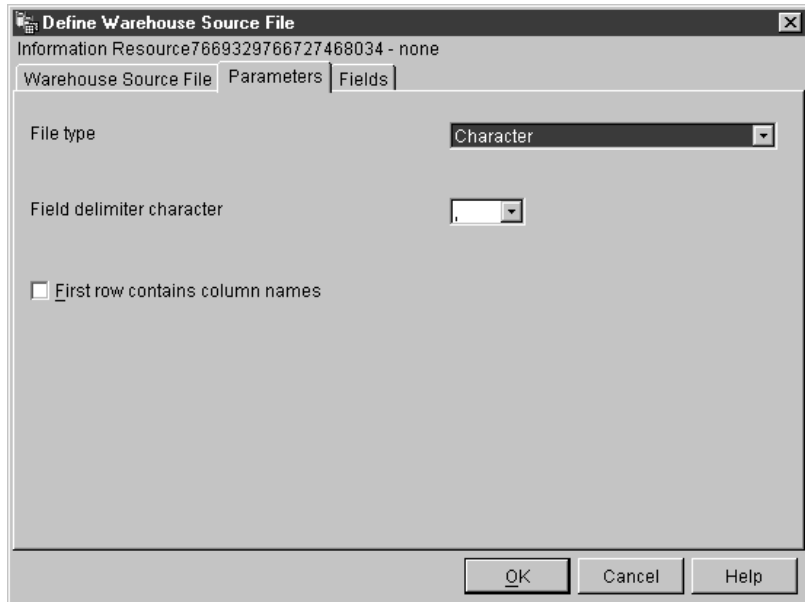
You cannot view data in Local File or Remote File warehouse sources before you define the file to Data Warehouse Center.

To define a file source:

1. Right-click the **Warehouse Sources** folder.
2. Click **Define** → **Flat File**, then click **Local File** or **Remote File**.
The Warehouse Source notebook opens.
3. In the **Name** field, type the business name for the warehouse source.
4. In the **Administrator** field, type the name of a contact for the warehouse source.
5. In the **Description** field, type a short description of the data.

Setting up warehouse sources

6. Click the **Agent sites** tab. The sites that you can select are displayed in the **Available Agent Sites** list.
7. From the **Available agent sites** list, select the name of the site to which you want to give access to your warehouse source, and click **>**. The agent site is added to the **Selected agent sites** list. The site can now be used to access the warehouse source.
8. Click the **Files** tab.
9. Right-click the blank area of the **Files** list, and click **Define**.
The Define Warehouse Source File notebook opens.
10. In the **File name** field, type the fully qualified path and file.
The name of the file must not contain spaces. On a UNIX system, file names are case-sensitive.
11. Optional: In the **Description** field, type a short description of the file.
12. Optional: In the **Notes** field, type notes about the file.
13. Optional: In the **Business name** field type a business name for the file.
14. Click the **Parameters** tab.



15. From the **File type** list, select a file type.
16. In the **Field delimiter character** field, specify a character to be used as a delimiter. This field is available only when **Character** is selected from the **File type** list.
17. Select the **First row contains column names** check box if the first row of the file contains column names.

Setting up warehouse sources

The Data Warehouse Center will ignore the column names and extract data starting from the second row of the file, which contains data.

If you are defining a remote file warehouse source, specify one of the following FTP transfer formats.

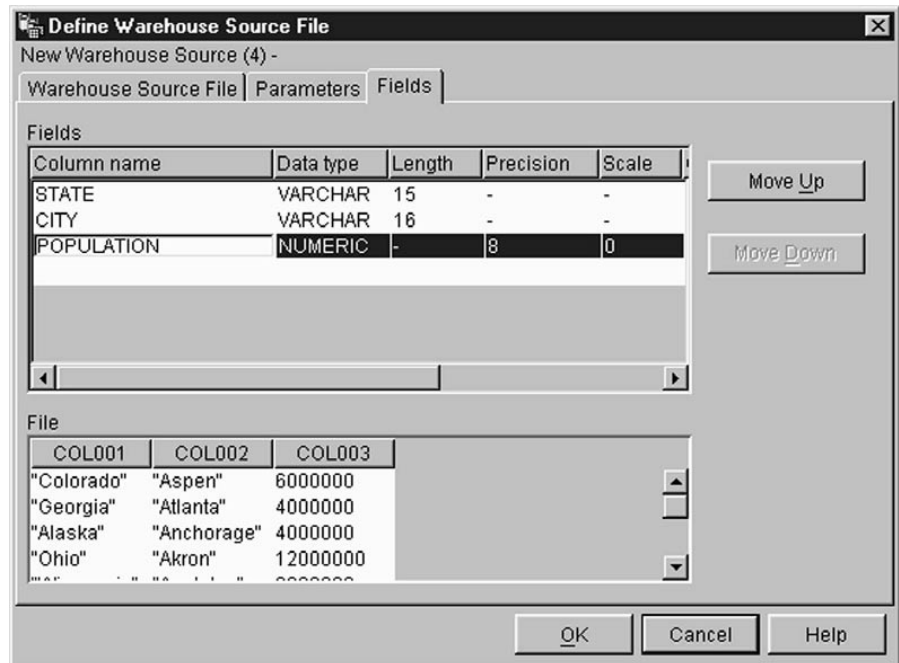
- Select **ASCII** to transfer files in ASCII format.
- Select **Binary** to transfer files in binary format.

18. Click the **Fields** tab.

The Data Warehouse Center reads the file that you specified on the Warehouse Source File page. It defines columns based on the fields in the file, and displays the column definitions in the **Fields** list. It displays sample data in the **File** preview area. Up to 10 rows of sample data are displayed. You can scroll to see all the sample data.

19. If you want to change the column names that the Data Warehouse Center generates, double-click the column name.
20. Type the new name for the column.
21. Press **Enter**.

The new name is displayed in the **Column name** field. In the following figure, COL001 was renamed to STATE.



22. Click **OK**.

The File notebook closes.

23. If you need to specify a pre-or post-access command, click the **Advanced** button on the Files page of the warehouse sources notebook.
The **Advanced** window opens.
24. In the **Pre-access command** field, type the command that you use to access a local file.
25. In the **Post-access command** field, type the command that you use after you access the file.
26. Click **OK** to save your changes and close the window.
27. Click the **Security** tab.
28. Select a warehouse group to grant users in the group the ability to create steps that use this warehouse source.
29. Click **>**. The warehouse group moves to the **Selected security groups** list.
30. Click **OK** to save your changes and close the Define Warehouse Source notebook.
31. Click **Warehouse** —> **Properties** to open the Properties notebook for the Data Warehouse Center.
32. Click the **Processing Options** tab.
33. In the **Count** field, type the value for the number of times that you want to retry the extract, or use the up and down arrows on the button to the right of the field to scroll through a list of values that you can select.
34. Click **OK** to save your changes and close the Properties notebook for the Data Warehouse Center.

Accessing non-DB2 database warehouse sources

You can access data from non-DB2 sources. You must define these sources before you create any steps that will access them.

There might be more than one way to access some sources. For example, you can use the Data Warehouse Center ODBC drivers to access many of the non-DB2 databases. However, you can also use DataJoiner to access the databases. For more information about using DataJoiner with the Data Warehouse Center, see “Defining warehouse sources for use with DataJoiner” on page 101.

Setting up non-DB2 database warehouse sources on Windows NT

The following sections describe how to set up Informix, Sybase, Oracle, Microsoft SQL Server, Microsoft Access, Microsoft Excel, IMS, and VSAM sources on Windows NT.

Informix

This section contains the steps for the following tasks:

- Configuring your Informix client

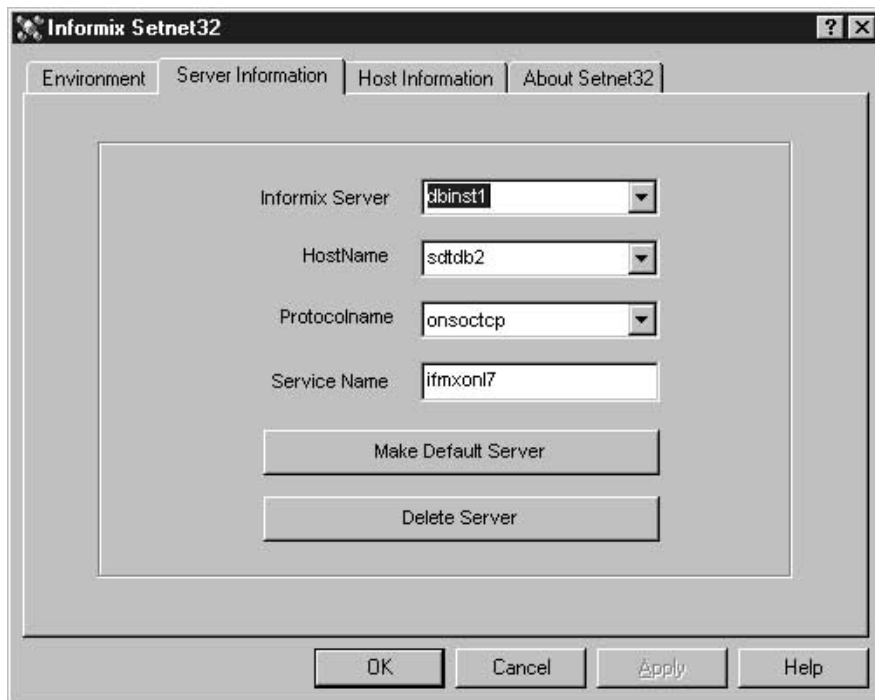
Setting up warehouse sources

- Installing and configuring the ODBC driver

Configuring your Informix client: To set up access to the Informix client, you must configure the Informix server and host information using the Informix-Setnet 32 utility.

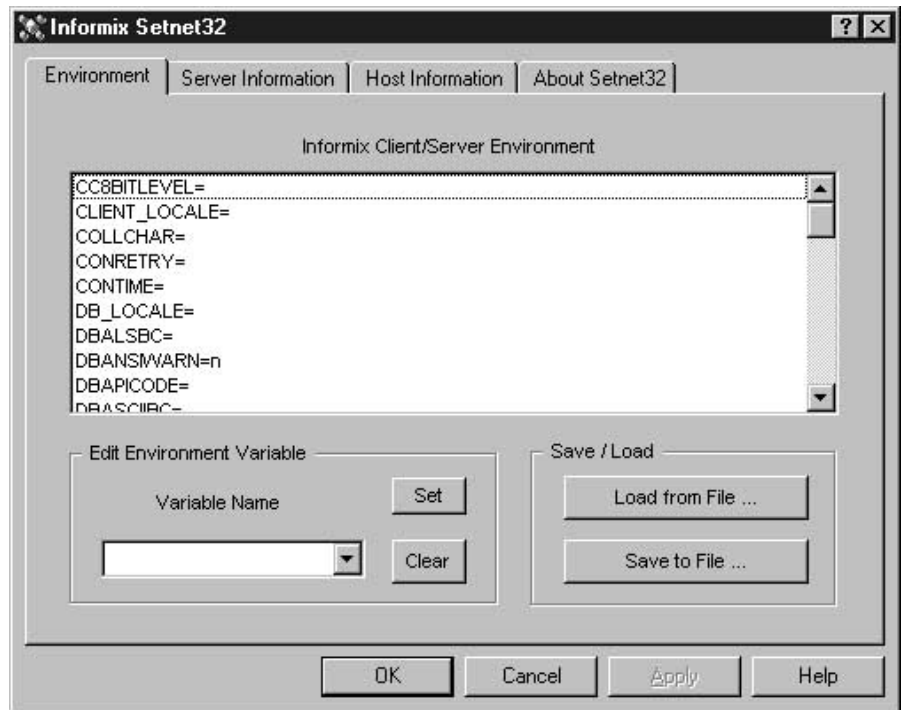
To configure the Informix server information:

1. Click **Start** → **Programs** → **Informix Client** → **Setnet32**.
A Setnet32 window opens.
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 you want to use from the **HostName** drop-down list, or type in 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 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.

- Click the **Environment** tab.



- Type **INFORMIXSERVER** in the **Variable Name** field.
- Click **Set**.
- Click **OK**.

To configure the Informix host information:

- Click the **Host Information** tab in the Setnet32 window.

Setting up warehouse sources



2. Select the name of the host computer that you want to use to establish a network connection from the **Current Host** list, or type a name in the **Current Host** field to define a new host name.
3. Ensure that the user name in the **Current Host** field is the user name for an account on the selected host computer.
4. Type your password in the **Password** field.
5. Click **OK**.

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

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. For information about how to select and install the appropriate driver, see *IBM DB2 Universal Database Quick Beginnings* for your operating system.

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

To register a DSN for an ODBC driver in Windows NT:

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.
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. Select the desired database alias from the **System Data Sources** window.
18. Click **OK**.
19. Close the ODBC windows.

Sybase

This section contains the steps for the following tasks:

- Configuring your Sybase client
- Installing and configuring the ODBC driver

Configuring your Sybase client: To configure a Sybase client:

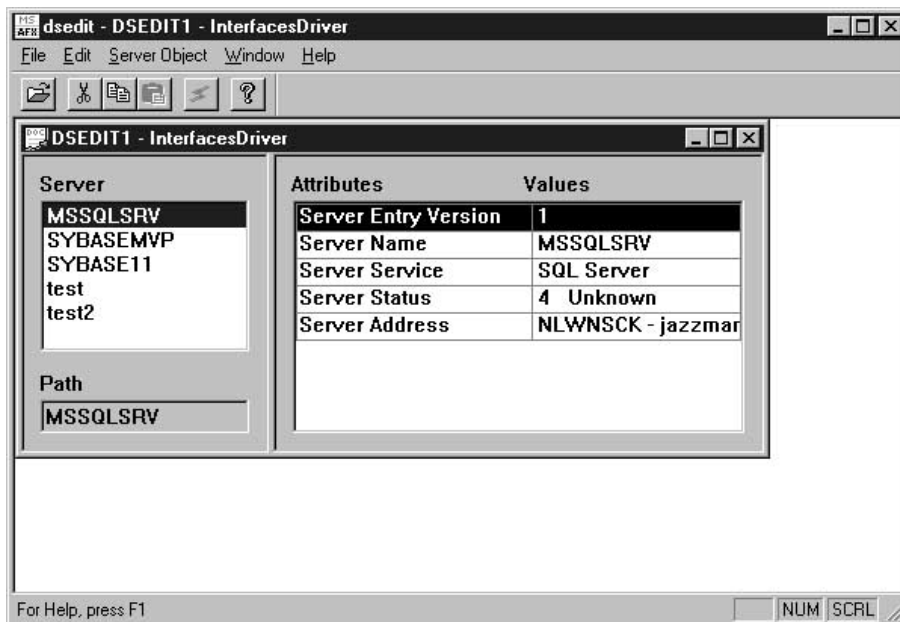
1. Click **Start** → **Programs** → **Sybase for Windows NT** → **DSEDIT**.
The DSEDIT window opens.

Setting up warehouse sources



2. Select the DS name that you want from the DS Name list.
3. Click OK.

The InterfacesDriver window opens.



4. Choose the server that you want to configure from the list in the Server window.
5. If the server that you want to use is not listed, add a new server to the list.

To add a new server to the list of servers:

- a. Right-click in the Server window.
- b. Click **Add**.
- c. Type the name of the server in the **Name** field.

In the right panel of the Server window, the following server attributes will be displayed: Entry version, Server name, Server service, Server status, Server address. You should leave the default values for Entry version, Server service, and Server status.

- d. Select the **Server address** attribute.
- e. Select **Modify Attribute** from the **Server Object** menu.

Optionally, you can double-click on the attribute, or you can right-click on the attribute and select **Modify Attribute**.

- f. Select **NLWNSCK** from the protocol list.

Optionally, if the protocol that you want to use is not in the list:

- 1) Click **Add**.
- 2) Select the protocol type from the **Protocol** list.
- 3) Type the server address and port number in the **Network Address field**.
- 4) Click **OK**.

- g. Verify that the server that you added is highlighted.
- h. Click **OK**.
- i. Verify that the server that you want to configure is highlighted.

6. Close the DSEDIT program.

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

To enable delimited identifier support for Sybase, select the **Enable Quoted Identifiers** check box in the Advanced page of the ODBC Driver Setup notebook.

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. For information about how to select and install the appropriate driver, see *IBM DB2 Universal Database Quick Beginnings* for your operating system.

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.

Setting up warehouse sources

To register a DSN for an ODBC driver in Windows NT:

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 server in the **Server Name** field.
11. Type the name of the database in the **Database Name** field.
12. Click the **Connection** tab.
13. Type the logon ID in the **Default Logon ID** field.
14. Type the ID of the workstation in the **Workstation ID** field.
15. Click **OK**.
16. Select the desired database alias from the System Data Sources window.
17. Click **OK**.
18. Close the ODBC windows.

Oracle

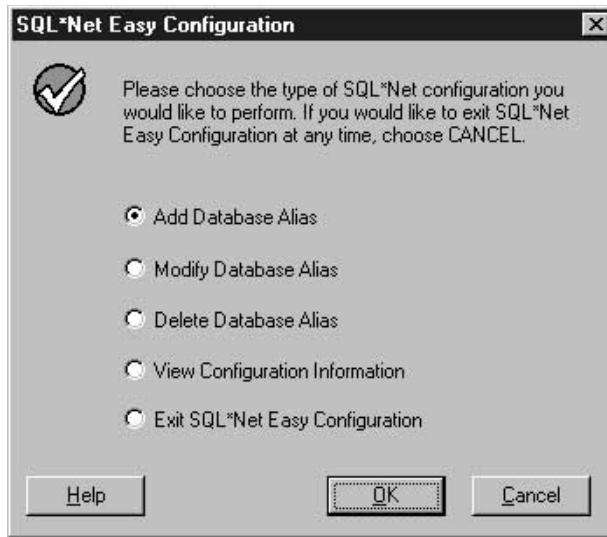
This section contains the steps for the following tasks:

- Configuring your Oracle client
- Installing and configuring the ODBC driver

Configuring your Oracle 7 client: To configure an Oracle 7 client:

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

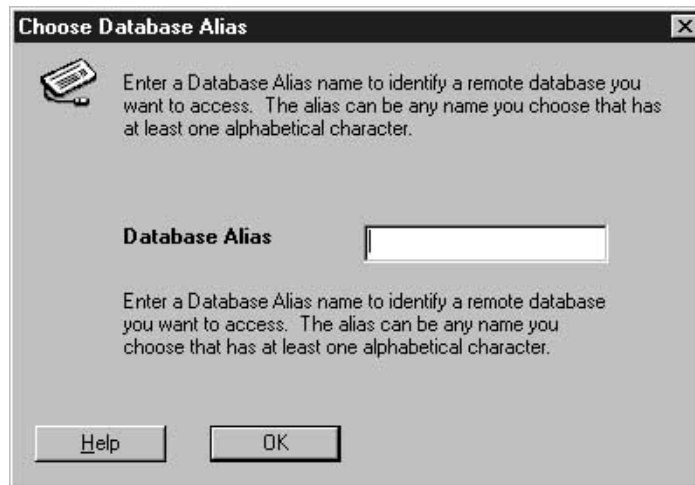
The SQL Net Easy Configuration window opens and displays several choices for working with your client configurations.



2. Click the radio button next to the client configuration choice that you want.

You can add a new client configuration or change or view an existing configuration.

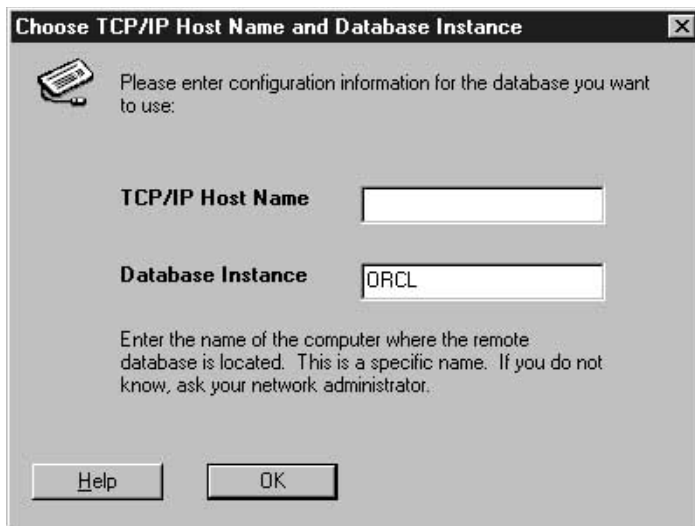
3. If you click **Add database alias**:
 - a.



Click **OK**.

- b. Type the database alias name in the **Database alias** field.
4. Click **OK**.

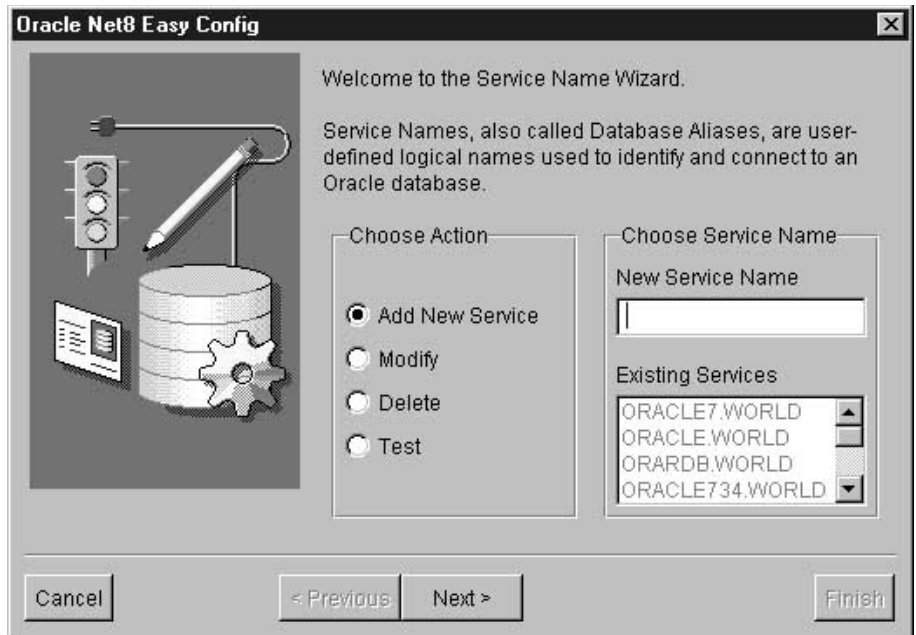
Setting up warehouse sources



5. Type the TCP/IP host name in the **TCP/IP host name** field.
6. Type the database instance in the **Database instance** field, or use the default.
7. Click **OK**.
A confirmation window opens.
8. Review the information in the confirmation window.
9. If the information is correct, click **Yes**.
If the information in the confirmation window is not correct:
 - a. Click **Back** until you reach the window with the incorrect information.
 - b. Correct the information.
 - c. Repeat steps 7 through 9.

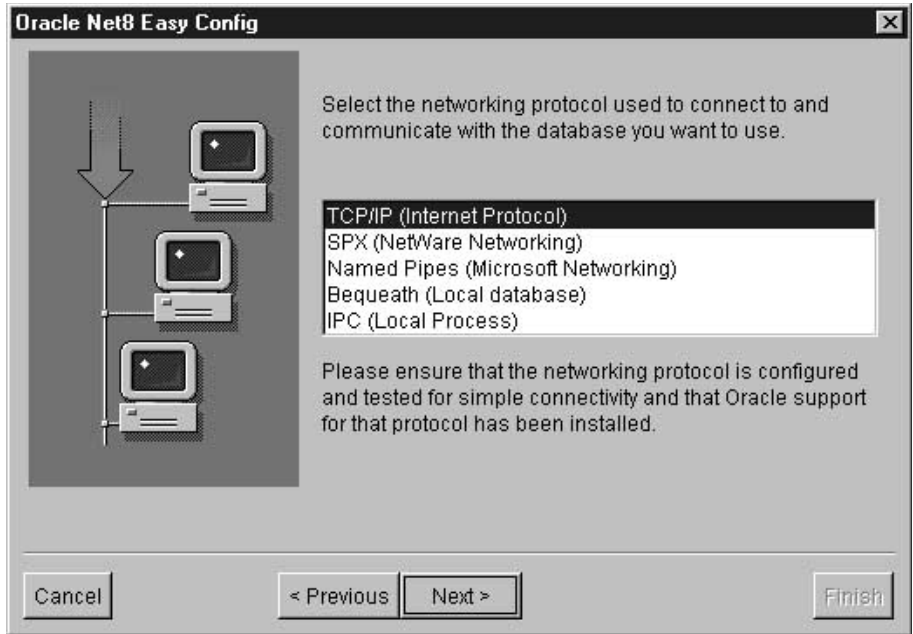
Configuring your Oracle 8 client: To configure an Oracle 8 client:

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.

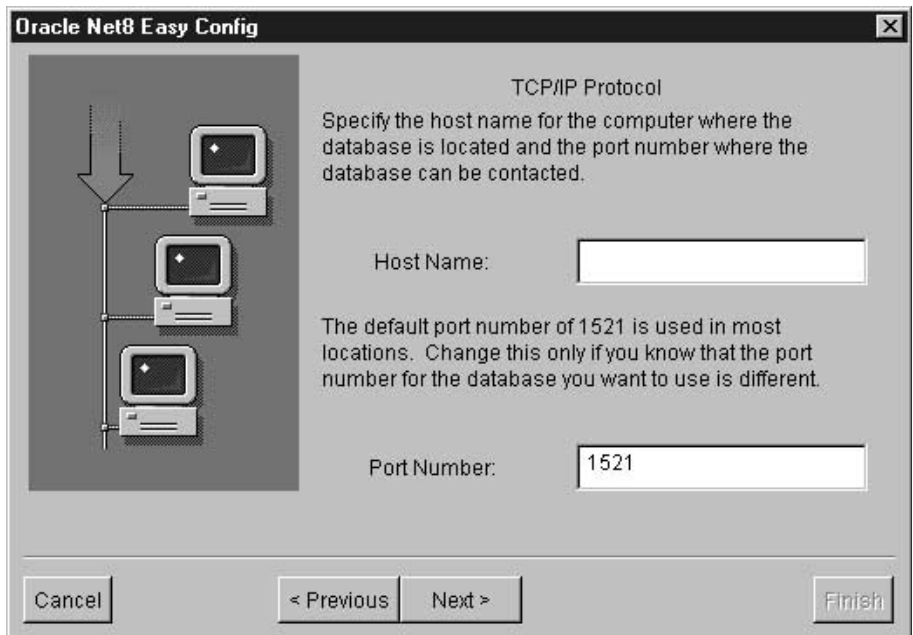


2. Click the radio button next to the client configuration choice you want. You can add a new client configuration or change or view an existing configuration.
3. If you click **Add**, you must also type a database alias name in the **New Service Name** field.
4. Click **Next**.
5. Select the type of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

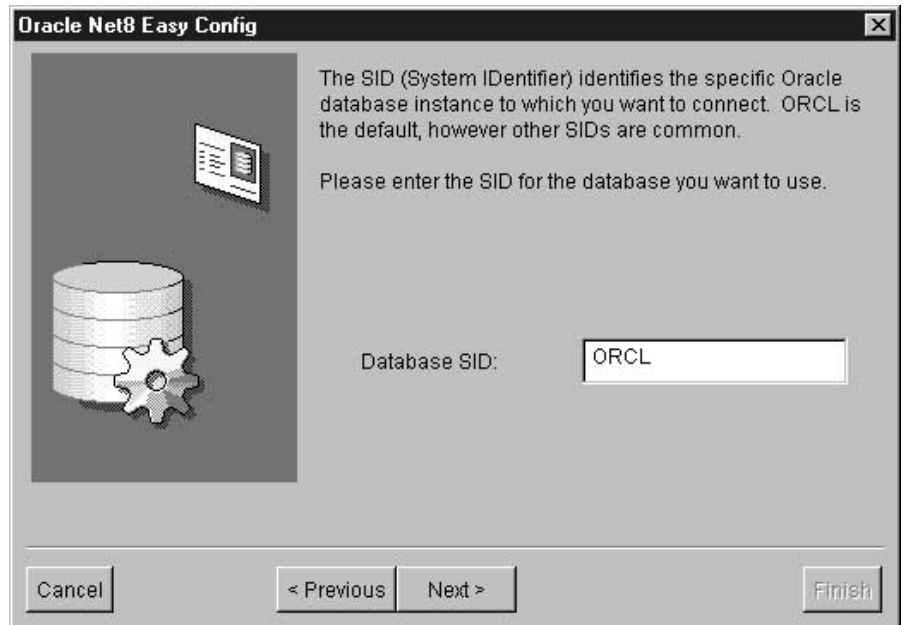
Setting up warehouse sources



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



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



11. Click **Next**.
The Test window opens.
12. Click **Test** to test the configuration.
If you do not want to test the configuration, click **Next**.
If you click **Test**:
 - a. Type the user ID in the **User ID** field.
 - b. Type the password in the **Password** field.
 - c. Click **Test**.
13. Click **Next**.
14. Click **Done**.
A confirmation window opens.
15. Review the information in the confirmation window.
16. 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.

Setting up warehouse sources

- c. Repeat steps 9 through 16.

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

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. For information about how to select and install the appropriate driver, see *IBM DB2 Universal Database Quick Beginnings* for your operating system.

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

To register a DSN for an ODBC driver in Windows NT:

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.

Microsoft SQL Server

This section contains the steps for the following tasks:

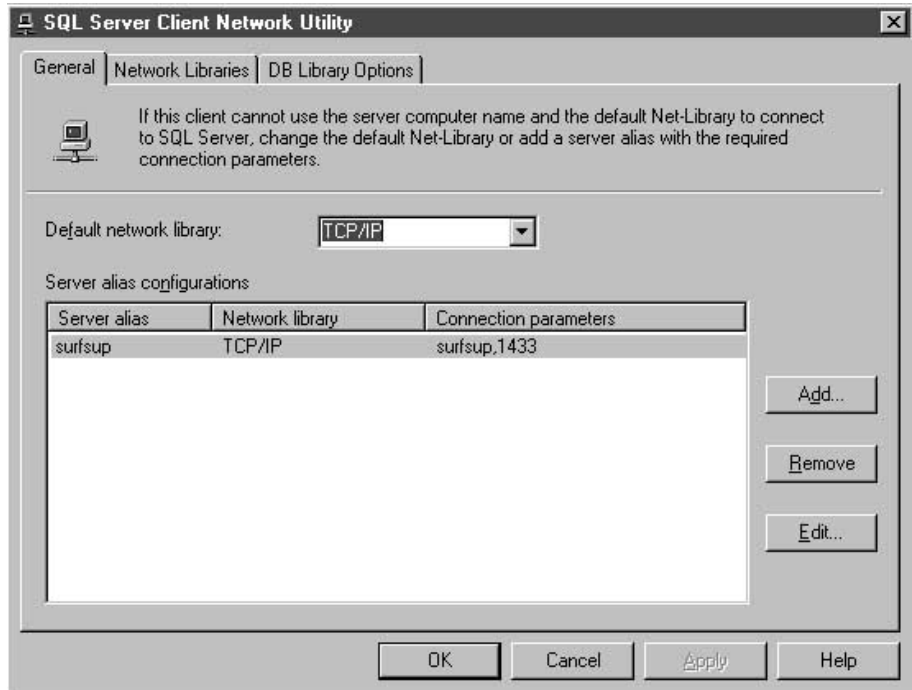
- Configuring your Microsoft SQL Server client
- Installing and configuring the ODBC driver

Configuring your Microsoft SQL Server client: 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.

To configure a Microsoft SQL Server client:

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

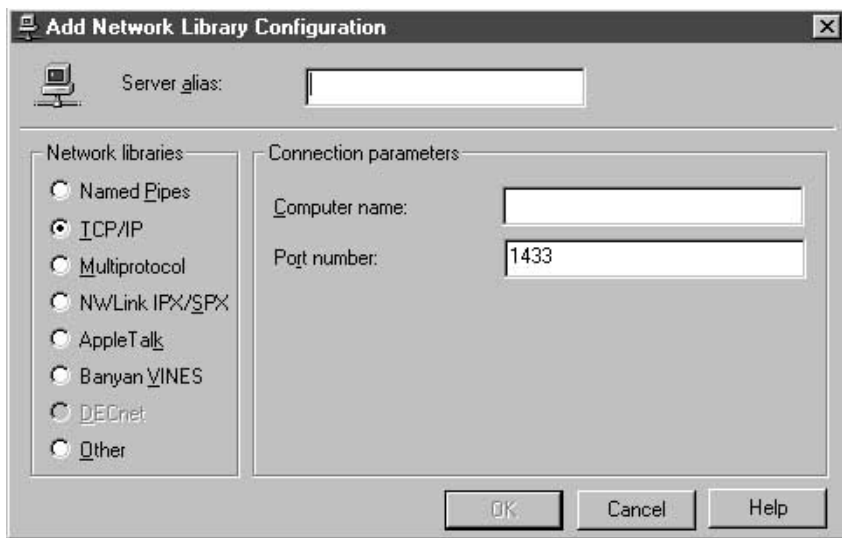
The SQL Server Client Network Utility window opens.



2. Click **TCP/IP** from the **Default network library** list.
3. Select the server name that you want from the **Server alias configurations** list.

Optionally, click **Add** to add a new server to the list. The Add Network Library Configuration window opens.

Setting up warehouse sources



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

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

If you use the Merant ODBC drivers to access Microsoft SQL Server, verify that the system path contains the sqllib\odbc32 directory.

To enable delimited identifier support for Microsoft SQL Server, select the **Enable Quoted Identifiers** check box in the Advanced page of the ODBC Driver Setup notebook.

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. For information about how to select and install the appropriate driver, see *IBM DB2 Universal Database Quick Beginnings* for your operating system.

Setting up warehouse sources

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

Setting up warehouse sources

To register a DSN for an ODBC driver in Windows NT:

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**.
The ODBC SQL Server Driver Setup window opens.
7. Click the **General** tab.
8. Type the database alias name in the **Data Source Name** field.
9. Type a description of the database in the **Description** field.
10. Type the name of the server that you want to use in the **Server Name** field.
11. Type the name of the database that you want to use in the **Database Name** field.
12. Click the **Advanced** tab.
13. Type the user name in the **Default Logon** field.
14. Type the workstation ID in the **Workstation ID** field.
15. Click **OK**.
16. Select the desired database alias from the System Data Sources window.
17. Click **OK**.
18. Close the ODBC windows.

Microsoft Access

This section contains the steps for the following tasks:

- Creating and cataloging a Microsoft Access database
- Creating and cataloging a target warehouse database
- Defining a warehouse that uses the databases

Creating and cataloging a Microsoft Access database: 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.

To catalog the database in ODBC:

1. Click **Start** → **Settings** → **Control Panel**.
2. Double-click **ODBC**.
3. Click **System DSN**.
If you do not have a **System DSN** push button, see the *DB2 Universal Database Troubleshooting Guide*.
4. Click **Add**.
5. Select **Microsoft Access Driver** from the **Installed ODBC Drivers** list.
6. Click **OK**.
7. Type the database alias in the **Data Source Name** field.
8. Optional: Type a description of the database in the **Description** field.
9. Click **Select**.
10. Select the path and file name of the database from the list boxes.
11. Click **OK**.
12. Click **Advanced**.
13. Type a user ID in the **Login Name** field.
14. Type a password in the **Password** field.
15. In the **Options** list, verify that the value of **FIL** is **MS Access**.
16. Click **OK** in the Set Advanced Options window.
17. Click **OK** in the ODBC Microsoft Access Setup window.
18. Click **Close**.

Creating and cataloging a target warehouse database: To create a target warehouse database in DB2:

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

Setting up warehouse sources

To catalog the target warehouse database in ODBC:

1. Click **Start** → **Settings** → **Control Panel**.
2. Double-click **ODBC**.
3. Click **System DSN**.
If you do not have a **System DSN** push button, see *DB2 Universal Database Troubleshooting Guide*.
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**.

Defining a warehouse that uses the database: To create the Data Warehouse Center definitions for the database that you created:

1. Create a warehouse source for the Microsoft Access database by following the instructions in “Defining a non-DB2 database warehouse source in the Data Warehouse Center” on page 97. Specify the following values for the listed parameters:
 - On the Database page:
 - Select the **Customize ODBC Connect String** check box.
 - Type the following string in the **ODBC Connect String** field:
`DSN=database-alias;UID=userID;PWD=password;`
database-alias
The name under which you registered the Microsoft Access database in ODBC.
userID The user ID that you specified when you registered the Microsoft Access database in ODBC.
password
The password that you specified when you registered the Microsoft Access database in ODBC.
 - On the Agent Sites page, specify the agent site on which you registered the Microsoft Access source database and the DB2 warehouse database.
 - On the Tables page:
 - a. Expand the **Tables** folder to import table definitions from your Microsoft Access database.

Setting up warehouse sources

- b. Click **OK** on the Filter window to list all the tables in your database. 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.

Note:

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 OS/390 and OS/400. The CLOB data type is supported for OS/390 beginning with DB2 Version 6. The CLOB data type is supported for OS/400 beginning with Version 4, Release 4 with DB FixPak 4 or later (PTF SF99104). For OS/400, the install disk Version 4, Release 4 dated February 1999 also contains support for the CLOB data type.

2. Create a warehouse for the DB2 database by following the instructions in “Chapter 4. Setting up access to a warehouse” on page 105.
3. Create a step by following the instructions in “Adding steps to the process” on page 128. Give the step the following attributes:
 - It uses one or more source tables from the warehouse source for the Microsoft Access database.
 - It creates a target table in the DB2 warehouse database.
4. Promote the step to test mode.
5. Run the step by right-clicking the step and clicking **Test**.
6. 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.database-name
```

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

Setting up warehouse sources

database-name

The name of the warehouse database.

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

Microsoft Excel

This section contains the steps for the following tasks:

- Creating and cataloging a Microsoft Excel database
- Creating and cataloging a target warehouse database
- Defining a warehouse that uses the databases

Creating and cataloging a Microsoft Excel database: To create a Microsoft Excel database:

1. Open Microsoft Excel.
2. Click **File** → **New**.

The New window opens and displays spreadsheet templates.

3. Select a template.
4. Click **OK**.
5. Click **File** → **Save** to save the spreadsheet.
6. Type the name of the spreadsheet in the **File name** field.
7. Click **Save**.

Note the path and file name of the spreadsheet, because you will use it later.

8. Enter data into the spreadsheet.

To catalog the database in ODBC:

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

If you do not have a **System DSN** push button, see *DB2 Universal Database Troubleshooting Guide*.

4. Click **Add**.
5. Select **Microsoft Excel 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. Select **Excel 97** from the **Version** list.
10. Click **Select Workbook**.
11. Select the path and file name of the database from the list boxes.
12. Click **OK**.

13. Click **OK** in the ODBC Microsoft Excel Setup window.
14. Click **Close**.

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. To create a named table for each worksheet:

1. Select the columns and rows that you want to include in the table.
2. Click **Insert** → **Name** → **Define**.
3. Ensure the **Refers to** field of 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 have 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 checking the **Include system tables** check box.

Creating and cataloging a target warehouse database: To create a target warehouse database in DB2:

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

To catalog the target warehouse database in ODBC:

1. Click **Start** → **Settings** → **Control Panel**.
2. Double-click **ODBC**.
3. Click **System DSN**.
If you do not have a **System DSN** push button, see *DB2 Universal Database Troubleshooting Guide*.
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**.

Setting up warehouse sources

10. Select the path and file name of the database from the list boxes.
11. Click **OK**.
12. Click **Close**.

Defining a warehouse that uses the database: To create the Data Warehouse Center definitions for the database that you created:

1. Create a warehouse source for the Microsoft Excel spreadsheet by following the instructions in “Defining a non-DB2 database warehouse source in the Data Warehouse Center” on page 97. Specify the following values for the listed parameters:
 - On the Database page:
 - Select the **Customize ODBC Connect String** check box.
 - Type the following string in the **ODBC Connect String** field:
`DSN=database-alias;UID=userID;PWD=password;`
database-alias
The name under which you registered the Microsoft Excel spreadsheet in ODBC.
 - userID* The user ID you specified when you registered the Microsoft Excel spreadsheet in ODBC.
 - password*
The password you specified when you registered the Microsoft Excel spreadsheet in ODBC.
 - On the Agent Sites page, specify the agent site on which you registered the Microsoft Excel source spreadsheet and the DB2 warehouse database.
 - On the Tables page:
 - a. Expand the **Tables** folder to import the table definition from your Microsoft Excel spreadsheet.
 - b. Click **OK** on the Filter window to list all the tables in your spreadsheet.
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.
2. Create a warehouse target for the DB2 database by following the instructions in “Chapter 4. Setting up access to a warehouse” on page 105.
3. Create a step by following the instructions in “Adding steps to the process” on page 128. 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.
4. Promote the step to test mode.
 5. Run the step by right-clicking the step and clicking **Test**.
 6. 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 should see the data that you entered in the Microsoft Excel database.

IMS and VSAM

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.

For more information about using Classic Connect with the Data Warehouse Center, see “Appendix F. Using Classic Connect with the Data Warehouse Center” on page 403.

Setting up non-DB2 database warehouse sources on AIX or the Solaris Operating Environment

The following sections describe how to set up Informix, Sybase, Oracle, and Microsoft SQL Server sources on AIX or the Solaris Operating Environment.

Informix

This section contains the steps for the following tasks:

- Configuring your Informix client
- Installing and configuring the ODBC driver

Configuring your Informix client: To configure your Informix client on AIX or the Solaris Operating Environment:

1. Open the `sqlhosts` file.
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.

Setting up warehouse sources

3. Enter the database name.
4. Enter the protocol type onsoctcp.
5. Enter the host name.
6. Enter the port name.
The port name must be the same as the port name defined in the /etc/services file in the Windows installation directory on the client computer.

Figure 5 shows an example of the sqlhosts file with the new entry listing.

```
# Informix V5
database1  olsoc tcp    test0    ifmxfrst1
database2  olsoc tcp    test0    ifmxfrst2
```

Figure 5. A completed sqlhosts file entry listing

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

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. For information about how to select and install the appropriate driver, see *IBM DB2 Universal Database Quick Beginnings* for your operating system.

After the ODBC driver is installed, you must set up access to your Informix database.

To set up access to your Informix database:

1. Open the .odbc.ini file.
2. Define your Informix database alias at the top of the file.
3. Add a new entry to the file.
4. Enter the directory location of the ODBC driver.
5. Enter a description of the database.
6. Enter the database name.
7. Enter the server host name.
8. Enter in the login ID.
9. Enter the password for the login ID.
10. Enter the server name.
11. Enter the service name.

12. Save and close the file.

Figure 6 shows an example of a completed entry for an Informix database.

```
[INF72]
Driver=/home/db2_07_01/3.6/odbc/lib/ivinf12.so
Description=Informix7.23
Database=test7
HostName=xxyyy.zzz.ibm.com
LoginID=informix
Password=password
ServerName=ifmx72
Service=ifmxon72
```

Figure 6. A completed .odbc.ini entry for an Informix database

Sybase

This section contains the steps for the following tasks:

- Configuring your Sybase client
- Installing and configuring the ODBC driver

Configuring your Sybase client: To configure your Sybase client on AIX or the Solaris Operating Environment:

1. Open the interfaces file.
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. Enter the database alias you want to use for the CONNECT statement.
4. Enter query.
5. Enter the protocol you want to use.
6. Enter the server host name.
7. Enter the port number.

Figure 7 shows an example of the interfaces file with the new entry listing.

```
Sybase11
    query tcp ether superman 2000
```

Figure 7. A completed interfaces file entry listing

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

Setting up warehouse sources

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. For information about how to select and install the appropriate driver, see *IBM DB2 Universal Database Quick Beginnings* for your operating system.

After the ODBC driver is installed, you must set up access to your Sybase database.

To set up access to your Sybase database:

1. Open the `.odbc.ini` file.
2. Define your Sybase database alias at the top of the file.
3. Add a new entry to the file.
4. Enter the directory location of the ODBC driver.
5. Enter a description of the database.
6. Define the database type as **master**.
7. Enter the server name.
8. Enter in the logon ID.
9. Enter the password for the logon ID.
10. Enter the directory location of the interfaces file.
11. Optional: To enable delimited identifier support, edit the Sybase data source to include the connect attribute `EQI=1`.
12. Save and close the file.

Figure 8 shows an example of a completed entry for a Sybase database.

```
[Sybase10]
Driver=/home/db2_07_01/3.6/odbc/lib/ivsytb1112.so
Description=Sybase 10 ODBC Database
Database=master
ServerName=Sybase10
LogonID=sybase
Password=password
InterfacesFile=/public/sdt_lab/sybase/AIX/System10/interfaces
```

Figure 8. A completed .odbc.ini entry for a Sybase database

Oracle

This section contains the steps for the following tasks:

- Configuring your Oracle client
- Installing and configuring the ODBC driver

Configuring your Oracle client: To configure your Oracle client on AIX or the Solaris Operating Environment:

1. Open the `tnsnames.ora` file.
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. Enter the protocol you want to use.
4. Enter the server host name.
5. Enter the port name.
6. Enter the SID.

Figure 9 shows an example of the `tnsnames.ora` file with the new entry listing.

```
# Oracle 8.1.5
Oracle8i=
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS =
        (PROTOCOL = TCP)
        (HOST = superman)
        (PORT = 2000)
      )
    )
  (CONNECT_DATA =
    (SID=oracle8i)
  )
)
```

Figure 9. A completed `tnsnames.ora` file entry listing

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

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. For information about how to select and install the appropriate driver, see *IBM DB2 Universal Database Quick Beginnings* for your operating system.

After the ODBC driver is installed, you must set up access to your Oracle database.

To set up access to your Oracle database:

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

Setting up warehouse sources

4. Enter the directory location of the ODBC driver.
5. Enter the server name.
6. Enter a description of the database.
7. Save and close the file.

Figure 10 shows an example of a completed entry for an Oracle database.

```
[Oracle_8]
Driver=/home/db2_07_01/3.6/lib/ivor814.so
ServerName=Oracle8
Description=Oracle 8 ODBC Database
```

Figure 10. A completed `.odbc.ini` entry for an Oracle database

Microsoft SQL Server

This section contains the steps for the following tasks:

- Configuring your Microsoft SQL Server client
- Installing and configuring the ODBC driver

Configuring your Microsoft SQL Server client: To configure your Microsoft SQL Server client on an AIX or Solaris Operating Environment, configure TCP/IP on both the AIX or Solaris Operating Environment client and the Windows server where you have Microsoft SQL Server 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. For more information, see the *Merant DataDirect Connect ODBC Reference* for your operating system.

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

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. For information about how to select and install the appropriate driver, see *IBM DB2 Universal Database Quick Beginnings* for your operating system.

After the ODBC driver is installed, you must set up access to your Microsoft SQL Server database.

To set up access to your Microsoft SQL Server database:

1. Open the `.odbc.ini` file.
2. Define your Microsoft SQL Server database alias at the top of the file.
3. Add a new entry to the file.

4. Enter the network address of the server running Microsoft SQL Server.
5. Enter your preference for ANSI-defined behaviors the driver should use.
6. Enter the database name.
7. Enter in the user ID.
8. Enter the password for the user ID.
9. Enter your preference for recognizing quotation marks in SQL statements.
10. Enter the version of TDS that is used with your database.
11. If you are using SQL Server 6.5, enter your preference for creating temporary stored procedures for SQL Prepare.
12. Save and close the file.

For more information on configuring the ODBC driver and on .odbc.ini values and attributes, see the *Merant DataDirect Connect ODBC Reference* for your operating system.

Figure 11 shows a sample entry for a Microsoft SQL Server database.

```
[MSSQL6.5]
Driver=/home/db2_07_01/3.6/odbc/lib/ivmsss14.so
Address=xyyy.zzz.ibm.com
AnsiNPW=yes
Database=test7
UID=MSSQL
PWD=password
QuotedID=no
TDS=4.2
UseProcForPrepare=1
```

Figure 11. A sample .odbc.ini entry for a Microsoft SQL Server database

Setting up non-DB2 database warehouse sources on OS/2

Informix

This section contains the steps for the following tasks:

- Configuring your Informix client
- Installing and configuring the ODBC driver

Configuring your Informix client: To configure your Informix client on OS/2:

1. Double-click the **Informix** icon.
2. Double-click the **Setnet** icon.
The Informix-Net for OS/2 Utility window opens.
3. Type the host name in the **Hostname** field.

Setting up warehouse sources

4. Type the user name in the **Username** field.
5. Type the service name in the **Servicename** field.
6. Select the protocol from the **Protocolname** drop-down list.
7. Select the radio button next to your password choice in the **Password** field.
8. Click **Save**.

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

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. For information about how to select and install the appropriate driver, see *IBM DB2 Universal Database Quick Beginnings* for your operating system.

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

To register a DSN for an ODBC driver on OS/2:

1. Click the **ODBC Administrator** icon.
If you installed the ODBC Administrator with the OS/2 agent instead of updating an existing ODBC Administrator, the icon is in the Data Warehouse Center Agent folder.
2. Select the ODBC driver that you want to register.
3. Click **Add**.
4. Select the ODBC driver that you want to register, and click **OK**.
5. Select the desired database alias, and click **OK**.
6. Close the ODBC windows.

Sybase

This section contains the steps for the following tasks:

- Configuring your Sybase client
- Installing and configuring the ODBC driver

Configuring your Sybase client: To configure your Sybase client on OS/2:

1. Double-click the **Sybase** icon.
2. Double-click **SQL Edit** icon.
The Open window opens.
3. Select the file that you want to open.

4. Click **Open**.
5. Select the server that you want from the list in the window.
The Server window opens.
If the server that you want is not listed:
 - a. Click **Edit** → **Add Service**.
 - b. Type the name of the server that you want to configure in the **Name** field.
 - c. Click **Edit** → **Add Service**.
You must set up both master and query service.
 - d. Click **Master** from the **Service** list.
 - e. Select the network driver that you want from the **Network driver** list.
 - f. Type the name of the server and the port number in the **Connection information** field.
 - g. Click **OK**.
 - h. Click **Edit** → **Add Service**.
 - i. Click **query** from the **Service** list.
 - j. Select the network driver that you want from the **Network driver** drop-down list.
 - k. Type the name of the server and the port number in the **Connection information** field.
 - l. Click **OK**.
6. Click **Exit** → **Exit edit services**.
7. Close the windows and exit the program.

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

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. For information about how to select and install the appropriate driver, see *IBM DB2 Universal Database Quick Beginnings* for your operating system.

Once 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.

To register a DSN for an ODBC driver on OS/2:

1. Click the **ODBC Administrator** icon.

Setting up warehouse sources

If you installed the ODBC Administrator with the OS/2 agent instead of updating an existing ODBC Administrator, the icon is in the Data Warehouse Center Agent folder.

2. Select the ODBC driver that you want to register.
3. Click **Add**.
4. Select the ODBC driver that you want to register, and click **OK**.
5. Select the desired database alias, and click **OK**.
6. Close the ODBC windows.

Oracle

This section contains the steps for the following tasks:

- Configuring your Oracle client
- Installing and configuring the ODBC driver

Configuring your Oracle client: To configure your Oracle client on OS/2, use SQL*Net for OS/2 clients.

To configure Oracle SQL*Net for OS/2:

1. Go to the \oraos2\network\admin directory.
2. Open the TNSNAMES.ORA file.
3. Add a new connect descriptor to the file for your database.
The connect descriptor must include the following information:
 - a. A descriptor alias.
 - b. The protocol adapter type.
 - c. The port number
 - d. The host name.
 - e. The system ID.
4. Type the information for your client into each of the connect descriptor fields.
5. Save and close the file.

Figure 12 shows an example of the tnsnames.ora file with the new connect descriptor entry.

```
testtcp = (DESCRIPTION=
          (ADDRESS=
           (PROTOCOL=tcp) (PORT=1111) (HOST=oracle))
          (CONNECT_DATA=(SID=ORACLE7))
        )
```

Figure 12. A completed tnsnames.ora connect descriptor entry

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

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. For information about how to select and install the appropriate driver, see *IBM DB2 Universal Database Quick Beginnings* for your operating system.

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

To register a DSN for an ODBC driver on OS/2:

1. Click the **ODBC Administrator** icon.
If you installed the ODBC Administrator with the OS/2 agent instead of updating an existing ODBC Administrator, the icon is in the Data Warehouse Center Agent folder.
2. Select the ODBC driver that you want to register.
3. Click **Add**.
4. Select the ODBC driver that you want to register, and click **OK**.
5. Select the desired database alias, and click **OK**.
6. Close the ODBC windows.

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

To define a non-DB2 warehouse source in the Data Warehouse Center, you must complete the following tasks:

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. For information about creating a source table see "Defining a warehouse source table" in the online help.
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.

Setting up warehouse sources

If a 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.

The following procedures apply to Informix, Sybase, Oracle, and Microsoft SQL Server warehouse sources.

Opening the Warehouse Source notebook

1. Click **Start** → **Programs** → **IBM DB2** → **Control Center**.
The Control Center opens.
2. Click the Data Warehouse Center icon in the Control Center tool bar.
The Data Warehouse Center Administrative Client opens.
3. Right-click the **Warehouse Sources** folder.
4. Click **Define**, then click the type of Warehouse source you want to create.
The Warehouse Source notebook opens.

Adding information about your warehouse source

1. On the Warehouse Source page, type a business name for the warehouse source in the **Name** field. Depending on the type of source that you select, subsequent pages of the notebook might not be available.
You will use this name to refer to your warehouse source throughout the Data Warehouse Center.
2. Optional: Type the name of a person to contact about the warehouse source in the **Administrator** field.
3. Optional: Type a short description of the data in the **Description** field.
4. Optional: Type any additional information about the warehouse source in the **Notes** field.

Specifying an agent site

1. Click the **Agent Sites** tab.
The sites that you can select are displayed in the **Available agent sites** list.
2. Select the name of the site, in the **Available agent sites** list, to which you want to give access to your warehouse source.
3. Click **>**.
The agent site moves to the **Selected agent sites** list. The site can now be used to access the warehouse source.
To include all of the items in the **Available agent sites** list, click **>>**.

Specifying database information

1. Click the **Data source** tab.

The screenshot shows a dialog box titled "Define Warehouse Source" with a "New Warehouse Source" subtitle. It has five tabs: "Warehouse Source", "Agent Sites", "Data Source", "Tables and Views", and "Security". The "Data Source" tab is active. The fields are as follows:

- Data source name: COSTS (dropdown menu)
- System name: (empty dropdown menu)
- User ID: db2admin
- Password: masked with asterisks
- Verify password: masked with asterisks
- Customize ODBC connect string
- ODBC connect string: DSN=;UID=;PWD=;

Buttons for "OK", "Cancel", and "Help" are at the bottom right.

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. In the **System Name** field, specify the host name of the workstation on which the database or file that you are defining as a warehouse source exists.
This step is optional and is used only if you are using a database or file that has the same name as another database or file on a different workstation.
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 a specific connect string.
8. If you selected the ODBC Connect String check box, type the ODBC connect string in the **ODBC Connect String** field.

Importing source tables and views

If the data source was created using Relational Connect and the Create Nickname statement, you must define the source table manually.

1. Click the **Tables and Views** tab.
2. Expand either the **Tables** or **Views** folder depending on which type of object you want to import.

To import both tables and views, right-click the white space of the **Available tables and views** list, and click **Refresh**.

The Filter window opens.

Setting up warehouse sources

3. Specify the types of objects that you want to import from the database by using the Filter window options.

The options available on the Filter window depend on whether you want to import a table or a view.

- Select **Retrieve only those tables that can be replicated** to retrieve only tables that are enabled for replication. This option is available if you are importing a table.
- Select **Retrieve only those views that can be replicated** to retrieve only views that are enabled for replication. This option is available if you are importing a view.
- Select **Include system tables** to retrieve system tables from the database. This option is available if you are importing a table.
- In the **Object schema** field, type a search string, including wildcards, to define the characteristics of the schema names that contain the objects to be returned from the database.

For example, XYZ* returns tables and views with schemas that start with these characters.

- In the **Object name** field, type a search string, including wildcards, to define the characteristics of the table and view names to be returned from the database.

For example, XYZ* returns tables and views that start with these characters.

4. Click **OK** to submit the filter criteria that you entered.

The objects that met your filter criteria are returned by the system and listed in the **Available tables and views** list.

5. From the **Available tables and views** lists, select the tables, views, or individual columns that you want to include in your warehouse source.
6. Click >.

The tables, views, and columns that you selected move to the **Selected tables and views** list.

To include all of the items in the **Available tables and views** list, click >>.

Authorizing access for warehouse groups

1. Click the **Security** tab.
2. From the **Available warehouse groups** list, select the warehouse groups for which you want to grant access to the warehouse source.
3. Click >.

The groups that you select move to the **Selected warehouse groups** list.

To include all of the items in the **Available warehouse groups** list, click >>.

4. Click **OK** to save your changes and close the Define Warehouse Source notebook.

Specify the retry count

1. In the main Data Warehouse Center window, click **Warehouse** —> **Properties** to open the Properties notebook for the Data Warehouse Center.
2. Click the **Processing Options** tab.
3. In the **Count** field, specify how many times a data extract will be retried.
4. In the **Hours** and **Minutes** fields, specify the amount of time that you want to elapse until the Data Warehouse Center retries the extract.

Defining warehouse sources for use with DataJoiner

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

You can define Data Warehouse Center steps that take advantage of DataJoiner's benefits. First, you define warehouses that use DataJoiner databases. Then you define steps that write to those warehouses.

The Data Warehouse Center transformers are not supported with a DataJoiner target database.

Before you read this section, read "Chapter 2. Setting up your warehouse" on page 7, "Chapter 3. Setting up warehouse sources" on page 27, "Chapter 4. Setting up access to a warehouse" on page 105, and "Chapter 5. Defining and running processes" on page 125 to learn about the basic components of a warehouse. You should also be familiar with creating server mappings and nicknames in DataJoiner.

For more information about creating a server mapping in DataJoiner, see the *DB2 DataJoiner: Planning, Installation, and Configuration Guide*.

For information about creating tables and nicknames in DataJoiner, see the *DB2 DataJoiner: Administration Guide*.

Define a separate warehouse source definition in the Data Warehouse Center for each DataJoiner source database. Before you define the warehouse sources, you must map each source database to a DataJoiner database through

Setting up warehouse sources

DataJoiner's server mapping. You might also have to create a user mapping that maps the DataJoiner 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 DataJoiner database. You also must create nicknames for each data source table that you want to use with the Data Warehouse Center.

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

```
CREATE SERVER MAPPING FROM oracle1 TO NODE "oranode"  
TYPE Oracle VERSION 7.2 PROTOCOL "sqlnet"
```

```
create user mapping from USER to server oracle1 authid "iwhserve" password "VWPW"
```

```
create nickname iwh.oracle_target for ORACLE1.iwhserve.oratar
```

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

oracle1 The name that identifies the remote database in DataJoiner.

oranode

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

Oracle The type of database.

7.2 The version of the database.

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

The value of DATABASE is null because Oracle allows only one database per node. For some other data sources, you can specify a database. For more information about creating a server mapping in DataJoiner, see the *DB2 DataJoiner: Planning, Installation, and Configuration Guide*.

- The create user mapping command specifies the user ID that DataJoiner 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 will connect 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 DataJoiner, you can substitute the nickname for the name of the remote table. For example, if you specify SELECT * FROM iwh.oracle_target, DataJoiner returns data from the Oracle table iwhserve.oratar.

Setting up warehouse sources

After you create the mapping and nicknames, you can define the warehouse sources. To define the source tables for each warehouse source, import the DataJoiner nicknames as table definitions. In the previous example, you would import `iwh.oracle_target` from DataJoiner.

Chapter 4. Setting up access to a warehouse

Each set of users will access a warehouse. The operating system you choose for the warehouse depends on capacity planning, the operating system that the users work on, and on your network constraints. The DB2 family of databases can support a wide variety of capacity requirements, so select the database that is appropriate for your warehouse.

Warehouse users must have access to the warehouse database. They can use any application that can access DB2 data.

Setting up a DB2 Universal Database warehouse

You can use any DB2 Universal Database database as a warehouse. For more information about connecting to DB2 Universal Database databases through a DB2 Universal Database server or through any DB2 client, see *DB2 Universal Database Installation and Configuration Supplement*.

Defining privileges to DB2 Universal Database warehouses

Before setting up a warehouse, you need to create a database on the target system. Then set up a user ID with the following privileges for the database:

- CREATETAB (database-level privilege)
- BINDADD (database-level privilege)
- CONNECT (database-level privilege)

In addition, explicit SELECT privilege is required for the following system tables:

- SYSIBM.SYSTABLES
- SYSIBM.SYSCOLUMNS
- SYSIBM.SYSDBAUTH
- SYSIBM.SYSTABAUTH
- SYSIBM.SYSINDEXES
- SYSIBM.SYSRELS
- SYSIBM.SYSTABCONST

Establishing connectivity to DB2 Universal Database warehouses

After you define the required privileges, 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 or OS/2 warehouse agent, or the version of the AIX, OS/390, or Solaris Operating Environment warehouse agent that uses ODBC. If you are using the AIX or Solaris Operating Environment 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 AS/400 warehouse

You can use DB2 Connect or Client Access/400 (CA/400) to access a DB2 for AS/400 warehouse.

Using DB2 Connect

To set up access to a DB2 for AS/400 warehouse with DB2 Connect:

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

Prerequisite products

If you want to access data from a DB2 for AS/400 warehouse, you need a gateway site. Configure the site for DRDA by installing DB2 Connect.

For more information about DRDA, see the *Distributed Relational Database Architecture Connectivity Guide*.

For information about DB2 Connect, see the following books:

- *DB2 Connect Personal Edition Quick Beginnings* or *DB2 Connect Enterprise Edition Quick Beginnings*
- *DB2 Connect User's Guide*
- *Installing and Configuring DB2 Clients*

Defining privileges to DB2 for AS/400 warehouses

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

In addition, explicit SELECT privilege is required for the following system tables:

- SYSIBM.SYSTABLES
- SYSIBM.SYSCOLUMNS
- SYSIBM.SYSINDEXES

- SYSIBM.SYSREFCST
- SYSIBM.SYSCST

The user ID also needs the ALLOBJ privilege to create AS/400 collections.

Setting up the DB2 Connect gateway site

After you define the required privileges, 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.

Establishing connectivity to DB2 for AS/400 warehouses

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 as described in the *DB2 Connect User's Guide*. Perform the bind only once from each type of client.

Using CA/400

To set up CA/400 for access to a DB2 for AS/400 database:

1. Install the latest Client Access PTFs on the AS/400.

To see a full list of the latest informational APARs that will tell you which PTFs are needed for your AS/400, visit the following web page:

<http://www-1.ibm.com/servers/eserver/series/clientaccess/>

2. Under **Support Information**, select **Service Packs**.

On the Client Access Service Packs web page, you can find information about the latest workstation service packs and host PTFs.

To check the current level of your operating system:

- a. Type the following command:

```
GO LICPGM
```

- b. Select Option 10.

- c. Press F11.

3. Ensure that the AS/400 QUSER password is set (not expiring). You can set the password to no password, if required.
4. Enter the TCP/IP host name for the AS/400 system and the Windows NT workstation in the WINNT\SYSTEM32\DRIVERS\ETC\HOSTS file. For example:

```
9.123.456.7 demo400
9.123.456.8 mypc
```

5. Install CA/400 on the Windows NT workstation.
6. Verify that all the servers on the AS/400 system are running. Enter the following command from a DOS command prompt on the Windows NT workstation:

```
cwbping hostname ip
```

You will receive a response like this:

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

```
IBM AS/400 Client Access for Windows 95/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 AS/400 system to start the servers:

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

7. Verify that the AS/400 has a defined DB2 database name. You can check this entry by issuing the Work with Relational Database Directory Entries (WRKRDBDIRE) command on the AS/400.

If the AS/400 does not have a DB2 database name, you will receive an SQL0114 error when you try to access the database.

8. Set the DB2CODEPAGE environment variable to 850 in the Windows NT system variables (not user variables).
9. Download the latest CA/400 service pack and the CWBCFG utility from the Web. You can obtain full details of how to download SP45545 and cwbcfg.exe from the following Web page:

```
http://www-1.ibm.com/servers/eserver/series/clientaccess/service.htm
```

10. Apply the service pack and run the CWBCFG utility to enable CA/400 to run as a system task on Windows NT.

If you do not apply service pack SP45545 and run the CWBCFG utility, you will receive the following error message:

"IBM.[Client Access ODBC Driver (32 bit)][DB2/400 SQL] Communications failure. COMM RC=0x3"

CAUSE: according to the Rochester lab. the CA/400 connection was registered under the "current user" (HKEY_CURRENT_USERS) but should be under HKEY_USERS (.Default).

SOLUTION: you need to register the connection in the correct register, using the stand-alone utility cwbcfg.exe, which can be used to configure a session for the HKEY_USERS.Default user. The correct syntax for cwbcfg is:
cwbcfg /host HOSTNAME /s

where HOSTNAME is the AS/400 name in your HOSTS file, (see step 3).

NOTE: the "/s" option is important. After this command you can verify that the NT registry is set up by running regedt32 and browse HKEY_USERS (.Default)

11. Use the CA/400 ODBC driver to register the AS/400 database as a system DSN. Add other libraries as required, for example, DORMERK,QGPL,MYLIBRARY
12. Test the connection to the AS/400 database with Lotus Approach[®] or Microsoft Query.
If you receive CA/400 errors, click **Help** for a detailed explanation of the error.

Setting up a DB2 for OS/390 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 OS/390.

Defining privileges to DB2 for OS/390 warehouses

Before setting up a warehouse, you need to create a database on the target system. Then set up a user ID with the following privileges for the database:

- Create a database and table space to use as the warehouse.
- Give a user ID DBADM privilege for the database.

Establishing connectivity to the DB2 for OS/390 warehouse database

To establish connectivity to the DB2 for OS/390 database, perform the following tasks:

1. Set up access to DB2 for OS/390.
2. Connect to the warehouse database.

Creating the Data Warehouse Center status table

Create the Data Warehouse Center status table by using the following command:

```
CREATE TABLE IWH.BVBESTATUS (BVNAME VARCHAR(80) NOT NULL, RUN_ID INT NOT NULL, UPDATIME CHAR(26) NOT NULL) IN database-name.tablespace-name
```

database-name

The name of the warehouse database

tablespace-name

The name of the table space that contains the database

Defining the warehouse to the Data Warehouse Center

When you define a target table for a DB2 for OS/390 warehouse, you must specify a tablespace in which to create the table. If you do not specify a tablespace, DB2 for OS/390 creates the table in the default DB2 database defined for the given subsystem.

To specify the tablespace:

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 OS/390 database.

Setting up a DB2 EEE 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 UDB Enterprise - Extended Edition (DB2 EEE).

Defining privileges to the DB2 EEE database

Before setting up a warehouse, you need to create a database on the target system. Then set up a user ID with the following privileges for the database:

- Create a database, node group, and table space to use as the warehouse.

- Give a user ID BINDADD, CONNECT, and CREATETAB privileges for the database.

Establishing connectivity to the DB2 EEE database

Set up access to DB2 EEE by following the instructions in “Establishing connectivity to DB2 Universal Database warehouses” on page 105, starting with the explicit SELECT privilege.

Defining the DB2 EEE database to the Data Warehouse Center

After setting up access to the system, perform the following tasks in the Data Warehouse Center to use a DB2 EEE warehouse:

1. Define a warehouse target.
2. Define or generate a target table.
3. Right-click the target table, and click **Properties**.
The Table notebook opens.
4. Click **Show SQL**.
The Show SQL window opens.
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**.
The Table notebook closes.
When you promote the step to test mode, the Data Warehouse Center creates the target table in the DB2 EEE database.

Defining warehouses for use with DataJoiner

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

You can define Data Warehouse Center steps that take advantage of DataJoiner's function. First, you define warehouses that use DataJoiner databases. Then you define steps that write to those warehouses.

Before you read this section, read "Chapter 2. Setting up your warehouse" on page 7, "Chapter 3. Setting up warehouse sources" on page 27, "Chapter 4. Setting up access to a warehouse" on page 105, and "Chapter 5. Defining and running processes" on page 125 to learn about the basic components of a warehouse. You should also be familiar with creating server mappings and nicknames in DataJoiner.

Create a separate logical warehouse definition in the Data Warehouse Center for each DataJoiner target database. For example, define one warehouse for each Oracle database, another warehouse for each Sybase database, and so on.

Specify the same DataJoiner database as the database for each logical warehouse. 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 DataJoiner database is both the source and target database. DataJoiner then optimizes the query for the DataJoiner target databases (such as Oracle and Sybase). You can define steps with sources from more than one database by taking advantage of the DataJoiner 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. You must create a `BVBESTATUS` table in the DataJoiner database or in each remote database.

If you create the `BVBESTATUS` table in the DataJoiner database, you must use two-phase commit to keep the table in sync with the remote warehouse databases.

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 DataJoiner database for each remote database, because the Data Warehouse Center requires that the name of the table be `BVBESTATUS`. One DataJoiner 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 DataJoiner.

For more information about creating tables and nicknames in DataJoiner, see the *DB2 DataJoiner: Administration Guide*.

For more information about creating a server mapping in DataJoiner, see the *DB2 DataJoiner: Planning, Installation, and Configuration Guide*.

Creating target tables with DataJoiner

With DataJoiner Version 2.1.1 or later, the Data Warehouse Center can create tables directly into a remote database, such as Oracle.

To create the target table:

1. Create a step with a target table in a DataJoiner database.
2. In the Table notebook, type the server mapping name in the **Table space** field.

In the example in “Defining warehouse sources for use with DataJoiner” on page 101, the server mapping name is Oracle 1.

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 proceed to step 4.

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 have a version of DataJoiner before to Version 2.1.1, you can either:

- Create the table in a DataJoiner database and move it to a remote database.
- Create a table in a remote database, update a table in a remote database, or both.

If you are using DataJoiner with DB2 Version 7.1 on Windows NT, UNIX, or OS/2 and FixPak 2 or later you might get an error indicating a bind problem. For example, when using a DataJoiner 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, the db2cli.ini file is located in the .../sqllib directory.

Moving the target table

You can create and test a step in a DataJoiner database, and then move it to a remote database:

1. Create a step with a target table in a DataJoiner 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. (You can also use a modeling or data dictionary tool.) The data types of the DataJoiner tables and the Oracle tables must be compatible.
 - a. Create the table in the Oracle database.
 - b. Drop the DataJoiner table.

For information about how DB2 data types map to Oracle data types, see the *DB2 DataJoiner: Planning, Installation, and Configuration Guide*.

5. Manually create a nickname for the remote table in DataJoiner. 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 DataJoiner to the target correctly.
7. Promote the step to production mode.

Creating or 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.

1. Create the warehouse tables or use existing warehouse tables.

2. Create a nickname for the target table in DataJoiner.
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.

Defining a warehouse target

After you define the sources for your warehouse as warehouse sources, you define the warehouse target that will contain the data. The Data Warehouse Center supports the DB2 Universal Database and DB2 for AS/400 databases for your warehouse. You can also change the CREATE statement for the target table to work with DB2 EEE and DB2 for OS/390. For more information, see “Setting up a DB2 EEE warehouse” on page 110 and “Setting up a DB2 for OS/390 warehouse” on page 109.

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.

Table 10 lists the version and release levels of the warehouse targets that the Data Warehouse Center supports.

Table 10. Version and release levels of supported IBM warehouse targets

Target	Version/Release
DB2 Universal Database for Windows NT	6 - 7
DB2 Universal Database Enterprise-Extended Edition	6 - 7
DB2 Universal Database for OS/2	6 - 7
DB2 Universal Database for AS/400	3.1-5.1
DB2 Universal Database for AIX	6 -7
DB2 Universal Database for Solaris Operating Environment	6 -7

Table 10. Version and release levels of supported IBM warehouse targets (continued)

Target	Version/Release
DB2 Universal Database for OS/390	5 - 7
DB2 DataJoiner	2.1.1
DB2 DataJoiner/Oracle	8
DB2 for VM	3.4 - 5.3.4
DB2 for VSE	3.2, 7.1
CA/400	3.1.2

Defining warehouse target properties

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. For information about defining users and security groups, see the online help.

To define a warehouse target:

1. Right-click the **Warehouse Targets** folder.
2. Click **Define** → **DB2 Family**, then select the version of DB2 Universal Database for your operating system.
The Define Warehouse Target notebook opens.
3. In the **Name** field, type the business name for the warehouse target.
4. In the **Administrator** field, type a contact for the warehouse target.
5. In the **Description** field, type a short description of the data.
6. Click the Agent Sites page. The sites that you can select are displayed in the **Available Agent Sites** list.
7. From the **Available agent sites** list, select the name of the site that you want to give access to your warehouse target, and click >. Your agent site is added to the **Selected agent sites** list. The site can now be used to access the warehouse target.
To include all of the items in the **Available agent sites** list, click >>.
8. Click the **Database** tab.
9. In the **Database name** field, type the name of the database. If you are defining a warehouse target on OS/2, type the database name in uppercase letters. For OS/390, this must be the location name of the database.
10. In the **User ID** field, type the user ID that will access the database from the agent site.
11. In the **Password** field, type the password for the user ID.
12. In the **Verify Password** field, type the password again.

13. Optional: In the Target database area, specify whether the Data Warehouse Center transformers should be created in the target database. If you do not want to create Data Warehouse Center transformers in the target database, click **Drop or do not create transformers**.
 - If you drop transformers for this target, processes or steps that are defined in the warehouse target that use transformers will not run. If you decide later that you want transformers, you can create them again.
 - Dropping transformers in this warehouse target will affect other warehouse targets that are associated with the database.
 - To create the transformers and run them as fenced, click **Create transformers and register as fenced**.
 - To create the transformers and run them as unfenced, click **Create transformers and register as unfenced**.
 - When you change the transformers from fenced to unfenced or from unfenced to fenced, the stored procedures are dropped and then created again. This change affects other warehouse targets that are associated with the database.

Note: The option to create Data Warehouse Center transformers in the target database is not available for OS/390 because OS/390 does not support scripts. For information about creating transformers on OS/390, see the *DB2 Warehouse Manager Installation Guide*.

14. Optional: In the Warehouse target area, specify whether to enable the warehouse target for transformers.
 - To specify that the warehouse target will be able to use transformers, click **Enable target for transformers**. If transformers are already enabled for this warehouse target, **Enable for transformers** will already be selected.
 - To specify that the warehouse target will not be able to use transformers, click **Do not enable for transformers**.
15. Click the **Tables and views** tab.
16. If the data source was created using Relational Connect and the Create Nickname statement, skip to step16c on page 118. If the target tables already exist, you can import their definitions from the database:
 - a. Expand the **Tables** or **Views** folder.

On OS/400 Version 4 Release 2 and Version 4 Release 3, you must select the **View** folder to import system tables.

The Filter window opens.

 - Optional: Select **Only tables that can be replicated** to retrieve only those tables that have been enabled for replication.

- Optional: Select **Include system tables** to retrieve system tables from the database.
- Optional: In the **Object schema** field, type a search string, including wildcards, to define the characteristics of the schema names that contain the objects to be returned from the database. For example, XYZ* would return tables with schemas that start with these characters.
- Optional: In the **Object name** field, type a search string, including wildcards, to define the characteristics of the table names to be returned from the database. For example, XYZ* would return tables that start with these characters.

b. Click **OK**.

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 OS/390 Agent. The warehouse server uses the agent site named AIX Agent for the import.

The objects that meet the filter criteria that you entered are displayed in the **Available tables** list.

c. If the target tables do not exist, define the tables:

- 1) Right-click the white space in the **Selected tables** list.
- 2) Click **Define**.
The Define Warehouse Target Table notebook opens.
- 3) In the **Table schema** field, type the user ID under which you created the warehouse database.
- 4) In the **Table name** field, type a fully qualified name for the table that you are creating. The name can contain up to 128 characters and be an ordinary or delimited identifier.

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 does not support target resource tables that use delimited identifiers. A delimited identifier:

- Is enclosed within quotation marks.

- Can include uppercase and lowercase letters, numbers, underscores, and spaces.
- Can contain a double quotation mark, represented by two consecutive quotation marks.

Table names can be case sensitive, or have other restrictions, for some warehouse target types. See the documentation for your warehouse target type for information about naming conventions.

- 5) Optional: In the **Table space** field, specify the table space in which to create the table.
- 6) Optional: In the **Index table space** field, specify the table space in which to create any indexes on the table.
- 7) In the **Table name** field, type the name of the target table.
- 8) In the **Description** field, type the description of the table.
- 9) In the **Business name** field, type the business name (a descriptive name that users will understand) for the table.
- 10) If you want the Data Warehouse Center to create this table when the step that populates this table is run, verify that the **Data Warehouse Center created table** check box is selected.
Use this option when you want the Data Warehouse Center to create the target table, such as when the target table is the result of an SQL step. Clear this check box if you are using a target table that is already defined.
- 11) Optional: Select the **Transient data** check box to specify that the source data is temporary and will be deleted immediately after the target that receives the data is populated.
- 12) If anyone who has access to the database is to have access to the table, verify that the **Grant to Public** check box is selected.
- 13) Optional: In the **Number of editions** field, specify the number of times that you want the data in this table to be replaced when the process is run. Leave the default of 0 if you want the Data Warehouse Center to append data each time the process runs.
- 14) In the **Edition column** list, click the column you want to use as the edition column for the target table. The number of editions must be greater than 0 for this field to be active. The edition column must be of type integer. Select **<System Generated>** from the list if you want the Data Warehouse Center to create an edition column in the table.
 - 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.
- 15) Verify that the **Part of an OLAP schema** check box is selected if the table is a dimension table or fact table that is to be exported to DB2 OLAP Integration Server.
 - 16) Click the **Columns** tab.
 - 17) On the **Columns** page, right-click in the white space of the table and click **Add**, or right-click a row, and click **Insert**. If you click **Add**, a blank row is added below all other rows. If you click **Insert**, a blank row is inserted above the selected row. To remove a column, right-click on a column and click **Remove**.
 - 18) Type the name of the column in the **Column name** field.
Column names can be case sensitive for some data source types. See the documentation for your data source type for information about naming conventions.
 - 19) In the **Data type** field, specify a data type for the column.
 - 20) If you specify a data type that requires a byte count, specify the length in the **Length** field.
For example, you must specify a length for CHAR data types.
 - 21) If you specify the DECIMAL data type, specify the precision in the **Precision** field.
Precision refers to the total number of digits, which can range from 1 to 131.
 - 22) If you specify the DECIMAL data type, specify the scale in the **Scale** field.

Scale refers to the number of digits to the right of the decimal point, which can range from 0 to the precision of the number.

- 23) To ensure that the code page is translated properly, select the **Is Text** check box if the CHAR or VARCHAR field contains text.
- 24) Select the **Nullable** check box if null (nonexistent or unknown) values are allowed in the table.
- 25) Optional: Select **Replication before** to capture the before-image of the column if the source is a replication source.
- 26) Optional: Select **Replication after** if the column is defined for change capture.
- 27) Optional: In the **Description** field, type a description for the column.
- 28) Click **OK**.

The Table notebook closes. The target table is added under the **Tables** folder in the **Selected tables** list.

17. Click the **Security** tab.
18. Select a warehouse group to grant your users in the group the ability to create steps that use this warehouse target.
19. Click **>**.
The warehouse group moves to the **Selected security groups** list.
20. Click **OK** to save your changes and close the Warehouse Target notebook.

Defining a primary key

To define a warehouse primary key:

1. On the Warehouse Primary Key page, select the column from the **Available columns** list that you want to define as a warehouse primary key, and click **>**. The column that you select is moved to the **Warehouse Primary key columns** list. The order of the columns that you specify is significant, because the database manager uses the warehouse primary key for efficient access to table data.
To include all of the columns in the **Available columns list**, click **>>**.
2. To remove a column from the warehouse primary key definition, select the column from the **Warehouse Primary Key columns** list and click **<**.
To remove all of the columns from the warehouse primary key definition, click **<<**.
3. Optional: Type a name for the warehouse primary key constraint for the table in the **Constraint name** field. You must move a column to the **Warehouse Primary Key columns** list in order to edit this field.
The Data Warehouse Center generates a default name for the warehouse primary key constraint.

If you specify a primary key for a table on OS/390, you must define a unique index for the table after it is created.

Defining foreign keys

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 the step is in development mode, the key is created when you create the table by promoting the step to test mode. If the step is in test mode, the Data Warehouse Center alters the table to add the key when you click **OK**.

To define foreign keys:

1. On the Foreign Keys page, right-click in the white space of the table, and click **Define**. The Define Foreign Key window opens.
2. Optional: In the **Constraint name** field, type a name for the foreign key constraint for the table.

If you don't provide a name, the Data Warehouse Center generates a default name for the foreign key constraint.

3. In the **Object schema** field, select the table schema for the parent table.
4. In the **Object name** field, select the name of the parent table.

The **Primary key columns** field shows the primary key columns that are associated with the table that you specified.

5. From the **Available columns** list, select the column that you want to define as a foreign key, and click **>**. The column is moved to the **Foreign key columns** list.

To include all of the columns in the **Available columns** list, click **>>**.

To remove a foreign key definition from the table or view, select the column from the **Foreign key columns** list, and click **<**.

To remove all of the columns from the foreign key definition, click **<<**.

6. Click **OK**. The Define Foreign Key window closes, and the foreign keys that you defined are displayed in the list of foreign keys on the Foreign Keys page.

If you selected a table that exists in the warehouse database (or you already ran the step), you can view the data in the target tables. You view the data from one table at a time. The Data Warehouse Center displays all the columns of the table, and up to a maximum of 200 rows.

To view the data, click **Sample Data**.

The Sample Data window opens and displays a sample of your data.

Creating the Data Warehouse Center status table in the warehouse database

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.

To create the status table:

1. Connect to the target warehouse database.
2. Issue the following CREATE TABLE statement:

```
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 for Windows NT, DB2 for OS/2, DB2 for AS/400, DB2 for AIX, or DB2 on a UNIX-based platform, use the statement as shown.
- If your target warehouse is in DB2 for OS/390, 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 Version 5 Extended Enterprise Edition, 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 used to determine in which partition a particular row of data is stored

Chapter 5. Defining and running 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. (See “Chapter 3. Setting up warehouse sources” on page 27.)

- A target table to which the Data Warehouse Center is to write data.

You can specify that the Data Warehouse Center is to create the table in a warehouse database, according to your specifications in the step, or you can specify that the Data Warehouse Center is to update an existing table.

- How the data is to be transformed:
 - By issuing an SQL statement that specifies the data to extract and how to transform the data to its target format.
For example, the SQL statement can select data from multiple source tables, join the tables, and write the joined data to a target table.
 - By running a warehouse program or transformer.

Defining and running processes

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*.

This chapter describes how to perform these tasks.

Defining a subject area

You must define a subject area before you create a process. Any user can define or edit a subject area.

To define a subject area:

1. From the Data Warehouse Center tree, right-click the **Subject Areas** folder, and click **Define**.

The Define Subject Area notebook opens.

2. In the **Name** field, type the business name of the subject area.

The name can be 80 characters, including spaces.

3. Optional: In the **Administrator** field, type the name of a contact for this item, for example, your database administrator.

The name can be up to 80 characters long and is case sensitive. The first character of the name must be alphanumeric.

4. Optional: In the **Description** field, type a short description of the subject area.

You can type up to 254 characters.

5. Optional: In the **Notes** field, type additional information about the subject area.

You can type up to 32,000 characters.

6. Click **OK** to create the subject area in the Data Warehouse Center tree.

See “Defining a process” to learn how to define processes under this subject area.

Defining a process

You define a process object to identify a sequence of steps. The sequence might be a sequence of transformations of the data, the sequence in which the steps are to start, or both.

To define a process object:

1. From the Data Warehouse Center window, expand the **Subject Areas** tree.
2. Expand the subject area that is to contain the process.

3. Right-click the **Processes** folder, and click **Define**.
The Define Process notebook opens.
4. In the **Name** field, type the name of the process.
The name can be up to 80 characters long and is case-sensitive. The first character of the name must be alphanumeric. You cannot use & as the first character in English.
5. In the **Administrator** field, type the name of the contact for the process object.
6. Optional: In the **Description** field, type the description of the process.
You can type up to 254 characters.
7. Optional: In the **Notes** field, type additional information about the process.
You can type up to 32,000 characters.
8. Click the **Security** tab.
9. In the **Available warehouse groups** list, click the warehouse groups that you want to grant access to the process and click >.
If you want to select all the warehouse groups in the **Available warehouse groups** list, click >>.
The warehouse groups that you select are moved to the **Selected warehouse groups** list.
10. Click **OK**.
The Define Process notebook closes. Your new process is displayed when you expand the **Processes** folder.

Opening the process

You open a process so that you can graphically define the flow of data in the process.

To open a process:

1. Right-click the process.
2. Click **Open**.

Adding sources and targets to a process

To define the flow of data, you need to add each source that the steps transform and the target tables that result from the transformation.

To add a source or target to a process:

Defining and running processes

1. Click the **Add Data** icon:



Click the canvas at the spot that you want to place the table. The Add Data window opens.

2. In the **Available source and target tables** list, expand the **Warehouse Sources** or **Warehouse Targets** tree.

A list of the warehouse sources or warehouse targets defined in the warehouse is displayed.

3. Expand the tree for the warehouse source or warehouse target.
4. Expand the **Tables**, **Views**, or **Files** tree.
5. Select the table, view, or file that you want to add.
6. Click > to add the table, view, or file to the **Selected source and target tables** list.

To add all the tables, views, or files for a warehouse source or warehouse target to the process, select the **Tables**, **Views**, or **Files** folder, and click >>.

7. Click **OK** to add the tables, views, and files that you selected to the process. If you make any changes to a table after it is added to a process, the changes are saved when you save the process.

The tables, views, and files that you selected are displayed on the Process Model window.

Adding steps to the process

You need to add the steps that define how the source data is to be moved and transformed into the target data. There are four main types of steps:

SQL steps

An 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.

Warehouse program steps

Warehouse program steps run predefined programs and utilities. There are several types of warehouse program steps:

- File programs
- OLAP programs
- Visual Warehouse Version 5.2 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 OS/390 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. For information about configuring the DSNUTILS stored procedure, see the *Warehouse Manager Installation Guide*.

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 several types of transformer steps:

- Statistical transformers
- Warehouse transformers

In addition to the transformer steps, there is a 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. For information about setting up warehouse transformers on OS/390 and AS/400 systems, see the *Warehouse Manager Installation Guide*.

Restriction: The Data Warehouse Center transformers are not supported with a DataJoiner target database.

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.

Defining and running processes

- 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.

Before you use the transformers with the iSeries (AS/400) warehouse agent:

1. Create the Java stored procedures in the IWH collection. Check that the IWH collection exists on your iSeries system by entering WRKLIB IWH on an iSeries command line. If the IWH collection does not exist, create the collection IWH on the iSeries SQL Interactive screen by using the CREATE COLLECTION IWH SQL statement.
2. Complete the following steps:
 - a. Set the environment.
 - b. Register the stored procedures.
 - c. Enable the target database for transformers.

For more information about completing these steps, see the *DB2 Warehouse Manager Installation Guide*

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 OS/390 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 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.

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. For more information, see “Extracting data from an SAP R/3 system” on page 206.

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. You cannot change the step when it is in Production mode. For more information, see “Extracting data from a WebSphere Site Analyzer database” on page 208.

i2 steps

Use the i2 steps to schedule updates to i2 business models, distributed datamarts, and OLAP cubes. For more information, see “Appendix I. Creating an i2 step” on page 441.

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. For information about writing your own warehouse program, see “Writing your own program for use with the Data Warehouse Center” on page 315.

Each group of steps (except the SQL group) has a number of step subtypes. In all cases other than the SQL group, you choose a specific step subtype to move or transform data. For example, the ANOVA transformer is a subtype of the Statistical transformer group. In the case of the SQL group, there is only one type of SQL step. You use this step to perform SQL select operations on your warehouse sources and targets.

You can print information about a step (such as subject area, source table names, and target table names) to a text file. To print step information to a file, right-click the step icon in the Process Model window, click **Print—> Print to File**, and specify the name of the file to which you want to print the information.

Step subtypes

Tables Table 11 on page 132 through Table 17 on page 142 list the step subtypes by program group. A *program group* is a logical grouping of related programs. For example, all the supplied warehouse programs that manipulate files are in the File warehouse program group. The program groups for the supplied warehouse programs and transformers correspond to the icons on the left side of the Process Model window.

Table 11 on page 132 lists file warehouse programs.

Defining and running processes

Table 11. File warehouse programs

Name	Description	Agent sites						See...
		Windows NT or 2000	AIX	Solaris Op. Env.	OS/2	AS/400	OS/390	
Copy file using FTP (VWPRCPY)	Copies files on the agent site to and from a remote host.	✓	✓	✓	✓	✓	✓	"Defining values for the Copy File using FTP (VWPRCPY) program" on page 189
Run FTP command file (VWPFTP)	Runs any FTP command file that you specify.	✓	✓	✓	✓	✓	✓	"Defining values for a Run FTP Command File program (VWPFTP)" on page 190
Data export with ODBC to file (VWPEXPT2)	Selects data in a table that is contained in a database registered in ODBC, and writes the data to a delimited file.	✓	✓	✓	✓			"Defining values for the Data export with ODBC to file (VWPEXPT2) warehouse program" on page 173
Submit OS/390 JCL Jobstream (VWPMVS)	Submits a JCL jobstream to an OS/390 system for processing.	✓	✓	✓	✓		✓	"Defining values for a Submit OS/390 JCL jobstream (VWPMVS) program" on page 192

Table 12 on page 133 lists DB2 warehouse programs.

Table 12. DB2 warehouse programs

Name	Description	Agent sites					
		Windows NT or 2000	AIX	Solaris Op. Env.	OS/2	AS/400	OS/390See...
DB2 UDB load	Loads data from a delimited file into a DB2 UDB database, either replacing or appending to existing data in the database.	✓	✓	✓	✓		“Defining values for a DB2 Universal Database load program” on page 174
DB2 for AS/400 load replace (VWPLOADR)	Loads data from a delimited file into a DB2 for AS/400 database, replacing existing data in the database with new data.					✓	“Defining values for a DB2 UDB for AS/400 Data Load Replace (VWPLOADR) program” on page 180
DB2 for AS/400 load insert (VWPLOADI)	Loads data from a delimited file into a DB2 for AS/400 table, appending new data to the existing data in the database.					✓	“Defining values for a DB2 UDB for AS/400 Data Load Insert (VWPLOADI) program” on page 175
DB2 for OS/390 load	Loads records into one or more tables in a table space.						✓ “Defining values for a DB2 for OS/390 Load program” on page 185
DB2 data export (VWPEXPT1)	Exports data from a local DB2 database to a delimited file.	✓	✓	✓	✓		“Defining values for a DB2 UDB export (VWPEXPT1) warehouse program” on page 172

Defining and running processes

Table 12. DB2 warehouse programs (continued)

Name	Description	Agent sites						See...
		Windows NT or 2000	AIX	Solaris Op. Env.	OS/2	AS/400	OS/390	
DB2 runstats (VWPSTATS)	Runs the DB2 RUNSTATS utility on the specified table.	✓	✓	✓	✓		✓	<p>“Defining values for a DB2 UDB RUNSTATS program” on page 301</p> <p>“Defining values for a DB2 OS/390 RUNSTATS program” on page 302</p>
DB2 reorg (VWPREORG)	Runs the DB2 REORG and RUNSTATS utilities on the specified table.	✓	✓	✓	✓		✓	<p>“Defining values for a DB2 Universal Database REORG program” on page 293</p> <p>“Defining values for a DB2 for OS/390 REORG program” on page 294</p>

Table 13 lists OLAP Server™ warehouse programs.

Table 13. DB2 OLAP Server® programs

Name	Description	Agent sites					
		Windows NT or 2000	AIX	Solaris Op. Env.	OS/2	AS/400	OS/390See...
OLAP Server: Free text data load (ESSDATA1)	Loads data from a comma-delimited flat file into a multidimensional DB2 OLAP Server database using free-form data loading.	✓	✓	✓		✓	“Defining values for the OLAP Server: Free text data load (ESSDATA1) warehouse program” on page 284
OLAP Server: Load data from file with load rules (ESSDATA2)	Loads data from a source flat file into a multidimensional DB2 OLAP Server database using load rules.	✓	✓	✓		✓	“Defining values for the OLAP Server: Load data from file with load rules (ESSDATA2) warehouse program” on page 285
OLAP Server: Load data from SQL table with load rules (ESSDATA3)	Loads data from an SQL table into a multidimensional DB2 OLAP Server database using load rules.	✓	✓	✓		✓	“Defining values for an OLAP Server: Load data from SQL table with load rules (ESSDATA3) warehouse program” on page 287
OLAP Server: Load data from a file without using load rules (ESSDATA4)	Loads data from a flat file into a multidimensional OLAP Server database without using load rules.	✓	✓	✓		✓	“Defining values for an OLAP Server: Load data from a file without using load rules (ESSDATA4) warehouse program” on page 288

Defining and running processes

Table 13. DB2 OLAP Server[®] programs (continued)

Name	Description	Agent sites					
		Windows NT or 2000	AIX	Solaris Op. Env.	OS/2	AS/400	OS/390See...
OLAP Server: Update outline from file (ESSOTL1)	Updates a DB2 OLAP Server outline from a source file using load rules.	✓	✓	✓		✓	“Defining values for an OLAP Server: Update outline from file (ESSOTL1) warehouse program” on page 289
OLAP Server: Update outline from SQL table (ESSOTL2)	Updates a DB2 OLAP Server outline from an SQL table using load rules.	✓	✓	✓		✓	“Defining values for an OLAP Server: Update outline from SQL table (ESSOTL2) program” on page 290
OLAP Server: Default calc (ESSCALC1)	Calls the default DB2 OLAP Server calculation script associated with the target database.	✓	✓	✓		✓	“Defining values for the OLAP Server: Default calc (ESSCALC1) warehouse program” on page 282
OLAP Server: Calc with calc rules (ESSCALC2)	Applies the specified calculation script to a DB2 OLAP Server database.	✓	✓	✓		✓	“Defining values for the OLAP Server: Calc with calc rules (ESSCALC2) warehouse program” on page 283

Table 14 lists replication programs.

Table 14. Replication programs

Name	Description	Agent sites						See...
		Windows NT or 2000	AIX	Solaris Op. Env.	OS/2	AS/400	OS/390	
Base aggregate	Creates a target table that contains the aggregated data for a user table appended at specified intervals.	✓	✓	✓	✓	✓	✓	“Defining a user copy, point-in-time, or base aggregate replication step” on page 197
Change aggregate	Creates a target table that contains aggregated data based on changes recorded for a source table.	✓	✓	✓	✓	✓	✓	“Defining a change aggregate replication step” on page 200
Point-in-time	Creates a target table that matches the source table, with a timestamp column added.	✓	✓	✓	✓	✓	✓	“Defining a user copy, point-in-time, or base aggregate replication step” on page 197
Staging table	Creates a consistent-change-data table that can be used as the source for updating data to multiple target tables.	✓	✓	✓	✓	✓	✓	“Defining a staging table replication step” on page 203
User copy	Creates a target table that matches the source table exactly at the time of the copy.	✓	✓	✓	✓	✓	✓	“Defining a user copy, point-in-time, or base aggregate replication step” on page 197

Defining and running processes

Because the command-line interface to some of the DB2 warehouse programs has changed from Visual Warehouse Version 5.2, the DB2 warehouse programs from Visual Warehouse Version 5.2 are supported separately. Table 15 lists the Version 5.2 warehouse programs.

Table 15. Visual Warehouse Version 5.2 warehouse programs

Name	Description	Agent sites					
		Windows NT or 2000	AIX	Solaris Op. Env.	OS/2	AS/400	OS/390See...
DB2 load replace (VWPLOADR)	Loads data from a delimited file into a DB2 UDB database, replacing existing data in the database with new data.	✓	✓	✓	✓		“Defining values for a Visual Warehouse DB2 UDB Data Load Replace (VWPLOADR) program” on page 385
DB2 load insert (VWPLOADI)	Loads data from a delimited file into a DB2 table, appending new data to the existing data in the database.	✓	✓	✓	✓		“Defining values for a Visual Warehouse DB2 UDB Data Load Insert (VWPLOADI) program” on page 382
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.		✓				“Defining values for a Visual Warehouse 5.2 Load flat file into DB2 UDB EEE (VWPLDPR) program (AIX only)” on page 389

Table 15. Visual Warehouse Version 5.2 warehouse programs (continued)

Name	Description	Agent sites					
		Windows NT or 2000	AIX	Solaris Op. Env.	OS/2	AS/400	OS/390See...
DB2 data export (VWPEXPT1)	Exports data from a local DB2 database to a delimited file.	✓	✓	✓	✓		“Defining values for a Visual Warehouse 5.2 DB2 UDB Data Export (VWPEXPT1) program” on page 381
DB2 runstats (VWPSTATS)	Runs the DB2 RUNSTATS utility on the specified table.	✓	✓	✓	✓		“Define values for a Visual Warehouse 5.2 DB2 UDB RUNSTATS (VWPSTATS) program” on page 388
DB2 reorg (VWPREORG)	Runs the DB2 REORG and RUNSTATS utilities on the specified table.	✓	✓	✓	✓		“Defining values for a Visual Warehouse 5.2 DB2 UDB REORG (VWPREORG) program” on page 388

For detailed information about the supplied warehouse programs, see the online help.

Table 16 on page 140 lists the warehouse transformers.

Defining and running processes

Table 16. Warehouse transformers

Name	Description	Agent sites						AS/400 Version 4 Release 5 and later	OS/390See...
		Windows NT or 2000	AIX	Solaris Op. Env.	OS/2				
Clean data	Replaces data values, removes data rows, clips numeric values, performs numeric discretization, and removes white space.	✓	✓	✓	✓	✓	✓	✓	“Cleaning data” on page 220
Generate key table	Generates or modifies a sequence of unique key values in an existing table.	✓	✓	✓	✓	✓	✓	✓	“Generating key columns” on page 227
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.	✓	✓	✓	✓	✓	✓	✓	“Generating period data” on page 229
Invert data	Inverts the rows and columns of a table, making rows become columns and columns become rows.	✓	✓	✓	✓	✓	✓	✓	“Inverting data” on page 233

Table 16. Warehouse transformers (continued)

		Agent sites							
Name	Description	Windows NT or 2000	AIX	Solaris Op. Env.	OS/2	AS/400 Version 4 Release 5 and later	OS/390See...		
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.	✓	✓	✓	✓	✓	✓	✓	"Pivoting data" on page 235

Table 17 on page 142 lists the statistical transformers.

Defining and running processes

Table 17. Statistical transformers

Name	Description	Agent sites						OS/390See...
		Windows NT or 2000	AIX	Solaris Op. Env.	OS/2	AS/400 Version 4 Release 5 and later		
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.	✓	✓	✓	✓	✓	✓	"ANOVA transformer" on page 241
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 Statistics transformer" on page 243

Table 17. Statistical transformers (continued)

		Agent sites						
Name	Description	Windows NT or 2000	AIX	Solaris Op. Env.	OS/2	AS/400 Version 4 Release 5 and later	OS/390See...	
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.	✓	✓	✓	✓	✓	✓	“Calculate Subtotals transformer” on page 245
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.	✓	✓	✓	✓	✓	✓	“Chi-square transformer” on page 248
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.	✓	✓	✓	✓	✓	✓	“Correlation transformer” on page 251

Defining and running processes

Table 17. Statistical transformers (continued)

Name	Description	Agent sites						OS/390See...
		Windows NT or 2000	AIX	Solaris Op. Env.	OS/2	AS/400 Version 4 Release 5 and later		
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.	✓	✓	✓	✓	✓	✓	"Moving Average transformer" on page 254
Regression	Shows the relationships between two different variables and shows how closely the variables are correlated by performing a backward, full-model regression.	✓	✓	✓	✓	✓	✓	"Regression transformer" on page 259

Table 18 on page 145 lists the transformer that is a user-defined function.

Table 18. User-defined function transformer

Name	Description	Agent sites					
		Windows NT or 2000	AIX	Solaris Op. Env.	OS/2	AS/400	OS/390See...
Format date and time	Changes the format in a source table date field.	✓	✓	✓	✓		“Changing the format of a date field” on page 238

The rest of this chapter provides general information about using steps. Subsequent chapters provide more detail about defining and using each step subtype.

Linking a step to sources and targets

Before you define values for your step, you can use data links to connect your step to applicable warehouse sources and targets. In some cases, the Data Warehouse Center can generate a target table for you. You link a step to sources and targets to define the flow of data from the sources, through transformation by a step, to the targets.

To set up a step to work with a data source, use the Process Model window:

1. Click a step icon in the left pane of the Process Model window. To view descriptions of the step icons, select **View** → **Legend**.
2. After you click a step icon in the left pane, select a step subtype (if applicable).
3. Move the cursor into the Process Model window, and click. A step icon is displayed in the Process Model window.

You can link the steps to its sources and targets.

To link a step:

1. Click the **Link** icon:

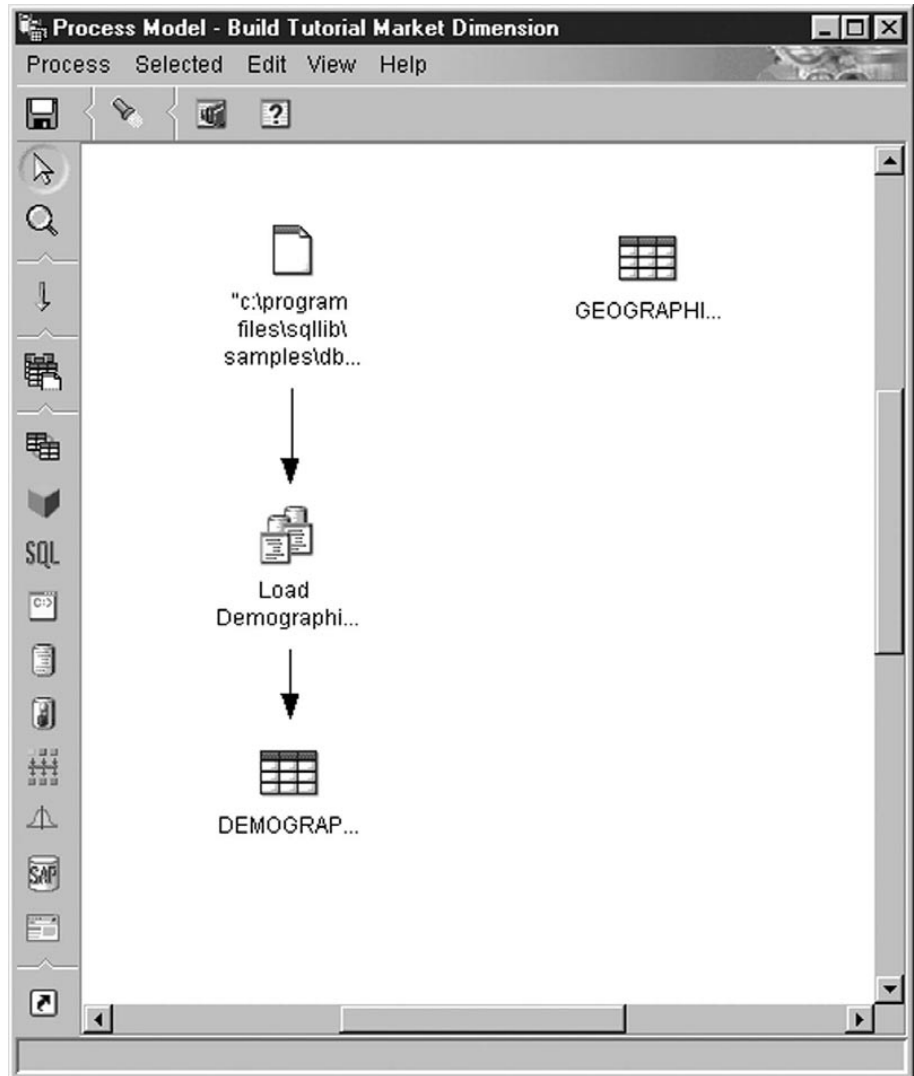


2. Click the **Data Link** icon:



Defining and running processes

3. Click the middle of the source object and drag it to the step.
The Data Warehouse Center draws a line between the source and the step.
The line indicates that the source contains the source data for the step.
4. Click the middle of the step and drag it to the target table.
The line indicates that the target table will contain the target data for the step.



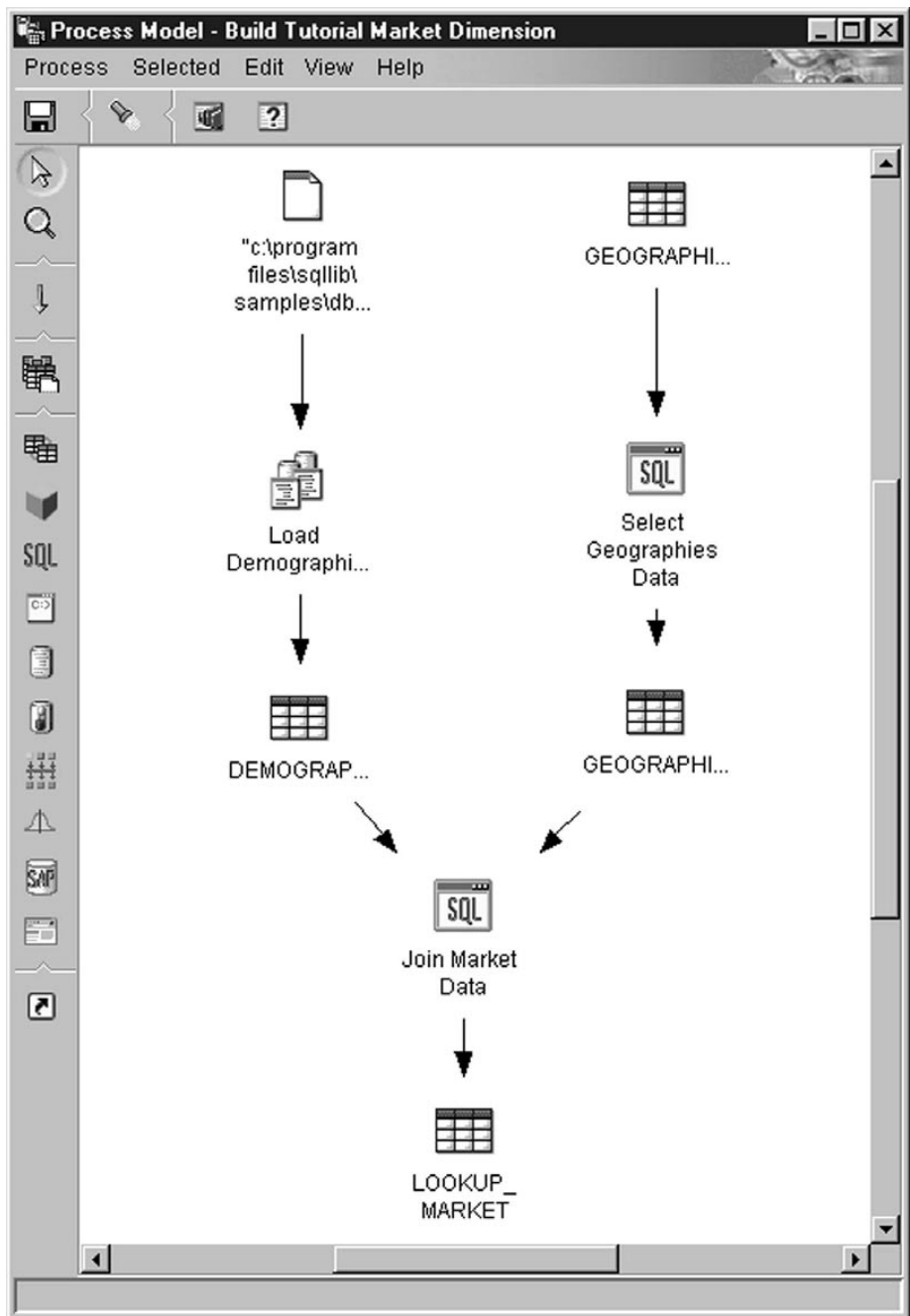
The line indicates that the target table will contain the target data for the step.

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**.

| In the following example, there are two original sources. The
| demographics.txt file contains demographics data about certain cities. The
| GEOGRAPHIES table contains information about which products sold in
| which regions. The Load Demographics Data step loads the demographics
| data into the DEMOGRAPHICS_TARGET target table. The Select Geographies
| Data step selects the GEOGRAPHIES data and writes it to the
| GEOGRAPHIES_TARGET target table. The Join Market Data step joins the
| data in the two target tables, and writes the data to the LOOKUP_MARKET

Defining and running processes

target table. You can use the resulting data to analyze sales by population.



For more information about the previous example, including the procedure for defining the sources, steps, and targets, see the *Business Intelligence Tutorial*.

Defining the basic values of a step subtype

The following sections tell you how to define the values for your step. They also provide basic information about the step, including the types of data sources that the step subtypes work with. For example, some step subtypes work only with warehouse target tables, while others work only with warehouse source or target files.

To define the values for your step, you first must open the step. Right-click the step, and click **Properties**.

Each step subtype notebook consists of four pages:

- The first page consists of general information about the step. This page has the same fields and controls in all step subtypes except the Replication step subtypes.
- The Parameters page consists of values that define the parameters of a step. This page has unique fields and controls for each step subtype.
- The Column Mapping page consists of mapping information between the columns that you selected in the Parameters page and the columns of the target table. In some cases, you can create a target table from the Column Mapping page. Not all steps use the Column Mapping page.
- The Processing Options page consists of values that define how the step is run. Each group of steps have different fields and controls on this page.

Providing general information about a step

The first page of a step subtype notebook is named after the step type. For example, the first page of the ANOVA transformer is called Statistical Transformer. All fields on the first page of a step subtype notebook are the same for all step subtypes.

To define values for the first page of a step subtype notebook:

1. In the **Name** field, type a new name for the step. Or you can keep the name that the Data Warehouse Center automatically supplied for the step.
2. Optional: In the **Administrator** field, type the name of the person who is responsible for the maintenance of this step.
3. Optional: In the **Description** field, type a business description for your step. This description can be a maximum of 254 characters.
4. Optional: In the **Notes** field, type detailed information that might be helpful to users who can access this step.

Defining parameter values

See the step subtype descriptions that follow for information about defining parameter values for a step subtype.

Defining and running processes

Defining column mapping information

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. Figure 13 shows the initial mapping between a source table and a warehouse table.

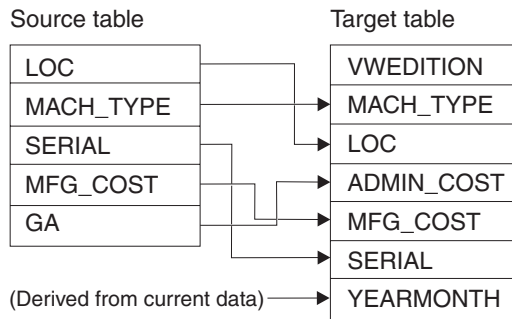


Figure 13. A mapping of source data to a warehouse table

Only certain steps use column mapping. If the column mapping page is blank after you define parameter values for your step, these values result in more than one column, and the Generate Default Table push button is not available, then your step does not use column mapping. Providing column mapping information is optional.

On the Column Mapping page, map the output columns that result from the transformations that you defined on the Parameters 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 create a mapping without using the mouse, use the tab and arrow keys to select the source and target columns you want to map. Then, tab to the **Map** push button and press enter. An arrow will be drawn between the source column and the target column.
- To map all of the unmapped columns by clicking the **Add columns and maps** push button. All of the unmapped source columns are then mapped

to the target columns based on column name, data type, and size attributes. If there is no matching target column, a new target column will be added.

- To delete a mapping, right-click an arrow, and click **Remove**.
- To rename a target column, double-click the column name and type the new name. You can also change any other attributes of the target column by double-clicking the attribute.

Creating a default target table:

1. To create a default target table based on the parameters you specified in the step:
2. Click the Generate Default Table push button. The Generate Default Table window opens.
3. In the Warehouse target list, click the warehouse target where the target table will be defined.
4. Optional: In the Table schema list, type or click the name of the table schema for the target table.
5. Optional: In the Table space list, type or click the name of the table space for the target table.
6. In the Table name field, type a name for the target table.
7. Click OK. The target columns are displayed on the right side of the page.
- 8.

To remove a column from a target table that you defined by clicking the **Generate Default Table** or the **Add columns and maps** push buttons, right-click the column, and click **Remove**. You can do this only if you have not saved your changes to the step notebook after you created the target table.

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.

Defining processing options

This section describes the values that you need to define for the Processing Options fields and controls that are common to all notebooks.

To provide values for processing options:

1. On the Processing Options page, the **Population Type** field displays the method that the step subtype uses to populate data. In some cases, you can choose from more than one population type. The population types are shown in Table 19 on page 152:

Defining and running processes

Table 19. Population types

Population type	Description
Normal	Appends data for a set number of editions, then replaces data one set at time thereafter. For example, your step appends data once a month for 12 months. On the 13th month, the step replaces the data that was written in the first month with the data that results from the run on the 13th month.
Append	Appends data.
Replace	Replaces data.
Program controlled	Population is managed by the program.
Drop	Table is dropped, recreated, and populated.
Replication	Population is managed by replication.

2. In the **Agent Site** list, select an agent site where you want your step to run. The selections in this list are agent sites that are common to the source tables, the target tables, and the transformer or program that you are defining.
3. If you want the option to run your step at any time, select the **Run on demand** check box. If you do not select this box, you cannot run the step from the Work in Progress window.
4. In the **Retry** area, specify how many times you want the step to run again if it needs to be retried, and the amount of time that you want to pass before the next run of the step.

The Data Warehouse Center retries the step when one of the following problems occurs:

- A change is made to a step upon which the step that failed to run is dependent.
- The Data Warehouse Center is unable to send a message to the agent daemon. This can occur when you attempt to run a step that uses an agent that is not running.
- The Data Warehouse Center is unable to send a message to the agent. This can occur when the wrong host name is specified for the agent, or when the connection to the agent is down.
- An error occurs while the Data Warehouse Center is receiving the start acknowledgment from the agent.
- An error occurs while the Data Warehouse Center is receiving a message from the agent.
- There is insufficient page file space for the agent.
- The agent site is overloaded.

The Data Warehouse Center also retries data extracts when one of the following return codes is returned:

- DWC07902
 - DWC07903
 - DWC07904
 - DWC07905
 - DWC07906
 - DWC07907
5. For a transformer step, specify a log table in the **Log table** field.
 6. For a transformer step, specify a trace level in the **Trace level** field.

Running warehouse steps

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.

Developing warehouse steps

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.

Testing warehouse steps

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.

Defining and running processes

Promoting a step to test mode

To promote a step:

1. Right-click the step.
2. Click **Mode** → **Test**.

A confirmation window asks whether you want to save the process. Click **Yes**.

The Data Warehouse Center starts to create the target table, and opens a progress window.

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.
A list of tables is displayed in the Contents pane of the window.
5. Verify that the target table is in the list.

Testing a step

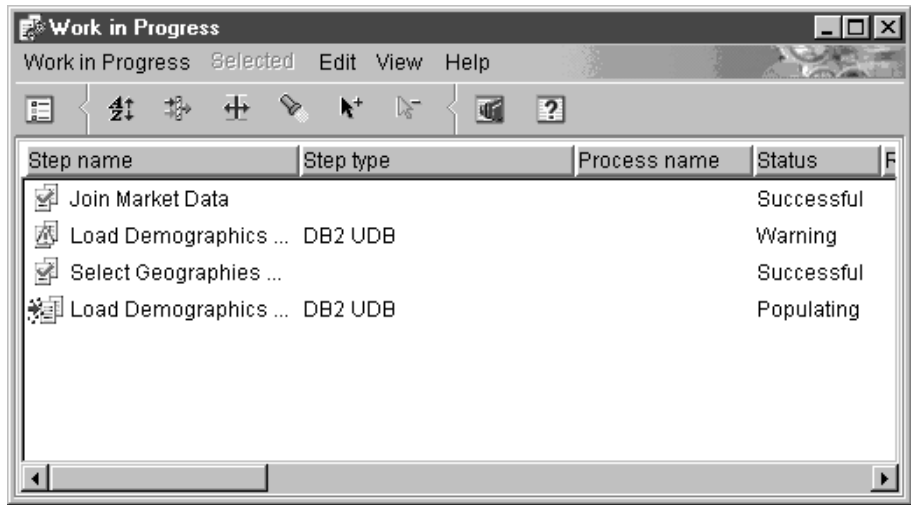
To test a step:

1. From the Process Model, right-click the step.
2. Click **Test**.

The step begins to run. The Data Warehouse Center issues the SQL statements for the step, or starts the warehouse program or transformer. A confirmation window opens after the step stops running.

3. From the main Data Warehouse Center window, click **Data Warehouse Center** → **Work in Progress**.

The Work in Progress window opens.



You use the Work in Progress window to monitor the progress of all steps in the Data Warehouse Center that are running or scheduled. You should see an entry for the step that is running. While the step is running, it has a status of **Populating**. The processing should complete successfully. However, if the processing fails:

For more information about the Work in Progress window, see “Work in Progress—Overview” in the online help.

1. Select the step.
2. Click **Log**.

The Log Viewer window opens.

3. Look for log records with a message type of Run Time Error.
4. Select a record.
5. Click **Details**.

The Log Viewer Details window opens.

If the **Error RC1** field has a value of 8410, then the program failed during processing. Look up the value of the **Error RC2** field, which is the value returned from the program, in the Return Codes section of the online help for the program.

Transformer error messages are different than other messages in the Data Warehouse Center:

- Transformer error messages start with DWC14.
- Transformer error messages, warning messages, and returned SQL codes are stored as secondary codes. Therefore, 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

Defining and running processes

transformer caused the error.

- 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

Recommendation: Periodically clean up the output log tables so that they will not contain obsolete log data.

6. Look in the program log files for more information about the program processing. These files are located in the directory specified by the VWS_LOGGING environment variable. The default value of VWS_LOGGING is `xprogram files\sqllib\logging` on Windows NT and OS/2, and `/var/IWH` on UNIX and OS/390.

The value of VWS_LOGGING is the default value of the **Trace Log Directory** field in the Configuration notebook. If you change the value of the **Trace Log Directory** field, the Data Warehouse Center writes the log files to the new directory you specified, but the value of VWS_LOGGING does not change.

Look in the `trcppp.log` file first. For some errors, this document indicates additional log files to use for problem determination.

One common problem is caused by running Windows NT warehouse agents as a system process rather than a user process. When the warehouse agent runs as a system process, it is not authorized to connect to network drives or products because the process does not have a user ID. Symptoms of this problem include the warehouse agent being unable to find the warehouse program (**Error RC2 = 128** or **Error RC2 = 1** in the Log Viewer Details window) or being unable to initialize the program.

If the warehouse agent runs as a user process, the warehouse agent has the characteristics of the user, including the ability to access network drives or programs to which the user is authorized.

To avoid these problems, perform the following steps:

- a. Change the warehouse server, warehouse logger, and warehouse agent daemon services to run as user processes by performing the following steps:
 - 1) Double-click the **Services** icon in the Windows NT **Control Panel** folder.
 - 2) Stop the services.
 - 3) Select the service, and click **Startup**.
 - 4) Click **This Account**.

- 5) Click the ... button after the **This Account** field to select a user ID.
The user ID must have administration authority in Windows NT and authorization to any required network drive.
 - 6) Type the password for the user ID twice.
 - 7) Click **OK**.
 - 8) Restart the services.
- b. 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.
 - c. 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.

To verify the results of the step's processing:

1. From the Process Model window, right-click the target table.
2. Click **Sample Contents**.

The Data Warehouse Center displays a subset of the data in the table.

You can also view a sample of the data from the DB2 Control Center. Right-click the target table and click **Sample Contents**.

When you run Sample Contents on a warehouse target table from the process modeler, the warehouse source table, warehouse target table, and step must have a common agent site, or the request will fail. To avoid this problem, select the step on the Programs page in the Properties notebook for the agent site you want to use.

When you run Sample Contents against a target file, the first agent site on the selected list is used. However, database maintenance operations might affect the order of the agent sites listed. Sample contents will fail if the agent site selected does not reside in the same system as the source or target file.

Scheduling warehouse processes

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.

Using the Data Warehouse Center's scheduling functions

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

Defining and running processes

a specified date and time. Then, 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 forth.

Specifying that steps are to run in sequence: If a step uses data that is transformed by another step, you can schedule the step to start after the other step finishes processing.

To specify that steps are to run in sequence:

1. From the Process Model window, click the **Task Flow** icon:



2. Click one of the following icons:

On Success

Indicates that a step is to be started only if the step before it ran successfully.

On Completion

Indicates that a step is to be started when the step before it runs successfully or fails.

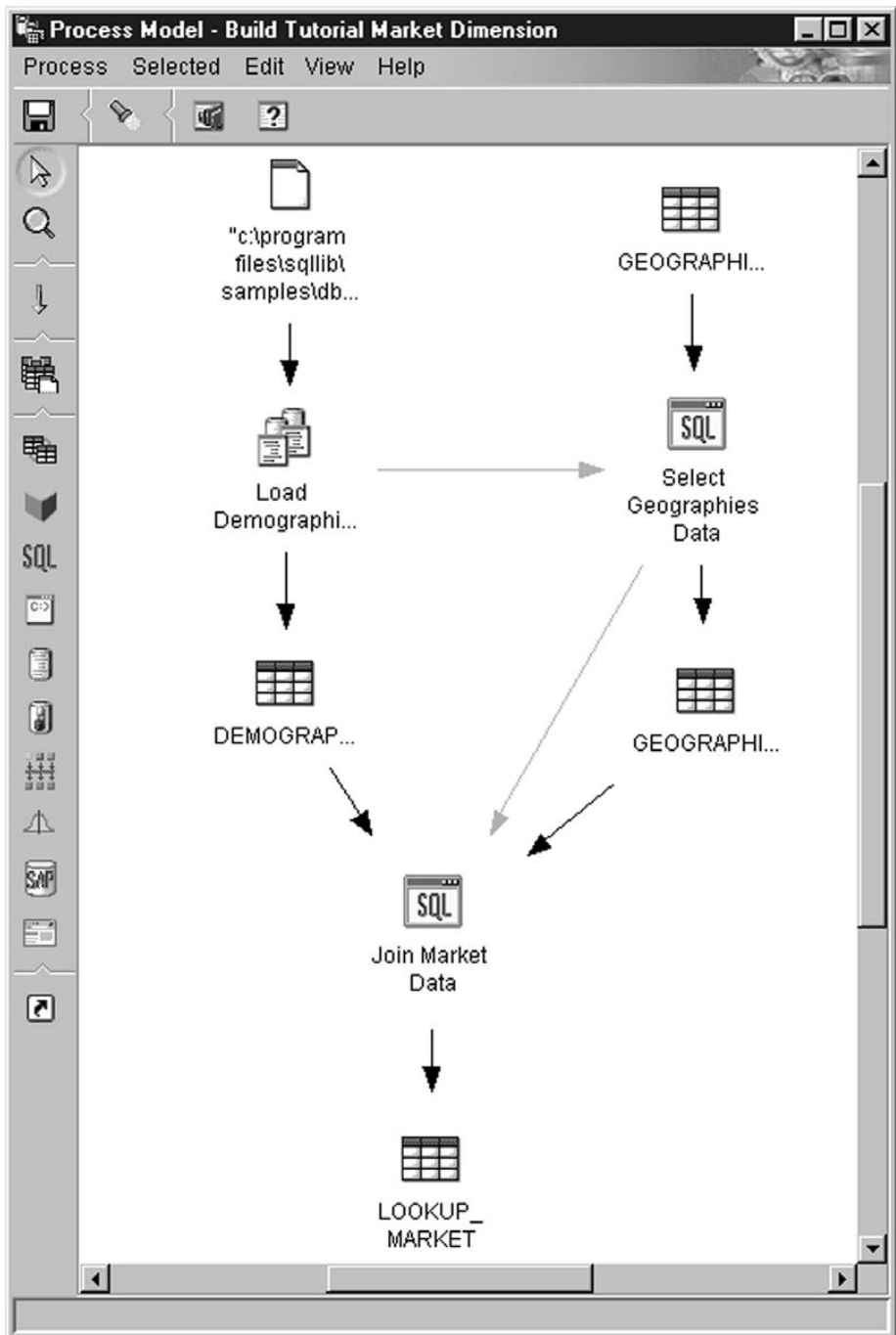
On Failure

Indicates that a step is to be started only if the step before it failed.

For more information, see “Scheduling a step” in the online help.

3. Click the step that is to run first.
4. Hold down the mouse button and drag the mouse to the step that is to run after the first step.
5. Release the mouse button.
An arrow that represents the task flow is displayed on the canvas between the two steps.
6. Repeat steps 3 to 5 for each step in the processing sequence.
The steps will now run in the order that you specified.

For example, in the following figure, the Load Demographics Data step runs first. When it finishes running, the Select Geographies data step runs. When the Select Geographies Data step finishes running, the Join Market Data step runs.



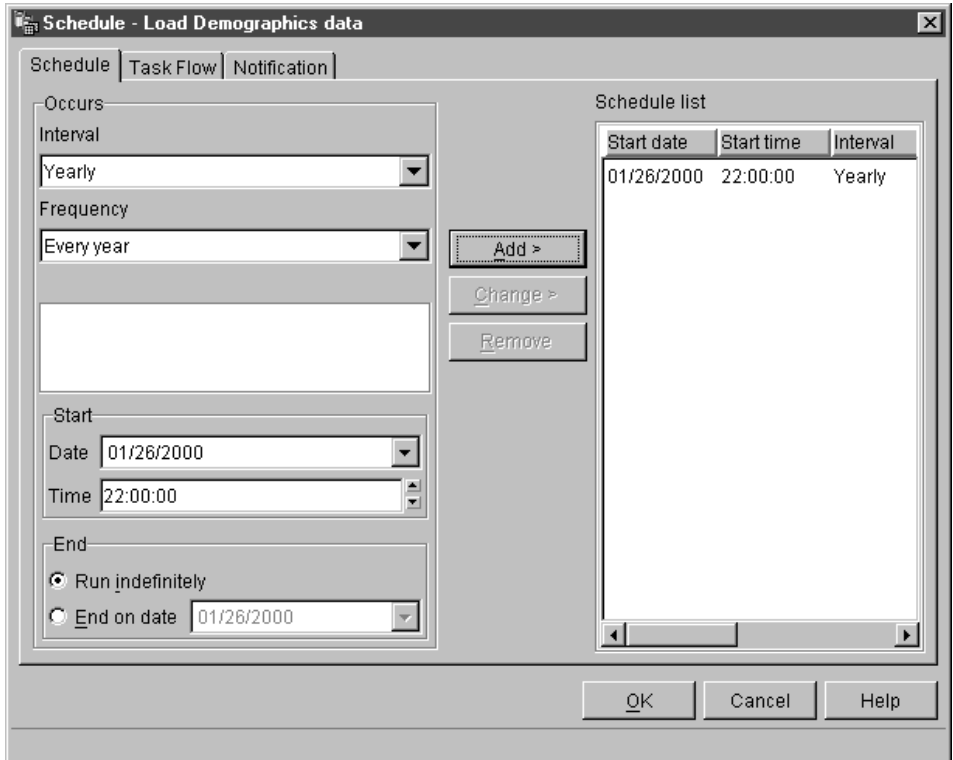
Defining and running processes

Scheduling a step to start at a specified date and time: You can schedule a step to start at a specified date and time. When you schedule a step, you can specify one or more dates and times on which the step is to run. You can also specify that the step is to run once or at a specified interval, such as every Saturday.

To schedule a step to start at a specified date and time:

1. Right-click the step, and click **Schedule**.
The Schedule notebook opens.
2. Click the **Schedule** tab.
3. From the **Interval** list, click the interval at which you want to run the step.
The default selection is **Weekly**.
4. From the **Frequency** list, click the frequency and day.
The default selection is **Every Friday**.
5. From the **Start date and time** field, select the first date and time at which the step is to run. After the first time, the step will run at the interval and frequency that you specify.
The default selection is the current date, at 10:00 in the evening.
6. In the **End** field, specify when to end the schedule: indefinitely or on a specific date.
The default selection is that the schedule is to run indefinitely.
7. Click **Add**.
The schedule is added to the **Schedule list**.
8. Click **OK**.
The specified schedule is created.

In the following example, the Load Demographics Data step is to run once a year, starting on January 26, 2000. The schedule is to run indefinitely.



Using the Data Warehouse Center with other products' scheduling functions

When you define a step, you specify how it is to be populated. You can define a step as *externally populated*, which means that the Data Warehouse Center create the target table, but another program will populate it. The other program is scheduled and runs outside of the Data Warehouse Center.

For example, you define your step as externally populated. You can then use the scheduling capabilities of DPropR to populate the target table.

Promoting a step to 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 step's parameters.

Defining and running processes

To promote a step to production mode:

1. Right-click the step.
2. Click **Mode > Production**.

The Data Warehouse Center opens a progress window.

Starting 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.

You cannot run a process from the external trigger program.

For information about starting the external trigger program from OS/390, see the *Warehouse Manager Installation Guide*.

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.

To use the external trigger program to start a step with a single-byte name, you must have JDK 1.1.7 installed on the warehouse server workstation and the agent site. If you are starting a step with a double-byte name, you must have JDK 1.1.8 instead. You can also use the JDK that is installed with the Data Warehouse Center and the Control Center.

Starting the external trigger server

You must start the external trigger server before you issue commands to the external trigger client.

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

XTServer

►►—java—XTServer—*TriggerServerPort*—————►►

TriggerServerPort

The TCP/IP port assigned to the external trigger server.

This value can be 11001, 11002, 11003, or 11004.

Starting the external trigger client

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

XTClient

►►—java—XTClient—*ServerHostName*—*ServerPort*—*DWCUserID*—*DWCUserPassword*—————►►

►—*StepName*—*Command*—————►►
 └─*WaitForStepCompletion*—┘ └─*RowLimit*—┘

ServerHostName

The TCP/IP host name for the workstation on which the warehouse server is installed.

Specify a fully qualified host name.

ServerPort

The TCP/IP port assigned to the warehouse server. The external trigger client must use the same port as the external trigger server.

This value can be 11001, 11002, 11003, or 11004.

DWCUserID

A user ID with Data Warehouse Center Operations privileges.

DWCUserPassword

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

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.

2 Promote 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 to production mode

The user ID under which you run the external trigger program must be in the same warehouse group as the process that contains the step.

4 Demote 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 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 a 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
```

Status

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. For information about the Data Warehouse Center operations codes, see *DB2 Universal Database Messages and Reason Codes*.

Chapter 6. Moving data

The Data Warehouse Center provides several different ways to move data, depending on your needs:

- To move a small amount of data, you can use SQL steps to select the source data and insert it into a target table.
- To move large amounts of data, you can use the export and load warehouse programs to export data from a table into a file, and then load the data into another table. If the performance of an SQL step is not meeting your needs, try using the export and load warehouse programs.
- To move changes to data rather than moving the whole source, use the replication warehouse programs with a replication source.

In addition to the above methods, the DB2 Control Center has an import and export utility that you can use to move data. For more information, see the *DB2 Universal Database Data Movement Utilities Guide and Reference*.

Selecting and inserting data

You can use an SQL step to select source columns and insert the data from the columns into a target table. You can either specify that the Data Warehouse Center generate 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. The Parameters page of the Step notebook will not be available for this step until you link it to a source in the Process Model window. You can also link this step to a target in the Process Model window. If you don't link the step to a target, you can specify that a table is created when the step runs.

Tip: When creating 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.

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

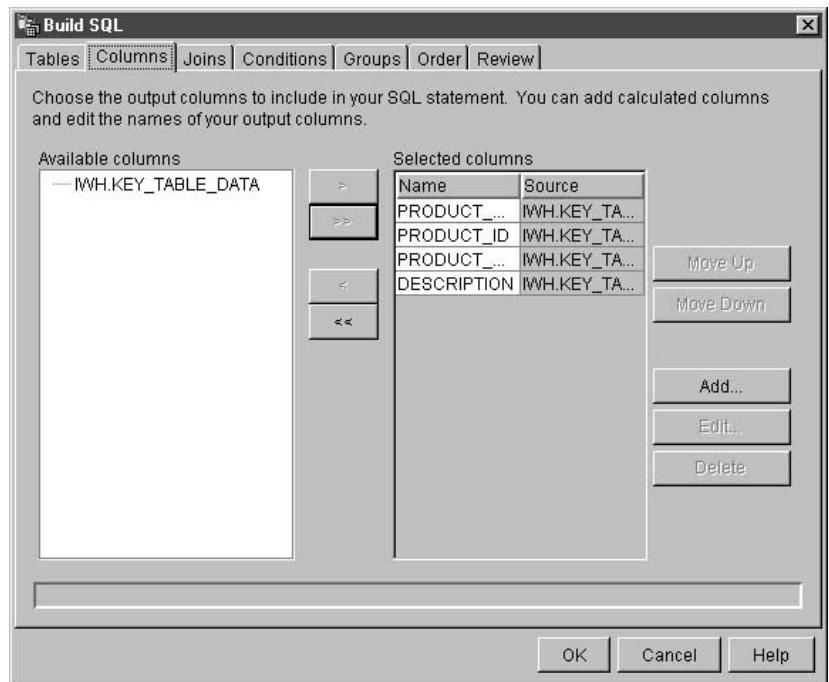
Defining the step

To select and insert data:

1. Open the step notebook.

Moving data

2. Specify general information about the program. For more information, see “Providing general information about a step” on page 149.
3. On the SQL Statement page, create an SQL statement either by using SQL Assist or manually:
 - To use SQL Assist to generate SQL, click **Build SQL**. SQL Assist is launched.
 - a. Click the **Columns** tab.
 - b. From the **Available columns** list, select a source columns that you want to include in your target table and click >.
To include all the columns in the **Available columns** list in your target table, click >>.



- c. Click **Finish**.
SQL Assist closes.
- d. Click the **Column Mapping** tab and view the column mapping that the Data Warehouse Center generated for you, or create a default target table. See “Creating a default target table” on page 151 for information about creating a default target table from the Column Mapping page.

After you close SQL Assist, the newly generated SQL is displayed in the **SQL statement** field.

- To create SQL manually, or to edit SQL that is generated by SQL Assist, click **Edit**. The SQL Statement field is available. SQL Assist is no longer available.

To erase the text in the SQL statement field, click **Clear**. If you want to create new SQL, you must click **Edit** again.

4. Optional: When you finish generating or editing your SQL statement, click **Test** to test the SQL query. The Data Warehouse Center returns sample results of your query.
5. Optional: On the Column Mapping page, view or modify your column mappings. For more information, see “Defining column mapping information” on page 150.
6. On the Processing Options page, in the **Population type** list, specify whether the step appends or replaces data in the target. For more information about population types, see “Defining processing options” on page 151.
7. In the **Agent Site** list, select an agent site where you want your step to run. The selections in this list are agent sites that are common to the source tables and the target table.
8. If you want to have the option to run your step at any time, select the **Run on demand** check box. Your step must be in test or production mode before you can run it.
9. Optional: Select the **Populate externally** check box if the step is populated externally, meaning that it is started in some way other than by the Data Warehouse Center. The step does not require any other means of running in the Data Warehouse Center in order for you to change the mode to production.

If the **Populate externally** check box is not selected, then the step must either have a schedule, be linked to a transient table that is input to another step, or be started by another program in order for you to change the mode to production.
10. In the **Retry** area, specify how many times you want the step to run again if it needs to be retried and the amount of time that you want to pass before the next run of the step. For more information about the Retry area, see “Defining processing options” on page 151.
11. In the **Return Codes** area, select the return code level that you want to receive if no rows are returned, or if an SQL warning occurs when your step is run.
12. In the **Incremental commit** check box and the **Every x rows** field, specify whether you want the step to perform an incremental commit and, if so, how often you want the commit to occur. For more information about incremental commit, see “Incremental commit” on page 170.
13. Click **OK** to save your changes and close the step notebook.

Moving data

Incremental commit

Incremental commit is an option that is available for all SQL steps that allows you to control the commit scope of the data that is managed by the Data Warehouse Center. Incremental commit can be used when the volume of data to be moved by the agent is large enough that the DB2 log files might fill up before the entire work of the step is complete, or when you want to save partial data. SQL steps will complete with an error if the amount of data being moved exceeds the DB2 maximum log files that have been allocated.

The 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 you do not specify incremental commit, and an error is returned, the data will be rolled back to its original state.
- 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.

Loading and exporting data

You can use the supplied export programs, such as DB2 data export, to extract data from a DB2 database and write it to a flat file. You can use the supplied load programs, such as DB2 load replace, to extract data from a file and write it to another DB2 database.

Recommendation: Use these programs 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 programs operate on a delimited 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. The number and order of the input and output fields must match.

These programs 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 and OS/2, and `/var/IWH` on UNIX and OS/390, where *x* is the drive on which you installed the warehouse agent.

The value of the VWS_LOGGING environment variable is the default value of the **Trace Log Directory** field in the Configuration notebook. If you change the value of the **Trace Log Directory** field, the Data Warehouse Center writes the log files to the new directory that you specified, but the value of VWS_LOGGING does not change.

For a complete list of parameters, restrictions, and return codes for these programs, see "Steps and tasks" in the online help.

The sections that follow describe how to define the different step subtypes for the DB2 Universal Database warehouse programs.

The sections about the DB2 UDB export and DB2 UDB load warehouse programs describe how to define the basic values for these programs. Information about defining the advanced features for the DB2 Universal Database Insert and DB2 Universal Database load warehouse programs are described in the online help.

Exporting data

You can use the supplied warehouse programs to export data from a DB2 UDB database or a database that is defined in ODBC.

Moving data

Defining values for a DB2 UDB export (VWPEXPT1) warehouse program

Use the Step Properties notebook for DB2 UDB export to create a step that can be used to export data from a DB2 Universal Database table or view to a file located at the agent site.

The database server does not need to be on the agent site. However, the target file must be on the agent site.

Specify the name of the target file as it is used on the agent site. DB2 UDB export creates the target file if it does not exist, and replaces it if it exists.

Requirement: The source table, or view, must be linked to the step in the Process Model window. The step must be linked to the warehouse target.

DB2 UDB export steps do not use the Column Mapping page.

To define values for a step that runs a DB2 Universal Database export warehouse program:

1. Open the step notebook.
2. Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149.
3. Optional: On the Parameters page, specify information for the export step:
 - In the **Delimiters** group box, change the delimiter types by clicking on the down arrow to display a list of valid delimiters, and select one:
 - The column delimiter specifies the character that is used to delimit each column exported to the target file.
 - The character delimiter specifies the character that is used to enclose character based columns that are exported to the target file.
 - The decimal delimiter specifies the character that is used as a decimal point in number columns that are exported to the target file.
 - Create or update SQL statements in the **SELECT statement** field:
 - Click **Build SQL** to open SQL Assist, which will help you build and generate the SQL statements. The Build SQL window opens.
 - Click **Edit** to edit the SQL statements shown. The SQL statement field becomes available.
 - Click **Test** to test the SQL statements shown. The Sample Contents window opens.
 - Click **Clear** to clear the SQL statements shown. The SQL statements in the SELECT statement field are deleted.
 - To specify advanced options for the export, click **Advanced**. See the online help for this window for information about specifying these options.

4. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
5. Click **OK** to save your changes and close the step notebook.

Defining values for the Data export with ODBC to file (VWPEXPT2) warehouse program

Use the Data export with ODBC to file (VWPEXPT2) warehouse program to select data in a table that is contained in a database registered in ODBC, and write the data to a delimited file. To run this program on AIX or UNIX, use the ODBC version of the warehouse agent.

This step subtype uses a warehouse source or target file as 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 program is available for the following operating systems:

- Windows NT
- AIX
- Solaris Operating Environment
- OS/2

The Column Mapping page is not available for this step.

To define values for a step that runs this warehouse program:

1. Open the step notebook.
2. Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149.
3. On the Parameters page, in the **Column delimiter** field, click or type the character that you want to use as a column delimiter.
4. Create or generate SQL either using SQL Assist or manually:
 - To use SQL Assist to generate a SELECT statement, click **Build SQL**. SQL Assist opens. After you close SQL Assist, the newly generated SQL is displayed in the SELECT statement field.
 - To create SQL manually, or you want to edit SQL that is generated by SQL Assist, click **Edit**. The SELECT Statement field is available. SQL Assist is no longer available.

Optional: If you want to erase the text in the SELECT statement field, click **Clear**. If you want to create new SQL, you must click **Edit** again.

5. Optional: When you finish generating or editing your SQL, click **Test** to test the SQL query. The Data Warehouse Center returns sample results of your query.

Moving data

6. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
7. Click **OK** to save your changes and close the step notebook.

Loading data

You can use the supplied warehouse programs to load data into a DB2 Universal Database, DB2 for AS/400 and DB2 for OS/390 database.

Defining values for a DB2 Universal Database load program

Use the DB2 Universal Database Load Step Properties notebook to create a step that loads data from a source or target file into a DB2 Universal Database table.

You can use a warehouse source or target as a source for this step subtype. Link the source to the step subtype in the Process Model window. Then, link the step subtype to a warehouse target, or specify that the Data Warehouse Center will create the target table.

The Column Mapping page is not available for this step.

To define values for a step that runs a DB2 Universal Database load warehouse program:

1. Open the step notebook.
2. Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149.
3. Optional: On the Parameters page do the following:
 - Select the load mode from the **Load mode** drop-down list. The possible values are:
 - INSERT - loads the table and appends data from the file to the existing table. (This is the default.)
 - REPLACE - loads the table and replaces all the data in the existing table.
 - RESTART - restarts a load process that was abnormally terminated.
 - TERMINATE - terminates a currently executing load process.
 - In the **Delimiters** group box, change the delimiter types by clicking on the down arrow to display a list of valid delimiters, and select one:
 - The column delimiter specifies the character that is used to delimit each column in the file to be loaded.
 - The character delimiter specifies the character that is used to enclose character-based columns in the file to be loaded.
 - The decimal delimiter specifies the character that is used as a decimal point in number columns in the file to be loaded.

- To specify advanced options for the load, click **Advanced**. See the online help for this window for information about specifying these options.
 - To specify that the step is to create a default target table, select the **Create warehouse target table based on parameters** check box and provide information about the target table. See “Creating a default target table” on page 151 for general information about the fields for creating a default target table.
4. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
 5. Click **OK** to save your changes and close the step notebook.

Defining values for a DB2 UDB for AS/400 Data Load Insert (VWPLOADI) program

Use the DB2 UDB for AS/400 Data Load Insert (VWPLOADI) program to load data from a flat file to a DB2 UDB for AS/400 table. The load operation appends new data to the end of existing data in the table.

Before you define this step subtype, you must connect the step to a warehouse source and a warehouse target in the Process Modeler.

Acceptable source files for the AS/400 implementation of VWPLOADI are AS/400 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 program 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: To use this program, you must have the following PTFs applied to the AS/400 agent site:

Moving data

Table 20. PTF requirements

OS/400 system	Product ID	Required PTF	PTFs for VWPLOADI or VWPLOADR
Version 4 Release 2	5769SS1	SF51534	SF46911, SF46976, SF49466
Version 4 Release 3	5769SS1	SF51535	SF49467
Version 4 Release 4	5769SS1	none	SF56124

These PTFs provide the AS/400 CPYFRMIMPF and CPYTOIMPF commands (LOAD and EXPORT), which are required to run VWPLOADI program. These PTFs also install the online help for these commands.

Security: The user profile under which this program and the warehouse agent run must have at least read/write authority to the table that is to be loaded.

Requirements: The following requirements apply to the VWPLOADI program. 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 AS/400 command prompt, and press F1.

1. The Data Warehouse Center definition for the agent site that is running the program 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 program detects a failure during processing, the table is emptied. If the load process generates warnings, the program returns as successfully completed.
3. The default behavior for VWPLOADI 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.

To define values for a step that uses this program:

1. Open the step notebook.
2. Specify general information about the program. For more information, see “Providing general information about a step” on page 149.
3. Click the **Parameters** tab.

4. Optional: To specify the delimiters of your load program, click **Specify only the delimiters of the MODSTRING**. Otherwise, the default entries in the **Record**, **Column**, **Character strings**, and **Decimal** point fields are used.

Specify the delimiters for your load program:

- In the **Record** list, specify how you want to indicate the end of record.
 - In the **Column** list, specify the character that you want to use as a column delimiter.
 - In the **Character strings** field, specify the character that you want to use to indicate character strings.
 - In the **Decimal point** field, click the character that you want to use to indicate a decimal point.
5. Optional: To add parameters to the program, click **Type Modstring parameters**. Type your parameters in the field under the radio button. 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:

- 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 AS/400 online help for the CPYFRMIMPF command.

The format for the FileMod string is:

- a. 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.
- b. Each keyword must be followed immediately by its value. The value must be enclosed in parentheses.
- c. 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 FLDDLMM command must have the values enclosed by two single quotation marks. The Data Warehouse Center generates the AS/400 CALL statement for VWPLOADI in the form:
 CALL PGM(QIWH/VWPLOADI) PARM('fromfile' 'totable' 'filemodstring')

Moving data

Two single quotation marks together tell the AS/400 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.

6. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
7. Click **OK** to save your changes and close the step notebook.

Traces and diagnostic information: The VWPLOADI program provides two kinds of diagnostic information:

- The return code, as documented in the Data Warehouse Center Concepts online help
- The VWPLOADI trace

Important: Successful completion of this program does not guarantee that the data was transferred correctly. For stricter error handling, use the ERRVLV parameter.

Reading the VWPLOADI trace file: The VWPLOADI trace files are located in the Integrated File System in the /QIBM/UserData/IWH directory.

The VWPLOADI trace file has the following name format:

VWxxxxxxx.VWPLOADI

where xxxxxxxx is the process ID of the VWPLOADI run that produced the file.

To view trace files from a workstation:

1. Use Client Access/400 to map your AS/400 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.

Viewing the VWPLOADI trace with Client Access/400: To use Client Access/400 to map an AS/400 system to a logical drive on an NT workstation:

1. Set up a Client Access/400 connection to your AS/400 system over TCP/IP.
2. Open the Windows NT 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 AS/400 system.

5. Click **OK**.

Requirement: If you use Client Access/400 to access the trace file, you must define the file extension .VWPLOADI to Client Access/400. Defining this extension allows Client Access/400 to translate the contents of files with this extension from EBCDIC to ASCII.

To define a file extension to Client Access/400:

1. From Windows NT, select **Start** → **Programs** → **IBM AS400 Client Access** → **Client Access Properties**.

The Client Access notebook opens.

2. Click the **Network Drives** tab.
3. In the **File extension:** field, type .VWPLOADR.
4. Click **Add**.
5. Click **Apply**.
6. Click **OK**.

You should now be able to load the file into any ASCII text editor or word processor.

AS/400 exceptions: If there was a failure of any of the system commands issued by VWPLOADI, then there will be an exception code recorded in the VWPLOADI trace file. To get an explanation for the exception:

1. At an AS/400 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 VWPLOADR trace file contains the information that you need to issue the WRKJOB command.

Moving data

To view the spool file, you can cut and paste the name of the message file to an AS/400 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.

Defining values for a DB2 UDB for AS/400 Data Load Replace (VWPLOADR) program

Use the DB2 UDB for AS/400 Data Load Replace (VWPLOADR) program to load data from a flat file to a DB2 UDB for AS/400 table. The load operation completely replaces existing data in the table.

Before you define this step subtype, you must connect the step to a warehouse source and a warehouse target in the Process Modeler.

Acceptable source files for the AS/400 implementation of the VWPLOADR program are AS/400 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. 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.

The Column Mapping page is not available for this step.

Prerequisites: To use this program, you must have the PTFs listed on Table 20 on page 176 applied to the AS/400 agent site. These PTFs provide the AS/400CPYFRMIMPF and CPYTOIMPF commands (LOAD and EXPORT). These are the commands that makes the VWPLOADR program work. These PTFs also install the online help for these commands.

Security: The user profile under which this program and the warehouse agent run must have at least read/write authority on the table that is to be loaded.

Requirements: The following requirements apply to the VWPLOADR program. 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 AS/400 command prompt and press F1.

- The Data Warehouse Center definition for the agent site that is running the program 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.
- If the program detects a failure during processing, the table will be emptied. If the load generates warnings, the program returns as successfully completed.
- This implementation of the VWPLOADR program differs from VWPLOADR on other platforms. Specifically, it does not delete all loaded records if the load operation fails for some reason.
Normally, this program 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 VWPLOADR 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.

To define a process to load a local file into a local table. This process will start the warehouse program AS/400 Load with Replace 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 (see iSeries Load with Replace documentation for naming rules).
2. Create a step with the warehouse-supplied AS/400 Load with Replace program. For more information about creating an AS/400 Load with Replace step, see “To define values for a step that uses this program”
3. Select your flat-file source, and add the source file to the step.
Restrictions: You cannot use SQL statements on flat files. SQL is not supported in Version 7 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 AS/400 Load with Replace warehouse program.
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.

To define values for a step that uses this program:

1. Open the step notebook.

Moving data

2. Specify general information about the program. For more information, see “Providing general information about a step” on page 149.
3. Click the **Parameters** tab.
4. Optional: If you want to specify the delimiters of your load program, click **Specify only the delimiters of the MODSTRING**. If you do not click this radio button, the default entries in the Record, Column, Character strings, and Decimal point fields are assumed to be correct.
5. If you clicked **Specify only the delimiters of the MODSTRING**, specify the delimiters for your load program:
 - In the **Record** list, specify how you want to indicate the end of record.
 - In the **Column** list, specify the character that you want to use as a column delimiter.
 - In the **Character strings** field, specify the character that you want to use to indicate character strings.
 - In the **Decimal point** field, click the character that you want to use to indicate a decimal point.
6. Optional: To add additional parameters to the program, click **Type Modstring parameters**. Type your parameters in the field under the radio button.

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:

- 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 AS/400 online help for the CPYFRMIMPF command.

The format for the FileMod string is:

- a. 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.
- b. Each keyword must be followed immediately by its value. The value must be enclosed in parentheses.
- c. Each keyword must be separated from the next keyword by a space.

Attention: Certain parameters require that you enclose values in two single quotation marks. For example, the FLDDLML command must have

the values enclosed by two single quotation marks. You must do this because the Data Warehouse Center generates the AS/400 CALL statement for VWPLOADI in the form:

```
CALL PGM(QIWH/VWPLOADI) PARM('fromfile' 'totable' 'filemodstring')
```

Two single quotation marks together tells the AS/400 command prompt processor that your parameter value contains a single quotation mark character. This prevents the command line processor from confusing your single quotation mark character with the regular end-of-parameter marker.

7. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
8. Click **OK** to save your changes and close the step notebook.

Traces and diagnostic information: The VWPLOADR program provides two kinds of diagnostic information:

- The return code, as documented in the Data Warehouse Center Concepts online help
- The VWPLOADR trace

Important: Successful completion of this program does not guarantee that the data was transferred correctly. For stricter error handling, use the ERRVLV parameter.

Reading the VWPLOADR trace file: The VWPLOADR trace files are located in the Integrated File System in the /QIBM/UserData/IWH directory.

The VWPLOADR trace file has the following name format:

```
VWxxxxxxx.VWPLOADR
```

where xxxxxxxx is the process ID of the VWPLOADR run that produced the file.

To view trace files from a workstation:

1. Use Client Access/400 to map your AS/400 root file system to a logical drive, or use FTP to copy the file to the workstation.
For information about using Client Access/400, see “Viewing the VWPLOADR trace via Client Access/400”.
2. Open the trace file with a text editor to view the information.

Viewing the VWPLOADR trace via Client Access/400: To use Client Access/400 to map an AS/400 system to a logical drive on an NT workstation:

1. Set up a Client Access/400 connection to your AS/400 system over TCP/IP.

Moving data

2. Open the Windows NT File Explorer.
3. From the Explorer menu, select **Tools** → **Map Network Drive**.
4. Type the pathname:
 \\hostname\.

where hostname is the fully qualified TCP/IP host name of your AS/400 system.

5. Click **OK**.

Requirement: If you use Client Access/400 to access the trace file, you must define the file extension .VWPLOADR to Client Access/400. Defining this extension allows Client Access/400 to translate the contents of files with this extension from EBCDIC to ASCII.

To define a file extension to Client Access/400:

1. From Windows NT, click **Start** → **Programs** → **IBM AS400 Client Access** → **Client Access Properties**.

The Client Access notebook opens.

2. Click the **Network Drives** tab.
3. Type .VWPLOADR in the **File extension:** field.
4. Click **Add**.
5. Click **Apply**.
6. Click **OK**.

You should now be able to load the file into any ASCII text editor or word processor.

AS/400 exceptions: If there was a failure of any of the system commands issued by VWPLOADR, then there will be an exception code recorded in the VWPLOADR trace file. To get an explanation for the exception:

1. At an AS/400 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
```

Recovery . . . : being copied.
 See the messages previously listed.
 Correct the errors, and then try the
 request again.

The second line in the VWPLOADR trace file contains the information that you need to issue the WRKJOB command.

To view the spool file, you can copy and paste the name of the message file to an AS/400 command prompt after the WRKJOB command, and press Enter. View the spool file for the job to get additional information about any errors that might have occurred.

Defining values for a DB2 for OS/390 Load program

Use the DB2 for OS/390 Load warehouse program to load records into one or more tables in a table space.

To define a values for a step that uses this warehouse program:

1. Define a warehouse target where the DSNUTILS stored procedure will run.
2. Create a step that uses this warehouse program.
3. Connect the step to the target.
4. Open the step notebook.
5. Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149 .
6. On the Parameters page, if you want to preformat pages in the table space and index spaces that are associated with the table, select the **Entire table space and index spaces are preformatted** check box to have the free pages between the high-used RBA (or page) to the high-allocated RBA be preformatted in the table space and index spaces associated with the table. The preformatting occurs after the data is loaded and the indexes are built.
7. To specify whether records are to be loaded into an empty or non-empty table space, select the **Specify RESUME option at table space level** check box.
 - 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.
 - 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

Moving data

current end of data in the table space. Space occupied by rows marked as deleted or by rows of dropped tables is not reused.

If the table space is not empty, and you are not replacing contents (the **Reset table space and indexes to empty before load** check box is not checked), a DB2 UDB for OS/390 and z/OS message is issued, and the utility job step terminates with a job step condition code of 8. For nonsegmented table spaces, space occupied by rows that are marked as deleted or by rows of dropped tables is not used.

8. If you want the newly loaded rows to replace all existing rows of all tables in the table space, and not just those of the table that you are loading, select the **Reset table space and indexes to empty before load** check box (LOAD REPLACE) .
9. Specify one of the **Input data file type** radio buttons to select the code page type for the input data. If you specify ASCII, numeric, date, time, and time stamp internal formats are not affected.
10. Select the **Format of the input records** check box to identify the format of the input record. The format must be compatible with DB2 unload format (UNLOAD) or with SQL/DS™ unload format. This action uniquely determines the format of the input. No field specifications are allowed in an INTO TABLE option.

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 unloaded by the REORG utility are loaded into the tables from which they were unloaded. Do not add or change 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.

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.

11. Select the **Allow no substitution characters in a string** check box to prohibit substitution characters from being placed in a string as a result of a conversion. When this check box is selected and DB2 UDB for OS/390 and z/OS determines that a substitution character has been placed in a string as a result of a conversion, it performs one of the following actions:
 - If discard processing is active, DB2 issues message DSNU310I and places the record in the discard file.
 - If discard processing is not active, DB2 issues message DSNU334I, and the utility abnormally terminates.

12. Select the **CCSIDs for the input file** check box to specify up to three coded character set identifiers (CCSIDs) for the input data. Type a valid integer value in one or all of the fields. The default CCSID is the one that you chose when you installed DB2 Universal Database for OS/390. If any of the fields relating to the CCSID are left blank, the field will use the installation default. You can specify SBCS, DBCS (double byte character set) data, or mixed DBCS (graphics and double byte character sets).
13. Select the **Enforce check** and **referential constraints** check boxes to have DB2 notify you when there are check constraints and referential constraints. When constraints are enforced, and the loading operation detects a violation, DB2 deletes the errant row and issues a message to identify it. If you choose to enforce constraint, and referential constraints exist, sort input and sort output data sets are required.
14. In the **Maximum number of records to be written on discard data set** field, specify a maximum number of source records to be written on the discard data set. The value can range from 0 to 2,147,483,647. If the maximum number is reached, LOAD processing abnormally ends, the discard data set is empty, and you cannot see which records were discarded. The default is 0, which means that there is no maximum. In this case, the entire input data set can be discarded.
15. Type the name of the discard data set in the **Discard data set** field. This is a work data set that is required if you specify a number of records to be discarded. This data set holds copies of records that did not get loaded. It also holds copies of records that were loaded, then removed. It must be a sequential data set that is readable by BSAM services. SYSDISC is the data definition name that is associated with the data set name.
16. Type the name of a work data set for error processing in the **Error data set** field. This data set stores information about errors that occur in the load processing. This field is required if you specified a number of records to be discarded during load processing. SYSERR is the data definition name that is associated with the data set name.
17. Click **Advanced** to open the DB2 for OS/390 Load Table Space notebook. You must click this push button or the load will fail. From this notebook you can specify additional options for loading data into a table. You can also choose to gather statistics for a table space, index, or both. The statistics are stored in the DB2 for OS/390 catalog.
18. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
19. Click **OK** to save your changes and close the step notebook.

Moving data

Copying data between DB2 for z/OS and OS/390 tables using the LOAD utility

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.

To copy data between DB2 for z/OS and OS/390 tables using the LOAD utility:

1. Use the Generic interface 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 Generic DSNUTILS interface 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.

For more detailed information about DSNUTILS and the DB2 utilities available for OS/390 or z/OS, see the *DB2 for z/OS and OS/390 Utility Guide and Reference*.

Manipulating files

The sections that follow describe how to define the different step subtypes for the warehouse file programs.

Defining values for the Copy File using FTP (VWPRCPY) program

Use the warehouse program Copy file using FTP (VWPRCPY) to copy files on the agent site to and from a remote host.

Before you copy files to OS/390, you must allocate their data sets. You cannot transfer VSAM data sets. When you define a step that uses this warehouse program, select a source file and select a target file. One file must be stored on the agent site, and the other must be stored on the OS/390 system.

This warehouse program is available for the following operating systems:

- Windows NT
- AIX
- Solaris Operating Environment
- AS/400
- OS/390
- OS/2

The Column Mapping page is not available for this step subtype.

To define a step that runs this warehouse program:

1. Open the step notebook.
2. Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149.
3. On the Parameters page, click either **Copy files from the remote host to the agent site (GET method)** or **Copy files from the agent site to the remote host (PUT method)**.
4. In the Transfer type area, click either **ASCII** or **Binary**.
5. In the **Remote system name** field, type the name of the remote system from which or to which to copy files.
6. In the **Remote user ID** field, type the user ID to use to connect to the remote host.
7. In the **Remote password** field, type the password for the user ID.
8. In the **Verify remote password** field, type the password again.

Moving data

9. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
10. Click **OK** to save your changes and close the step notebook.

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
```

Defining values for a Run FTP Command File program (VWPFTP)

Use the warehouse program Run FTP command file (VWPFTP) 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.

This warehouse program is available for the following operating systems:

- Windows NT
- AIX
- Solaris Operating Environment
- AS/400
- OS/390
- OS/2

The Column Mapping page is not available for this step.

To define a process that runs this warehouse program:

1. In the main Data Warehouse Center window, right-click on the **Warehouse Sources** folder and click **Define** → **Flat File** → **Remote files**. On the Files page, click **Advanced**, and specify the host name, user ID, and password.
Restrictions: You cannot use SQL statements on flat files. SQL is not supported in Version 7 of the iSeries warehouse agent.
2. Define a local flat file target on iSeries.
3. Create a step with the warehouse Copy File using FTP program.
4. Connect the step to a remote source and local target.
5. Open the step notebook.

6. Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149.
7. On the Parameters page, in the **Remote System Name** field, type the name of the remote system to access.
8. In the **FTP command file (full path name)** field type the path and file name of the FTP command file.
In the FTP command file, you must supply the following information, in the order listed, and on separate lines:
 - a. User ID
 - b. Password
 - c. One or more FTP commands, each on a separate line

In the following example, you use FTP to log on to the remote host using the user ID and password, get the remote file, and put it in the specified local directory:

```
nst1
password
get /etc/services d:/udprcpy.out
quit
```

9. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
10. Click **OK** to save your changes and close the step notebook.
11. Link this step to the AS/400 Load with Replace step.
12. Connect the local target file as a source to the AS/400 Load with Replace step.
13. Connect the local target table to the step.

The two combined steps can load a remote flat file on a local file on iSeries and then load data into a table.

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
```

Moving data

Defining values for a Submit OS/390 JCL jobstream (VWPMVS) program

Use the Submit OS/390 JCL jobstream (VWPMVS) warehouse program to submit a JCL jobstream that resides on OS/390 to an OS/390 system for execution.

The job must have a MSGCLASS and SYSOUT routed to a held output class.

Before you use the Submit OS/390 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.

This warehouse program runs successfully if the OS/390 host name, user ID, and password are correct. If you want to test the validity of the results generated by the JCL, you must write your own testing logic.

The Submit OS/390 JCL Jobstream warehouse program also receives the JES log file on 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.

The Submit OS/390 JCL Jobstream warehouse program requires TCP/IP 3.2 or later installed on OS/390. Verify that the FTP service is enabled before using the program.

When you define a step that uses this warehouse program, do not specify a source or target table for the step.

This warehouse program is available for the following operating systems:

- Windows NT
- AIX
- Solaris Operating Environment
- OS/390
- OS/2

The Column Mapping page is not available for this step.

To define values for a step that runs this warehouse program:

1. Open the step notebook.
2. Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149.
3. On the Parameters page, in the **MVS system name** field, type the name of the OS/390 host on which to run the job.

4. In the **MVS User ID** field, type the user ID to use to connect to the OS/390 host.
5. In the **MVS password** field, type the password for the user ID.
6. In the **Verify MVS password** field, type the password again.
7. In the **MVS JCL File** field, type the name of the JCL file to submit. The name must consist of the user ID plus one character.
8. In the **Local spool file full path name** field, type the path and file name of the file on the agent site that is to receive the JES log file.
 If a .netrc file does not exist, the warehouse program creates the file in the same directory as the JES file. If a .netrc file exists, the warehouse program renames the existing file and creates a new file. When the program finishes processing, it deletes the new .netrc file that it created and renames the original file to .netrc.
9. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
10. Click **OK** to save your changes and close the step notebook.

If you are having problems accessing a remote file on a secure UNIX system, verify that the home directory of the user ID contains a .netrc file. The .netrc file must contain an entry that includes the host name of the agent site and the remote user ID that you want to use.

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
```

Setting up replication in the Data Warehouse Center

Setting up and running replication with Data Warehouse Center requires that the replication control tables exist on both the warehouse control database and the warehouse target databases.

The replication control tables must exist in both the control and target databases. 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, they are automatically created in a database when you define a replication source in the Control Center.

To create the control tables, edit the SQL that is generated when you define a replication source. Remove the SQL statements that insert into the

Moving data

IBMSNAP_REGISTER table and the statements that create the CD table. You can also use DataJoiner Replication Administration (DJRA) to create control tables.

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

1. Optional: Install DJRA.

Note: DJRA is required for replication with non-IBM databases, and is recommended for AS/400 systems. For more information about installing the DJRA, see the *Replication Guide and Reference*.

2. Set up the Capture and Apply programs. For information about setting up the Capture and Apply programs, see *Replication Guide and Reference*.

You must bind the capture and apply programs on each database where they will be used. You do not need to create a password file. The Data Warehouse Center will automatically create a password file for the replication subscription.

Data Warehouse Center includes a JCL template for replication support. If you plan to use the zSeries warehouse agent to run the Apply program, you need to change the account and data set information in this template for your OS/390 or z/OS system. For information, see *Warehouse Manager Installation Guide*.

3. Define a replication source in the Control Center or DJRA.
4. Define a replication step in the Data Warehouse Center.
5. Start the Capture program on the same system as the source database.
6. Start the Apply program. It is recommended that you start the Apply program on the same system as the target database, but it is not required.

When you start the Apply program:

- Use the same Apply qualifier that you defined for the step
- Use the target server as the control server name when you start the Apply program

For more information about the parameters for the Apply program, see the *Replication Guide and Reference*.

7. Promote the replication step to test or production in the Data Warehouse Center.

Replicating tables

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

OS/390 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 DataJoiner 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.

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

Produces target tables that are read-only copies of the replication source with no replication control columns added. These 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

Produces target tables that are read-only copies of the replication source with a timestamp column added. The timestamp column is originally null. When changes are replicated, values are added to indicate the time when updates are made. Use this type of table if you want to keep track of the time of changes.

Base aggregate

Produces read-only tables that summarize the contents of a source table. Base aggregate replication tables are useful for tracking the state of a source table on a regular basis. Aggregate tables use SQL column functions (such as SUM and AVG) to compute summaries of the entire contents of the source tables or of the recent changes made to source table data.

Change aggregate

Produces read-only tables that summarize the contents of change data

Moving data

tables, instead of source tables. This type of replication is useful for tracking the changes made between each Apply program cycle.

Staging table

Produces read-only tables that contain data from committed transactions. Also called consistent change-data tables (CCD tables), these tables contain different data if they are condensed, noncondensed, complete, or noncomplete.

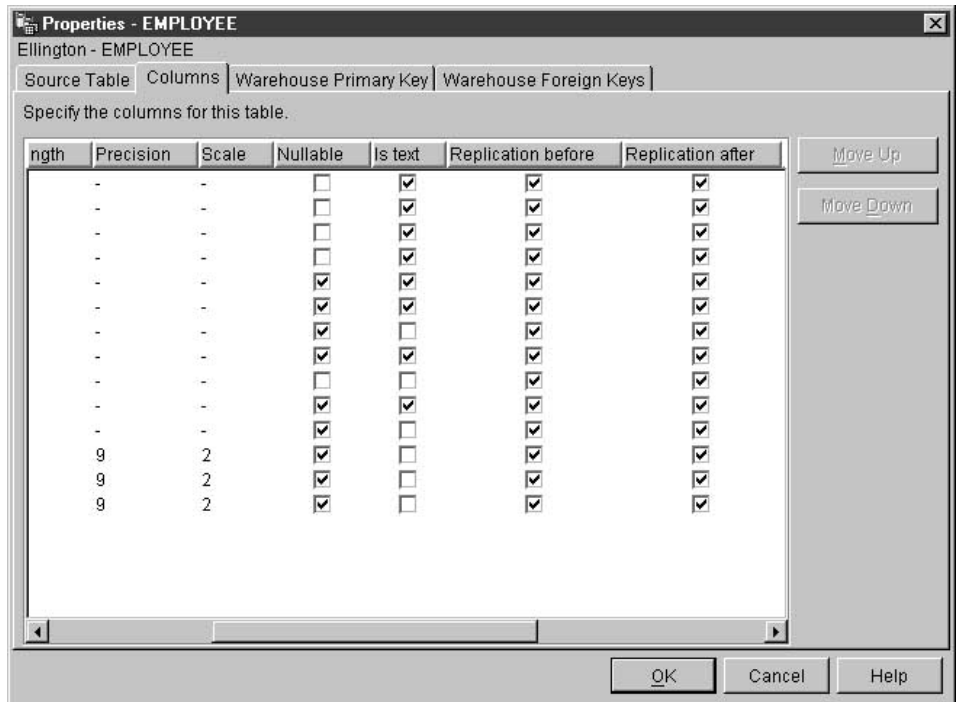
- A *condensed staging table* contains only the most current value for a row. The Apply program updates only rows that are already in condensed tables. Condensed tables are useful for staging changes to remote locations and for summarizing hot-spot updates before they are replicated to targets.
- A *noncondensed staging table* 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 all the rows that you want to replicate from 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 a replication step, promoting to test mode will create the target table and generate the subscription set. 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.

Defining a replication source in the Data Warehouse Center

You define replication sources in the same way that you define other relational sources with the Data Warehouse Center. In addition to the other database objects that can be returned from a database (tables, views, and system tables), you can choose to return replication-enabled tables and views. A table or view must be defined for replication using the DB2 Control Center or DJRA before it can be used as a replication source in the Data Warehouse Center. For instructions on defining a table or view as a replication source, see the *DB2 Replication Guide and Reference*.

When you define a replication-enabled table as a warehouse source table, the before and after image columns are identified by **before** or **after** after the column name.



If you choose to retrieve only replication-enabled tables from the source database, only the columns within the table that are enabled for replication will be retrieved.

For instructions on defining a replication source in the Data Warehouse Center, see “Defining a DB2 warehouse source” on page 54.

Defining a user copy, point-in-time, or base aggregate replication step

A source table used for a user copy or point-in-time replication step must have a primary key. Use the DB2 Control Center to define a primary key for each table that you want to include in a user copy or point-in-time replication step.

To define a user copy, point-in-time, or base aggregate replication step:

1. Define a process object.
2. Open the process object.
3. Add one or more warehouse sources.
4. Add one or more warehouse targets.
5. Open the step notebook.
6. Specify information for your step:

Moving data

- In the **Name** field, you can type a new name for the step. Otherwise, you can keep the name that the Data Warehouse Center automatically supplied for the step.
 - Optional: In the **Administrator** field, type the name of the person who is responsible for the maintenance of this step.
 - Optional: In the **Description** field, type a business description for your step. This description can be a maximum of 254 characters.
 - Optional: In the **Notes** field, type detailed information that might be helpful to users who can access this step.
7. On the Parameters page, select the columns that you want to replicate from the **Available columns** list, and click >. The columns that you select move to the **Selected columns** list. The **Available columns** list shows only the columns that have been enabled for change capture.
To include all of the items in the **Available columns** list, click >>.
 8. Optional: Click **Add Calculated Column** to open a window in which you can create derived columns. The derived columns that you create are displayed in the **Selected columns** list.
 9. Optional: To select which rows are to be replicated, write a WHERE statement to subselect rows.
 10. On the Column Mapping page, map the output columns that you defined on the Parameters page to columns in 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. See “Defining column mapping information” on page 150 for information about the tasks that can be performed using the Column Mapping page.

If the Parameters page produces no output columns, or if this step is not linked to a target table, you cannot use this page to map your columns. Some steps will not allow you to change the column mapping.
 11. On the Processing Options page, select an agent site where you want your step to run from the **Agent Site** drop-down list. The selections in this list are agent sites that are common to the source tables and the target table.
 12. The Population type for replication steps can have only one value, **Replicate**.
 13. If you want the option to run your step at any time, select the **Run on demand** check box. Your step must be in test or production mode before you can run it.
 14. Optional: Select the **Populate externally** check box if the step is populated externally, meaning that it is started in some way other than

by the Data Warehouse Center. The step does not require any other means of running in the Data Warehouse Center in order for you to change the mode to production.

If the **Populate externally** check box is not selected, then the step must have a schedule, be linked to a transient table that is input to another step, or be started by another program in order for you to change the mode to production.

15. In the **Retry** area, specify how many times you want the step to run again if it needs to be retried, and the amount of time that you want to pass before the next run of the step.
16. In the **Replication control database** field, select the control database or subsystem that contains the replication control tables for the Apply program.
17. In the **Database type** list, select the database type for the replication control database.
18. In the **User ID** field, type the user ID to access the replication control database.
19. In the **Password** field, type the password for the user ID that will access the database.
20. In the **Verify password** field, type the password again.
21. In the **Subscription set name** field, type the name of the subscription set. This name can contain a maximum of 18 characters and can be an ordinary or delimited qualifier.
22. Optional: In the **Apply qualifier** field, type the name of the apply qualifier. The apply qualifier name must be unique for every replication step that is defined. If you do not specify an apply qualifier, the Data Warehouse Center generates one for you.
23. Optional: In the **Event name** field, type the event name. The event name is the name of the event that is posted in the event table that the Apply program reads. The event name must be unique for every replication step that is defined. If you do not specify an event name, the Data Warehouse Center generates one for you.
24. In the **Blocking factor** field, specify the number of minutes worth of change data that can be replicated during a subscription cycle.
25. Click **OK** to save your changes and close the notebook.
26. Link the step to the warehouse sources.
27. Link the steps to the warehouse targets.
28. Start the Capture program. You must start the Capture program before you start the Apply program. To start the Capture program, type the following command in a DOS window:
 ASNCCP source-database COLD PRUNE

Moving data

The COLD parameter specifies a COLD start and deletes any existing data in the CD tables. The PRUNE parameter tells the capture program to maintain the IBMSNAP_PRUNCNTL table. With the Capture program running, go on to step 29.

29. Start the Apply program. You can start the Apply program at any time after you have started the Capture program. For information about starting the Apply program, see the *Replication Guide and Reference*.

30. Promote the step to test mode.

31. Run the step to test it.

When the step runs, the event record in the IBMSNAP_SUBS_EVENT table is updated and the subscription record in IBMSNAP_SUBS_SET is active. If you update the original source table after the event record is updated and the subscription record is active, the changed data is moved into the CD table. If you run the replication step after the changed data is moved into the CD table, and the Apply program runs again, the changed data is moved from the CD table to the target table.

32. Schedule the step.

33. Promote the step to production mode.

Defining a change aggregate replication step

A change aggregate replication step produces tables that work with the change data in the control tables, but not with the contents of the source table.

To define a change aggregate replication step:

1. Define a process object.
2. Open the process object.
3. Add one or more warehouse sources.
4. Add one or more warehouse targets.
5. Open the step notebook.
6. Specify information for your step:
 - In the **Name** field, you can type a new name for the step, or, you can keep the name that the Data Warehouse Center automatically supplied for the step.
 - Optional: In the **Administrator** field, type the name of the person who is responsible for the maintenance of this step.
 - Optional: In the **Description** field, type a business description for your step. This description can be a maximum of 254 characters.
 - Optional: In the **Notes** field, type detailed information that might be helpful to users who can access this step.
7. On the Parameters page, select the columns that you want to replicate from the **Available columns** list, and click >. The columns that you select

move to the **Selected columns** list. The **Available columns** list shows only the columns that are enabled for change capture.

If you want to include all of the items in the **Available columns** list, click >>.

8. Optional: Click **Add Calculated Column** to open a window in which you can create derived columns. The derived columns that you create are displayed in the **Selected columns** list.
9. Optional: To select which rows are to be replicated, write a WHERE statement to subselect rows.
10. Optional: To add calculated columns, add a GROUP BY statement. You can group rows according to the group defined in the GROUP BY statement.
11. On the Column Mapping page, map the output columns that you defined on the Parameters 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 that are linked to the step are listed on the right side of the page. See “Defining column mapping information” on page 150 for information about the tasks that can be performed using the Column Mapping page.

If the Parameters page produces no output columns, or if this step is not linked to a target table, you cannot use this page to map your columns. Some steps will not allow you to change the column mapping.
12. On the Processing Options page, select an agent site where you want your step to run from the **Agent Site** drop-down list. The selections in this list are agent sites that are common to the source tables and the target table.
13. The Population type for replication steps can have only one value, **Replicate**.
14. If you want the option to run your step at any time, select the **Run on demand** check box. Your step must be in test or production mode before you can run it.
15. Optional: Select the **Populate externally** check box if the step is populated externally, meaning that it is started in some way other than by the Data Warehouse Center. The step does not require any other means of running in the Data Warehouse Center in order for you to change the mode to production.

If the **Populate externally** check box is not selected, then the step must have a schedule, be linked to a transient table that is input to another step, or be started by another program in order for you to change the mode to production.

Moving data

16. In the **Retry** area, specify how many times you want the step to run again if it needs to be retried, and the amount of time that you want to pass before the next run of the step.
17. In the **Replication control database** field, select the control database or subsystem that contains the replication control tables for the Apply program.
18. In the **Database type** list, select the database type for the replication control database.
19. In the **User ID** field, type the user ID to access the replication control database.
20. In the **Password** field, type the password for the user ID that will access the database.
21. In the **Verify password** field, type the password again.
22. In the **Subscription set name** field, type the name of the subscription set. This name can contain a maximum of 18 characters and can be an ordinary or delimited qualifier.
23. Optional: In the **Apply qualifier** field, type the name of the apply qualifier. The apply qualifier name must be unique for every replication step that is defined. If you do not specify an apply qualifier, the Data Warehouse Center generates one for you.
24. Optional: In the **Event name** field, type the event name. The event name is the name of the event that is posted in the event table that the Apply program reads. The event name must be unique for every replication step that is defined. If you do not specify an event name, the Data Warehouse Center generates one for you.
25. In the **Blocking factor** field, specify the number of minutes worth of change data that can be replicated during a subscription cycle.
26. Click **OK** to save your changes and close the notebook.
27. Link the step to the warehouse sources.
28. Link the steps to the warehouse targets.
29. Start the Capture program by entering the following command in a DOS window:

```
ASNCCP source-database COLD PRUNE
```

The **COLD** parameter specifies a **COLD** start and will delete any existing data in the CD tables. The **PRUNE** parameter tells the capture program to maintain the **IBMSNAP_PRUNCNTL** table. Leave the Capture program running. You must start the Capture program before you start the Apply program.
30. Start the Apply program. You can start the Apply program at any time after you have started the Capture program. For information about starting the Apply program, see the *Replication Guide and Reference*.

31. Promote the step to test mode.

32. Run the step to test it.

When the step runs, the event record in the IBMSNAP_SUBS_EVENT table is updated and the subscription record in IBMSNAP_SUBS_SET is posted active. If you update the original source table after the event record is updated and the subscription record is active, the changed data is moved into the CD table. If you run the replication step after the changed data is moved into the CD table, and the Apply program runs again, the changed data will be moved from the CD table to the target table.

33. Schedule the step.

34. Promote the step to production mode.

Defining a staging table replication step

A staging table replication step produces read-only tables that contain data from committed transactions. A source table that is used for a staging table replication step must have a primary key. Use the DB2 Control Center to define a primary key for each table that you want to include in a staging table replication step.

To define a staging table replication step:

1. Define a process object.
2. Open the process object.
3. Add one or more warehouse sources.
4. Add one or more warehouse targets.
5. Open the step notebook.
6. Specify information for your step:
 - In the **Name** field, you can type a new name for the step. Otherwise, you can keep the name that the Data Warehouse Center automatically supplied for the step.
 - Optional: In the **Administrator** field, type the name of the person who is responsible for the maintenance of this step.
 - Optional: In the **Description** field, type a business description for your step. This description can be a maximum of 254 characters.
 - Optional: In the **Notes** field, type detailed information that might be helpful to users who can access this step.
7. On the Parameters page, select the columns that you want to replicate from the **Available columns** list, and click >. The columns that you select move to the **Selected columns** list. The **Available columns** list shows only the columns that are enabled for change capture.

To include all of the items in the Available columns list, click >>.

Moving data

8. Optional: Click **Add Calculated Column** to open a window in which you can create derived columns. The derived columns that you create are displayed in the **Selected columns** list.
9. Optional: To select which rows are to be replicated, write a **WHERE** statement to subselect rows.
10. Optional: Click **Staging table options** to change any properties for your table. This option is available if you choose a replication type of Staging Table.
11. On the Column Mapping page, map the output columns that you defined on the Parameters 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. See “Defining column mapping information” on page 150 for information about the tasks that can be performed using the Column Mapping page.

If the Parameters page produces no output columns, or if this step is not linked to a target table, you cannot use this page to map your columns. Some steps will not allow you to change the column mapping.
12. On the Processing Options page, select an agent site where you want your step to run from the **Agent Site** drop-down list. The selections in this list are agent sites that are common to the source tables and the target table.
13. The Population type for replication steps can have only one value, **Replicate**.
14. If you want the option to run your step at any time, select the **Run on demand** check box. Your step must be in test or production mode before you can run it.
15. Optional: Select the **Populate externally** check box if the step is populated externally, meaning that it is started in some way other than by the Data Warehouse Center. The step does not require any other means of running in the Data Warehouse Center in order for you to change the mode to production.

If the **Populate externally** check box is not selected, then the step must have a schedule, be linked to a transient table that is input to another step, or be started by another program in order for you to change the mode to production.
16. In the **Retry** area, specify how many times you want the step to run again if it needs to be retried, and the amount of time that you want to pass before the next run of the step.
17. In the **Replication control database** field, select the control database or subsystem that contains the replication control tables for the Apply program.

18. In the **Database type** list, select the database type for the replication control database.
19. In the **User ID** field, type the user ID to access the replication control database.
20. In the **Password** field, type the password for the user ID that will access the database.
21. In the **Verify password** field, type the password again.
22. In the **Subscription set name** field, type the name of the subscription set. This name can contain a maximum of 18 characters and can be an ordinary or delimited qualifier.
23. Optional: In the **Apply qualifier** field, type the name of the apply qualifier. The apply qualifier name must be unique for every replication step that is defined. If you do not specify an apply qualifier, the Data Warehouse Center generates one for you.
24. Optional: In the **Event name** field, type the event name. The event name is the name of the event that is posted in the event table that the Apply program reads. The event name must be unique for every replication step that is defined. If you do not specify an event name, the Data Warehouse Center generates one for you.
25. In the **Blocking factor** field, specify the number of minutes worth of change data that can be replicated during a subscription cycle.
26. Click **OK** to save your changes and close the notebook.
27. Link the step to the warehouse sources.
28. Link the steps to the warehouse targets.
29. Start the Capture program. You must start the Capture program before you start the Apply program. To start the Capture program enter the following command in a DOS window:

```
ASNCCP source-database COLD PRUNE
```

The COLD parameter specifies a COLD start and deletes any existing data in the CD tables. The PRUNE parameter tells the capture program to maintain the IBMSNAP_PRUNCNTL table. With the Capture program still running, go on to step 30.

30. Start the Apply program. You can start the Apply program at any time after you have started the Capture program. For information about starting the Apply program, see the *Replication Guide and Reference*.
31. Promote the step to test mode.
32. Run the step to test it.

When the runs, the event record in the IBMSNAP_SUBS_EVENT table is updated and the subscription record in IBMSNAP_SUBS_SET is active. If you update the original source table after the event record is updated and the subscription record is active, then the changed data will be

Moving data

moved into the CD table. If you run the replication step after the data is moved into the CD table, and the Apply program runs again, the changed data will be moved from the CD table to the target table.

33. Schedule the step.
34. Promote the step to production mode.

Extracting data 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 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 or Windows 2000 (with Service Pack 1 or later). The SAP R/3 server can be on any supported operating system.

Defining an SAP source

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.

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 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 window, 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 window, 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 R/3 server.

8. On the Business Objects page of the Define Warehouse Source window, 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 R/3 system.

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

Defining the properties of each source business object

After you define the SAP source to the Data Warehouse Center, you can define properties for 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 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.

Defining an SAP step

To define the properties for the step:

1. In the Process Model window, click the **SAP** icon, then click the canvas to add the step to the process.
2. Link the step to an SAP business object.
3. Right-click on the SAP Step on the canvas, and click **Properties**.
4. In the **Name** field, type a name for the SAP step.

Moving data

5. Optional: In the **Administrator field**, type the name of the person who is responsible for the step definition.
6. Optional: In the **Description field**, type a brief description of the step.
7. Optional: In the **Notes field**, type information about the step that might be useful to users who can access the source properties.
8. On the Input Parameters page of the Properties notebook, you see a list of the parameters, with the data type, length, precision and scale (if applicable), default value, and an indication of whether the parameter is mandatory. You can edit the values, if necessary, to specify the filtering criteria for the data to be imported from the SAP system.
9. On the Output Parameters page, select the parameters that you want to include in this extract step from the list of available parameters for the business object.
10. On the Column Mapping page, you can either generate a default table to match the selected output parameters or map the output parameters to an existing table. The output parameters are treated as source columns to be mapped to columns in the warehouse target. For more information, see “Defining column mapping information” on page 150.
11. On the Processing Options page, specify the population type, agent site, commit count, and retry count and interval. The commit count specifies that the SAP step will issue an SQL COMMIT statement after inserting the specified number of records into the target table. The default commit count is 0, which means that the SAP step will insert all records in one unit of work and then issue the COMMIT statement.

Extracting data from a WebSphere Site Analyzer database

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.

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, AIX, or the Solaris Operating Environment.

Defining a WebSphere Site Analyzer source

To define a new WebSphere Site Analyzer source to the Data Warehouse Center:

1. Right-click the **Warehouse Sources** folder, and click **Define**, then click **WebSphere Site Analyzer**. The WSA Webmart database must exist on the WSA system before you can define a WSA source in Data Warehouse Center. This is accomplished by successfully running the WSA data collection once.
2. On the Warehouse Source page of the Define Warehouse Source window, type the business name, and optionally the administrator name, description, and notes for the WSA source.
3. On the Agent Sites page of the Define Warehouse Source window, move one or more agent sites to the **Selected agent sites** box.
4. On the Database page of the Define Warehouse Source window, fill in the database-related fields for a DB2 database, or customize the ODBC connect string if the WSA webmart is an Oracle database.

For the WebSphere Site Analyzer properties area of the Database page, type the following information:

Site name

The project name that you specified in the Project Information notebook when you defined the WSA Project you are collecting Web traffic data for.

Host name

The name of the virtual host, or hosts, where your IBM WebSphere Site Analyzer is installed. This is usually the name of the workstation on which the IBM WebSphere Site Analyzer is installed.

Port number

The port number that the WebSphere Application Server Webserver is listening to. The default port number is 80. If you leave this field blank, a port number of 80 will be used when the WSA runs in a non-secure environment. A default port number of 443 is used when the WSA runs in a secure environment. The environment is considered a secure environment if it uses SSL for encryption.

User ID

The user ID that you use to log on to IBM WebSphere Site

Moving data

Analyzer. The user ID must be less than 128 characters in length. This field is optional, depending on the configuration of IBM WebSphere Site Analyzer.

Password

The password for the user ID that you use to log on to IBM WebSphere Site Analyzer. The password must be less than 64 characters in length. This field is optional, depending on the configuration of IBM WebSphere Site Analyzer.

5. On the Web Source Data page of the Define Warehouse Source notebook, select from a list of available data imports, tables, and views those specific items that you want to include in this new warehouse source.

Defining a Web traffic polling step

To define a Web traffic polling step:

1. In the Process Model window, click the **Web Traffic** icon, then click the canvas to add the step to the process.
2. Right-click the **Web Traffic Polling** step on the canvas, and click **Properties**.
3. On the Web Traffic Program page of the Properties notebook, type the business name, and optionally the administrator name, description, and notes for the Web traffic step.
4. On the Parameters page of the Properties notebook, click the warehouse source name and select one of the available WSA sources. Move one or more of the available data imports that you want to poll as part of this step to the list of **Selected data imports**. The available data imports are the ones that you specified when you defined the Web traffic source.
5. On the Processing Options page of the Properties notebook, specify the agent site, polling interval and duration, and agent site retry count and interval. The polling interval and polling duration specify how often the Web Traffic Polling step checks the data collection status of the selected data imports. The polling duration must be greater than the polling interval. The default for the polling duration is 60 minutes, and the default for the polling interval is 20 minutes. When the polling duration and the polling interval are set to the default values, the Web traffic step polls the WSA server a maximum of three times before returning with the last polling status. The step terminates successfully if the data collection status of all the selected data imports was successful.

The step remains enabled and is activated when the step is scheduled to run.

Chapter 7. Transforming data

This chapter describes the methods of transforming data that the Data Warehouse Center provides. It describes how to use SQL steps and warehouse transformers to transform data.

In many cases, there are multiple ways that you can transform data. For example, if you want to clean data, you have the following choices:

Table 21. Methods of cleaning data

Method	Description	For more information, see:
WHERE clause	Build an SQL WHERE clause to limit the rows that you extract from the source table.	“Filtering data” on page 217
Formulas and expressions	Use formulas and expressions to eliminate the data that you do not need and to create the data that you do need. Use the Expression window in SQL Assist to specify formulas, constants, and tokens.	“Adding calculated columns” on page 218
Clean Data transformer	Use the Clean Data transformer to perform rules-based find and replace operations on a table.	“Cleaning data” on page 220
Warehouse programs	Use warehouse programs to perform any functions on the source data that are not available from the previous methods.	“Chapter 13. Extending the Data Warehouse Center with user-defined programs” on page 309

Joining source columns

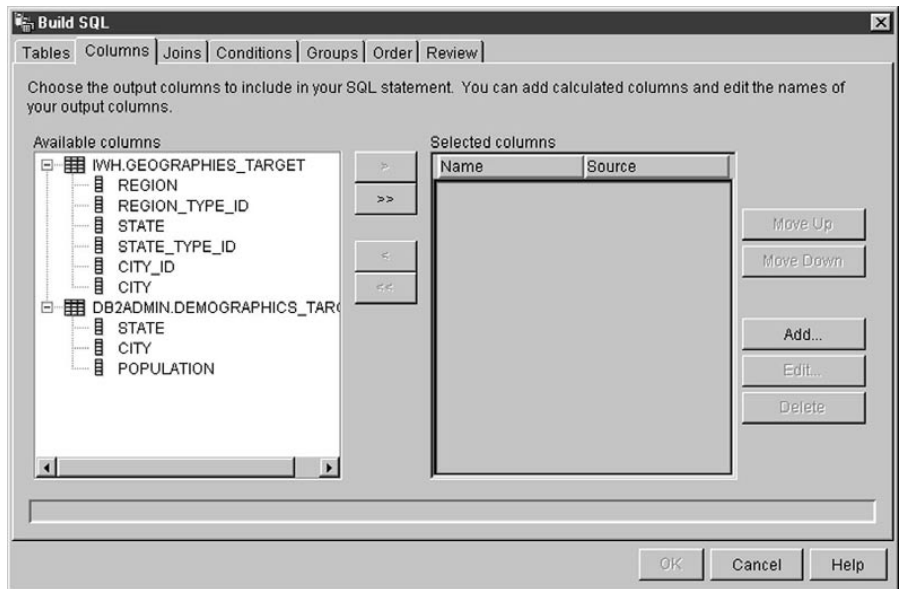
If you select multiple sources, you also need to specify how the rows of these columns will be associated in the join of the tables. You can use SQL Assist to define the join.

You must link the sources to the step before you define the join.

Transforming data

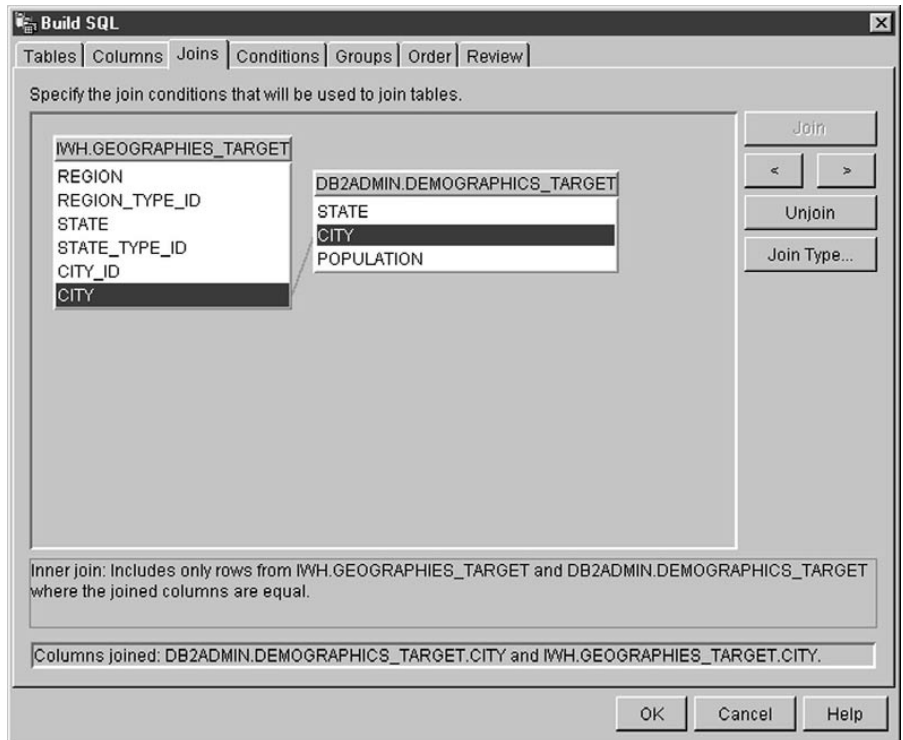
To join source tables:

1. Define an SQL step. (See “Selecting and inserting data” on page 167.)
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.
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.
To remove a join:
 - a. Select the joined columns, or navigate to the join that you want to remove by clicking > or <. A red line indicates the currently selected join. Other joins are indicated by blue lines.
 - b. Click **Unjoin**. The join line is removed.
13. Click the **Review** tab to view the SQL statement that you just built.
14. Click **OK**.
SQL Assist closes.

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**.
The Step Properties notebook closes.

Transforming data

Transforming codes

In many production environments, the source data includes encoded information. For example, you might use codes to refer to part numbers throughout your database. You also have a table that relates the part numbers to serial numbers and descriptions of each part. In these situations, you want the information in your warehouse to include the name and description of each part. To do this, you must combine the decoding table with the source data that contains the encoded part numbers.

First, you need to 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
```

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

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

Transforming data

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

Generating star joins

You can generate a *star join*, which is a join of source tables that are defined in a star schema. A *star schema* is a specialized design that consists of the following types of tables:

- *Dimension tables*, which describe aspects of a business
- A *fact table*, which contains the facts about the business

For example, if you have a mail-order business that sells books, some dimension tables are Customers, Books, Catalogs, and Fiscal_Years. The fact table contains information about the books that were ordered from each catalog by each customer during the fiscal year.

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.

For information about defining primary and foreign keys in the Data Warehouse Center, see “Defining a warehouse target” on page 115. For information about defining a star schema and exporting it to OLAP Integration Server, see “Chapter 15. Creating a star schema from within the Data Warehouse Center” on page 341.

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.

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 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.

Transforming data

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. To do this, click on a line in the **Values** list, then click **Add Variable**. The **Add Variable** window opens. Type the variable name, then click **OK**. The Add Variables window closes, and the variable is added 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.

You can edit the text in the **Conditions** field. You can remove search conditions by highlighting the portion of the condition that you want to remove in the **Conditions** field, and pressing the **Delete** key on your keyboard.

To build complex conditions, click **Advanced Expression**. The Expression Builder - Conditions window opens. For help on using the **Expression Builder**, see “Building expressions” in the online help.

Adding calculated columns

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.

The Data Warehouse Center lets you easily and accurately define steps that are summaries of source data. You can use standard SQL aggregation functions (AVG, COUNT, MAX, MIN, and SUM) and the SQL GROUP BY clause to create steps that summarize the source data.

Summary steps reduce the load on the network. They perform the aggregations on the source data before replicating the data across the network. You can also create composite steps that use summary techniques to summarize other steps. Summarizing reduces the size of the target warehouse that you create.

To create a step with this type of 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
```

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**.

The Expression Builder window opens.

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. To build an expression:

- a. 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.
- b. 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.
- c. To add a function to your expression:
 - 1) 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.

Transforming data

- 3) The Function Parameters window opens. Select the format of the function and specify the parameter values.
 - 4) Click **OK**. The Function Parameters window closes. The function and its parameters are displayed in the **Expression** field of the Expression Builder.
- d. To add a constant to your expression:
- 1) 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.
- e. 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.
- f. 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.
- g. Click the **Name** field of the new column, and type the name of the column.
- h. Press Enter.
- i. Click **Move Up** and **Move Down** to move the column to the appropriate position in the table.

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

Cleaning data

Use the 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 Clean Data transformer does not define rules or parameters for the carry over columns.

Use the 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 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

You can use the 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 Clean Data transformer, you must create a rules table for your clean type. A rules table designates the values that the 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.

Transforming data

Creating a rules table for a clean transformer

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 Clean Data transformer searches your source column for the value Desk. Wherever it finds the value Desk, it writes the value Chair in the corresponding field of the target column.

The 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	The first column contains find values. The second column contains replace values. The third column, if specified, contains rule-ordering information and is not required.

Discretize	3 - 4	<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>
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 Step notebook. You can change column names on the Column Mapping page of the Step notebook.

Defining the Clean Data transformer

To define a Clean Data transformer:

1. Open the step notebook.

Transforming data

2. Specify general information about the transformer. For more information, see “Providing general information about a step” on page 149.
3. On the Parameters page, define the clean transformations that you want performed when the step runs. Right-click anywhere on the Transformer definition table, and click **Add**. A row is added to the table.
4. Under the Input column heading in the row that you just added, click and select the name of the column that contains the data that will be cleaned.
5. In the **Clean type** list, select the type of clean transformation that will be performed on the input column when the step runs.

To copy your input column to the target table without making any changes, click **Carry Over**. The Rules list and Parameter list are not available. Proceed to step 7.

6. In the Rules list, click ... to specify the rules table that the step will use for the clean type that you selected. A window opens. The fields on the window vary depending on the selection you made. Use this window to define rules for the clean type that you selected:
 - To define rules for Find and replace:
 - a. Select a rules table from the **Rules table** list.
 - b. Select a column that contains find values. Click a column in the **Available columns** list, then click > next to the **Find column** field.
 - c. Select a column that contains replace values. Click a column in the **Available columns** list, then click > next to the **Replace column** field.
 - d. Optional: Select a column that contains the sequence in which to read rows in the rules table. Click a column in the **Available columns** list, then click > next to the **Order by column** field.
 - e. Click **OK** to save your changes and close the window.

Restrictions:

- The specification of the find value depends on the data type of the input column in your warehouse source. The data type of the find column must match the data type of the input column.
 - Only numbers are allowed in numeric find columns.
 - Pattern strings are allowed only for character columns. Pattern strings can contain _ for any single character or % for zero or more characters. Use the rule escape character if _ or % is contained in the pattern string. For example, assume the escape character is \. To specify the character %, type \% in the find column of the rules table. For either numeric or character data, the database null value can be used as the find or replacement value.
- To define rules for Discretize:

- a. Select a rules table from the **Rules table** list.
 - b. Select a column in the rules table that contains lower bound values to search for. In the **Available columns** list, select a column, then click > next to the **Lower bound column** field.
 - c. Select a column in the rules table that contains upper bound values to search for. In the **Available columns** list, select a column, then click > next to the **Upper bound column** field.
 - d. Select a column in the rules table that contains a replacement value. In the **Available columns** list, select a column, then click > next to the **Replacement value column** field.
 - e. Optional: Select a column that contains the sequence in which to read rows in the rules table. Click a column in the **Available columns** list, then click > next to the **Order by column** field. The column that you select should be of type Integer.
 - f. Click **OK** to save your changes and close the window.

Note: You can use character string or numeric data types. The comparison is performed in the locale specific order for the character set that you are using.
- To define rules for Clip:
 - a. Select a rules table from the **Rules table** list.
 - b. Select a column from the rules table that contains the lower bound values to search for. Click a column in the **Available columns** list, then click > next to the **Lower bound column** field.
 - c. Select a column from the rules table that contains the upper bound values to search for. Click a column in the **Available columns** list, then click > next to the **Upper bound column** field.
 - d. Select a column from the rules table that contains the lower bound replacement values. Click a column in the **Available columns** list, then click > next to the **Lower replace value column** field.
 - e. Select a column from the rules table that contains the upper bound replacement values. Click a column in the **Available columns** list, then click > next to the **Upper replace value column** field.
 - f. Optional: Select a column that contains the sequence in which to read rows in the rules table. Click a column in the **Available columns** list, then click > next to the **Order by column** field. The column that you select should be of type Integer.
 - g. Click **OK** to save your changes and close the window.

Note: You can use the Clip clean type for numeric data types only. To avoid lower or upper bound clipping, specify the database null value in the rules table. The database null value can also be used as the replacement value.

Transforming data

7. In the Parameters column, click Depending on the data type of the column that you specified in the **Input column** field, either the String Parameters window or the Numeric Parameters window opens.

Use the controls in the String Parameters window to customize how the Clean Data transformer performs the find and replace tasks. To specify information about string parameters:

- a. In the Specify string settings area, select the applicable check box to:
 - Compress white space. Leading and trailing spaces are removed. Two or more consecutive interior spaces are compressed into one.
 - Ignore white space during matching. White space is removed from the input column, find, upper bound, and lower bound columns.
 - Ignore case. Case is ignored when the Clean transformer performs match processing.
 - Not copy a value if it matches a column in the rule table. A value is not copied if it matches a column in the rule table.
- b. In the Escape character field, type an escape character. For example, you can specify \. You cannot use the characters % or _. An empty field does not suggest that a blank character is an escape character.
- c. Click **OK** to save your changes and close the window.

Use the controls in the Numeric Parameters window to specify a numeric tolerance for your transformation. The default value is 0.0. A numeric tolerance is a number that you apply to numeric find values. The numeric tolerance allows the replacement of a value that is just outside a find value or a range of find values. For example, you use the Discretize clean type to replace the find value numeric ranges in the following manner: 23.50 -24.50 replaced by 1000. If you specify a tolerance of 0.10, then input data of 23.40 and 24.60 would also be replaced by 1000.

To specify a numeric tolerance:

- a. In the Numeric tolerance field, type a numeric tolerance.
 - b. If you do not want the Clean Data transformer to copy a value if it matches a column in the rule table, select the **Do not copy record to output if rule match is found** check box.
 - c. Click **OK** to save your changes and close the window.
8. Define as many clean types as you want to. When you are finished, click the **Column Mapping** tab. If you want to modify your column mappings, or create a default target table, see “Defining column mapping information” on page 150.
 9. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.

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

Generating 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 or SMALLINT 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.

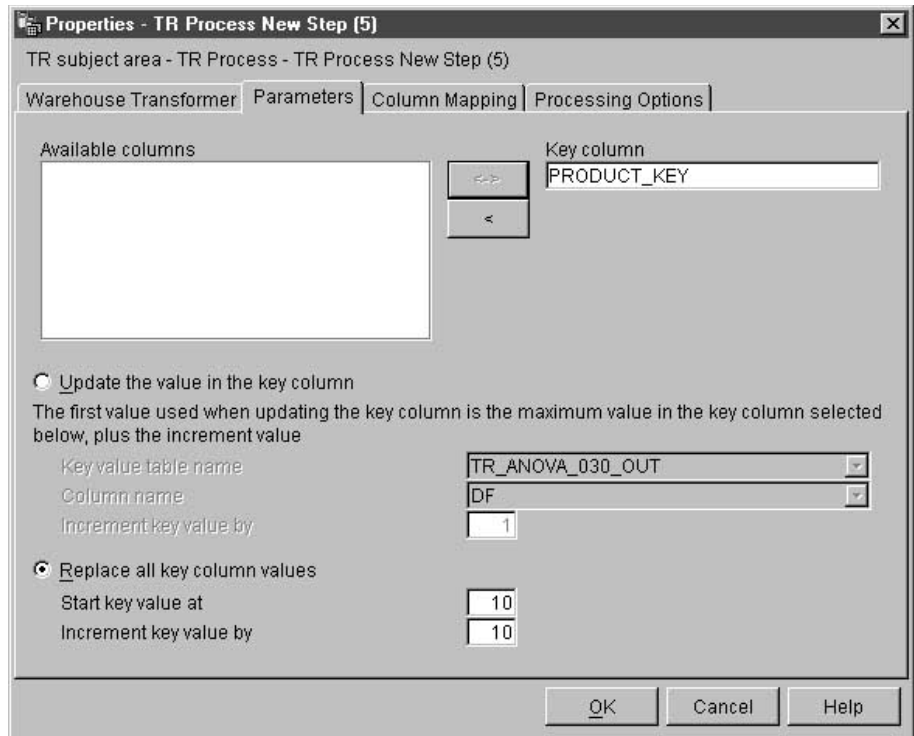
The Generate Key Table transformer uses a warehouse target table as a source. The transformer writes to a table on the warehouse target. Before you define this step, link the warehouse target to the step in the Process Model window, with the arrow pointing towards the step. You can make changes to the step only when the step is in development mode.

Column mapping is not available for this step.

To define a Generate Key table transformer:

1. Open the step notebook.
2. Specify general information about the transformer. For more information, see “Providing general information about a step” on page 149.
3. On the Parameters page, click the column from the **Available columns** list that you want to assign key values to. Then, click >. The selected column does not need to be a primary key column.

Transforming data



4. Specify whether you want to update values in the key column based on a column in the selected step or replace all key column values in the existing key column.

To update values in the key column based on a column in a selected step:

- a. Click **Update the value in the key column**.
- b. In the **Key value table name** list, select a table in the warehouse target that contains the target table. The table that you select must contain the column on which to base the generated key values.
- c. In the **Column name** list, click a column in the table that you just specified on which to base the generated key values.
- d. In the **Increment key value by** field, type a numeric increment value.

When you select Update the value in the key column, the transformer updates only those rows in the table that do not have key values. When additional rows are inserted into the table, the key values are null until you run the transformer again.

To avoid this problem, run the transformer for the first time, then follow the steps in 4a on page 229.

To replace all key column values in the existing key column:

- a. Click **Replace all key column values**.
 - b. Type a starting key value in the **Start key value at** field.
 - c. Type a numeric increment value in the **Increment key value by** field.
5. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
 6. Click **OK** to save your changes and close the step notebook.

Generating period data

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.

To define a Generate Period Table transformer:

1. Open the step notebook.
2. Specify general information about the transformer. For more information, see “Providing general information about a step” on page 149.
3. On the Parameters page, in the **Resolution** list, click the resolution for the date, time, or timestamp column for the period table.

Transforming data

Properties - TR Process - Generate period table

TR subject area - TR Process - TR Process - Generate period table

Warehouse Transformer Parameters Column Mapping Processing Options

Resolution Day

Start

Date June 17, 2001

Time 00:00:00

End

Create Rows 100

Create an end date and time

Date June 17, 2001

Time 00:00:00

Available columns

Sequence Number

Period Number

Selected columns

Julian Day

Hour of Day

Day of Week

Day of Month

Day of Year

Week of Month

Week of Year

Month of Year

Sequence number

Starting Value

Increment Value

Periodic number

Starting Value

Repeat Value

Increment Value

Cycle Value

OK Cancel Help

- Specify the starting period value by selecting either the date or the time check box, or both. Then, set your starting date or time (or both):
 - To create a period table with a column that has type Date, select the **Date** check box. Then, click A calendar is displayed. Use the << and >> buttons to scroll through the months on the calendar. After you find your start month, click a date in the calendar.
 - To create a period table with a column that has type Time, select the **Time** check box. Then, click either the hours, minutes, or seconds in **Time** field, and use the arrow buttons to specify your time.
 - To create a period table with a column that has type TimeStamp, select both check boxes, and specify a start date and a start time.
- Specify the ending period value using the number of rows, or using an ending date and time, or both:
 - To specify the total number of period rows to generate, click **Create rows**, and type the number of period rows in the **Create Row** field. Use the **Create Rows** field as an alternative to setting an end date or time for your period table. In this field, you enter the number of period rows that the transformer will generate in the Period Table. This field is optional. If you decide not to use this field, then the following conditions are true:

- If the Generate Period Table transformer is writing to a Date column, you must select an end date in the **Date** field. The end date must be greater than the start date.
- If the Generate Period Table transformer is writing to a Time column, you must select an end time in the **Time** field. The end time must be greater than the start time.
- If the Generate Period Table transformer is writing to a TimeStamp column, you must select a date in the **Date** field and a time in the **Time** field. The end date and time must be greater than the start date and time.
- To specify the ending period using a date and time or both, click **Create end date or time**, and select a date or time or both in the **Date and Time** fields.

If you do not enter a value in the **Create Rows** field, you must enter a value or values in the **Date and Time** fields in the End section.

In these fields, you either specify the date or time (or both) that you want to be last date/time value entered into the period table, based on the following conditions:

- If the Generate Period Table transformer is writing to a Date column, you must select an end date in the **Date** field. The end date must be greater than the start date.
- If the Generate Period Table transformer is writing to a Time column, you must select an end time in the **Time** field. The end time must be greater than the start time.
- If the Generate Period Table transformer is writing to a TimeStamp column, you must select an date in the **Date** field and a time in the **Time** field. The end date and time must be greater than the start date and time.

Transforming data

- Optional: In the **Available Columns** list, click one or more columns to be populated in the period table. Then, click >. The columns are moved to the **Selected columns** list. The following table describes the available columns:

Column	Data Type	Notes
Julian Day	Integer	Dates start January 1, 4712 B.C.
Hour in Day	Integer	1 - 24
Day in Week	Integer	1 - 7 (varies by locale, default locale used)
Day in Month	Integer	1 - 31
Day in Year	Integer	1 - 366
Week in Month	Integer	1 - 5 (weeks in the month begin on Sundays)
Week in Year	Integer	1 - 53
Month in Year	Integer	1 - 12
Quarter in Year	Integer	1 - 4
Year	Integer	Nnnn
Day Name	Varchar	Default locale used
Month Name	Varchar	Default locale used
Sequence Number	Integer	This column requires your input in the Sequence number section. If you select this option, the Sequence number controls become available.
Periodic Number	Integer	This column requires your input in the Periodic number section. If you select this option, the Period number controls become available.

- If you placed Sequence Number in the **Selected columns** list, type a numeric starting sequence value in the **Starting value** field. Then, type a numeric increment value in the **Increment value** field.
- If you placed Period number in the **Selected columns** list, type a numeric starting period value in the **Starting value** field. Type a numeric repeat value in the **Repeat value** field. Type a numeric increment value in the **Increment value** field. Type a numeric cycle value in the **Cycle value** field.

6. Optional: On the Column Mapping page, view or change your column mappings. If the step is not linked to a target table, you can create a default target table. For more information, see “Defining column mapping information” on page 150.
7. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
8. Click **OK** to save your changes and close the step notebook.

Inverting data

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.

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. For more information about automatic promotion of data types, see the *IBM DB2 Universal Database: SQL Reference*.

Prerequisite: 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.

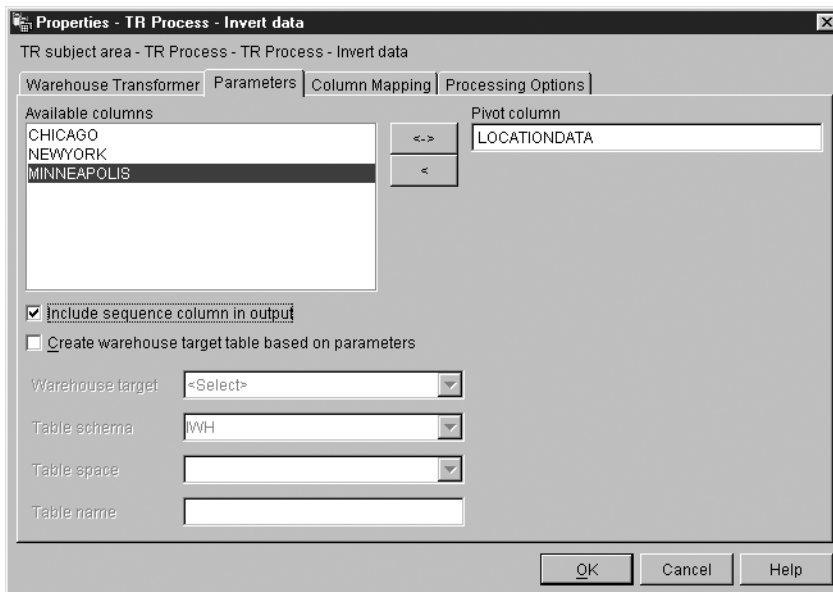
A step that uses the Invert Data transformer must be promoted to production mode before you can see the actual data that is produced.

Transforming data

This step does not use the Column Mapping page.

To define an Invert Data transformer:

1. Open the step notebook.
2. Specify general information about the transformer. For more information, see “Providing general information about a step” on page 149.
3. On the Parameters page, click the name of the column that contains target table column names in the **Available columns** list. Then, click > to move the selected column to the **Pivot column** field. The data in this column must have one or more of the following data types:
 - CHAR
 - DATE
 - TIME
 - TIMESTAMP
 - SMALLINT
 - INTEGER
 - NUMERIC
 - DECIMAL



All data in the pivot column must be fewer than 18 characters. Alphabetic characters are changed to uppercase on DB2 systems that require object names in uppercase.

If you don't select a column, the transformer assumes that the table contains raw data only, all of which it should invert. Then, the transformer creates column names for the target table based on the column number. The first column will be named 1, the second column will be named 2, and so on.

4. Optional: Select the **Include sequence column** in **output** check box to create a column in the output table that contains an ordered list of numbers. The list of numbers starts at 1 and increments by 1.
5. If your step is not connected to a target table, select **Create warehouse target table based on parameters**.
6. If you selected **Create warehouse target table based on parameters**, select a warehouse target from the **Warehouse target** list, and provide information about the target table you want to create. For more information about the fields used to create a warehouse target table, see "Creating a default target table" on page 151.
7. On the Processing Options page, provide information about how your step processes. For more information, see "Defining processing options" on page 151.
8. Click **OK** to save your changes and close the step notebook.

Pivoting data

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.

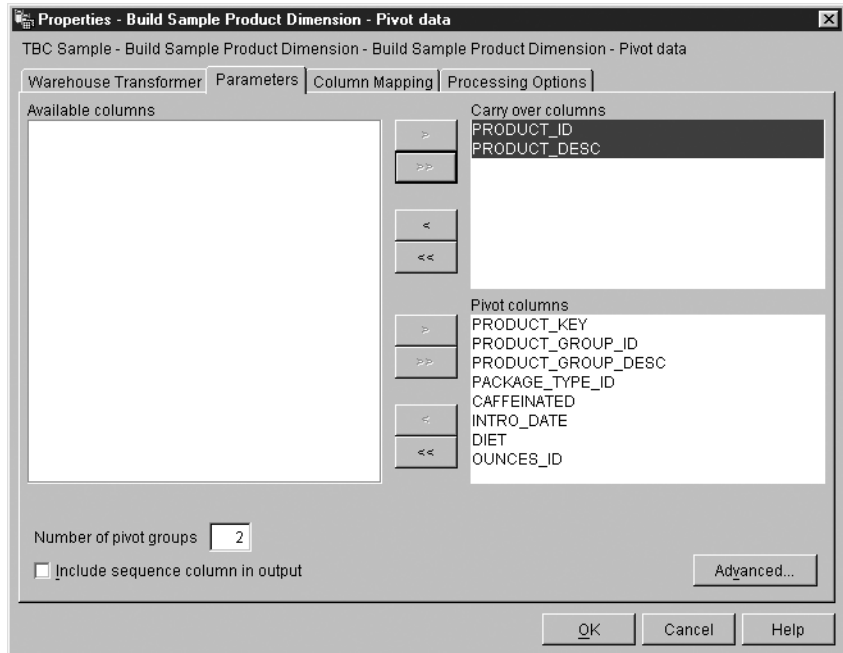
Columnar data in each pivot group must have be the same data type or data types that are related to each other through automatic promotion. For more information about automatic promotion of data types, see the *IBM DB2 Universal Database: SQL Reference*.

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.

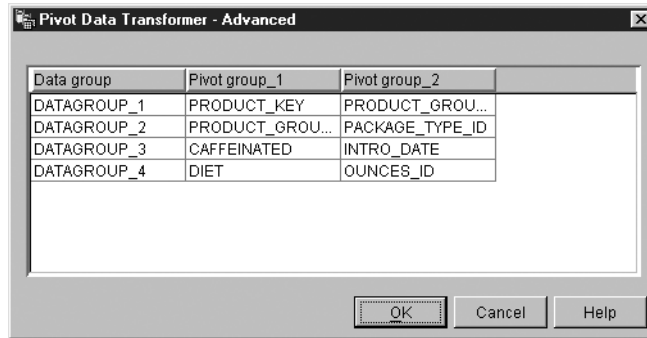
Transforming data

To define a Pivot data transformer:

1. Open the step notebook.
2. Specify general information about the transformer. For more information, see “Providing general information about a step” on page 149.
3. Click the **Parameters** tab.



4. Optional: In the **Available columns** list, click the column that you want to carry over to the target table without making any changes. Then, click > next to the **Carry-over columns** list. To select multiple columns, press Ctrl while you click columns.
5. In the **Available columns** list, click the columns that you want to use as pivot columns. Then, click > next to the **Pivot columns** list.
6. In the **Number of pivot groups** field, type the number of pivot groups that you want to create. The number of groups that you type must be a divisor of the number of pivot columns that you selected. For example, if you selected 6 pivot columns, you can type the number 1, 2, 3, or 6. If you selected 5 pivot columns, you can type 1 or 5 in this field.
7. If you selected pivot columns, specify the data group and pivot groups for each pivot column:
 - a. On the Parameters page of the Pivot Data transformer notebook, click **Advanced**. The Pivot Data Transformer - Advanced window opens.



Use the table in this window to define your pivot groups. The number of rows that are displayed in this table depend on the number of pivot columns that you selected and the number of pivot groups that you specified on the Parameters page. For example, If you selected 6 pivot columns and 2 pivot groups, 3 rows are displayed.

- b. Click the **Data group** field. Then, type a name for a data group. A data group is a column in the target table. This column is used to describe or label information that is collected into pivot group columns so that the original meaning of the data as expressed by the column labels in the input table is preserved in the output table. If you leave this field blank, and there is only one pivot group, the names of the pivot columns are pivoted into the data fields of the output column that results when the step is run.
- c. Select a pivot column for each pivot group. The number of pivot groups that are displayed in this table is the same number of pivot groups that you specified on the Parameters page. To select a pivot column, double-click the **Pivot group** field and select a column. You cannot select a column for more than one pivot group. You must select a column for each field in a pivot group.
- d. Click **OK**. The Pivot Data Transformer - Advanced window closes.

If you make changes to the selected pivot columns in the Pivot Data Transformer window, the data group and pivot group values that you specified in this window will be reset to blank.

8. If you want to include an ordered column of numbers in the output table, select the **Include sequence column in output** check box.
9. Optional: On the Column Mapping page, view or modify your column mappings, or create a default target table. For more information, see “Defining column mapping information” on page 150.
10. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.

Transforming data

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

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.

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.

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:
 - a. Select a category for the input column data from the **Category** list.
 - b. 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.
9. Specify the output format in the **Output format** area of the window:

- a. Select a category for the output column data from the **Category** list.
 - b. 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.
10. Click **OK**.

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

Transforming data

Chapter 8. Calculating statistics

Use statistical transformers to perform the following statistical functions:

- Analysis of variance (ANOVA)
- Calculate basic statistics
- Calculate subtotals
- Chi-square test
- Chi-square goodness-of-fit test
- Correlation analysis
- Calculate moving averages
- Regression

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.

For more information about the ANOVA types, see the online help.

The ANOVA transformer obtains two independent estimates of variance. The first estimate is based on variability between groups. The second estimate is based on variability within groups. After the ANOVA transformer computes these estimates, it calculates their ratio. A family of distributions, the Fisher-F distributions, describes the significance of this ratio.

This transformer also calculates a p-value. The p-value is the probability that the means of the two groups are equal. A small p-value leads to the conclusion that the means are different. For example, a p-value of 0.02 means that there is a 2% chance that the sample means are equal. Likewise, a large p-value leads to the conclusion that the means of the two groups are not different.

You can use this step only with tables that exist in the same database. Use a warehouse source or target table as a source for the ANOVA transformer and up to two warehouse target tables as targets for the ANOVA statistical calculations. If you do not want to select a target table for the ANOVA transformation, you can 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.

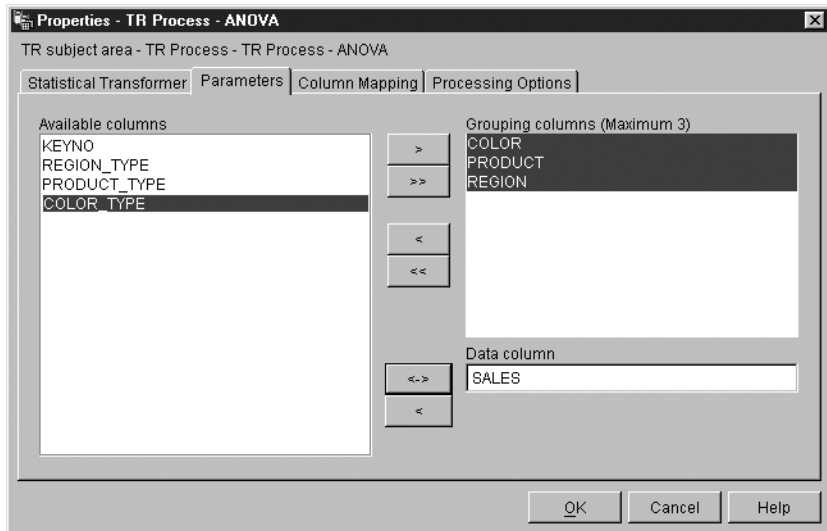
Calculating statistics

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.

To define an ANOVA transformer:

1. Open the step notebook for the ANOVA transformer.
2. Specify general information about the transformer. For more information, see “Providing general information about a step” on page 149.
3. On the Parameters page, select the column or columns from the **Available Columns** field that will be the Grouping columns for the ANOVA transformer. Depending on the type of ANOVA transformer that you want to create, select one, two, or three columns, and click > next to the **Grouping columns** field.
4. From the **Available Columns** field, click a column name, and click > next to the **Data column** field. This column cannot be used as a grouping column. This column must contain numeric data.



5. Select target tables for your transformer:
 - If your ANOVA transformer is using two target tables and is already linked to the target tables, select one of the tables to contain ANOVA statistics information. From the **ANOVA statistics table** list, select a statistics target table for the ANOVA transformer.
6. Optional: On the Column Mapping page, you can view the mappings between the output columns that result from the transformations that you

defined on the Parameters page and the columns on your target table. You cannot change these mappings. For more information about the tasks you can perform using the Column Mapping page, see “Defining column mapping information” on page 150.

7. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
8. Click **OK** to save your changes and close the step notebook.

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

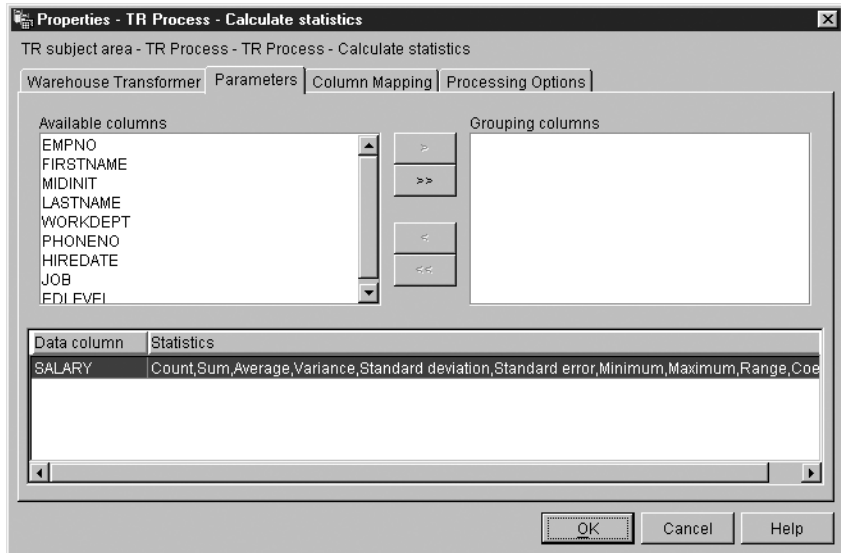
To use the Calculate Statistics transformer, connect the step to a warehouse source and a warehouse target that exist in the same database. Or, you can connect the step to a warehouse source and specify that the step create a target table for you in the same database.

You can make changes to the step only when the step is in development mode.

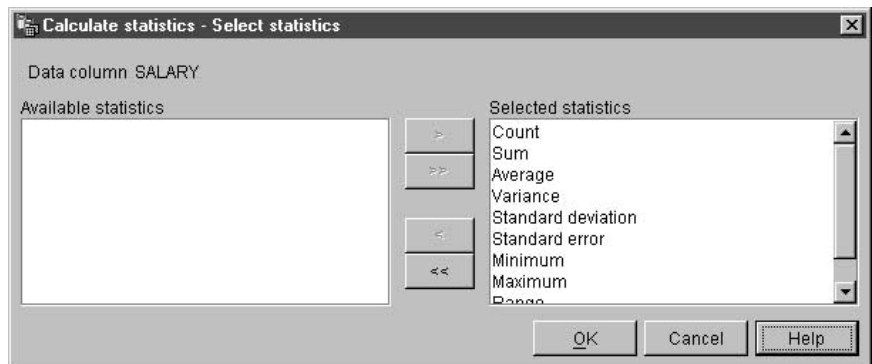
To define a Calculate Statistics transformer step:

1. Open the step notebook for the Calculate statistics transformer.
2. Specify general information about the transformer. For more information, see “Providing general information about a step” on page 149.
3. Click the **Parameters tab**.

Calculating statistics



4. Optional: From the **Available columns** list, select any columns that you want to use as grouping columns, and click >. Grouping columns can contain character or numeric data.
5. Define statistics calculations:
 - a. Right-click anywhere on the Transformer definition table, and click **Add**. A row is added to the table.
 - b. Under the **Data column** heading in the row that you just created, click and select a column for which you want to calculate statistics.
 - c. Double-click the **Statistics** field. The ... push button is available.
 - d. Click the ... push button. The Calculate Statistics - Select Statistics window opens.



- e. From the **Available statistics list**, click one or more statistics to run on the selected column. Then, click >. The type of statistics that you can

select from will depend on the data type of the input column. For character data types, only the Count function is available.

- f. Repeat steps 5a-e, as appropriate.
 - g. Click **OK**. The Elementary Statistics - Select Statistics window closes.
After you define a statistics calculation, it cannot be changed. You must delete unwanted definitions and redefine them. The Calculate Statistics transformer supports partial data. For example, if you select a column to define statistics for, but you don't select the statistics for it, the Calculate Statistics transformer will save your column selection. However, you cannot map columns for a row that has a partial data selection, nor can you successfully run a step that has a partial data selection.
6. On the Column Mapping page, map the output columns that result from your statistical calculations to columns in your target table. The column names for your statistical calculations are based on the data column that you select on the Parameters page and the statistic that you select for it. A column is created for each statistic that is selected for a data column. For example, if your data column, Sales, has the statistics "Sum" and "Average" defined to it, the columns Sales_sum and Sales_average will be displayed on the Column Mapping page. For more information about the tasks you can perform using the Column Mapping page, see "Defining column mapping information" on page 150.
If the Parameters page produces no output columns, or if this step is not linked to a target table and you did not create a target table using the Column Mapping page, you cannot use this page to map your columns. Some steps will not allow you to change the column mapping.
 7. On the Processing Options page, provide information about how your step processes. For more information, see "Defining processing options" on page 151.
 8. Click **OK** to save your changes and close the step notebook.

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.

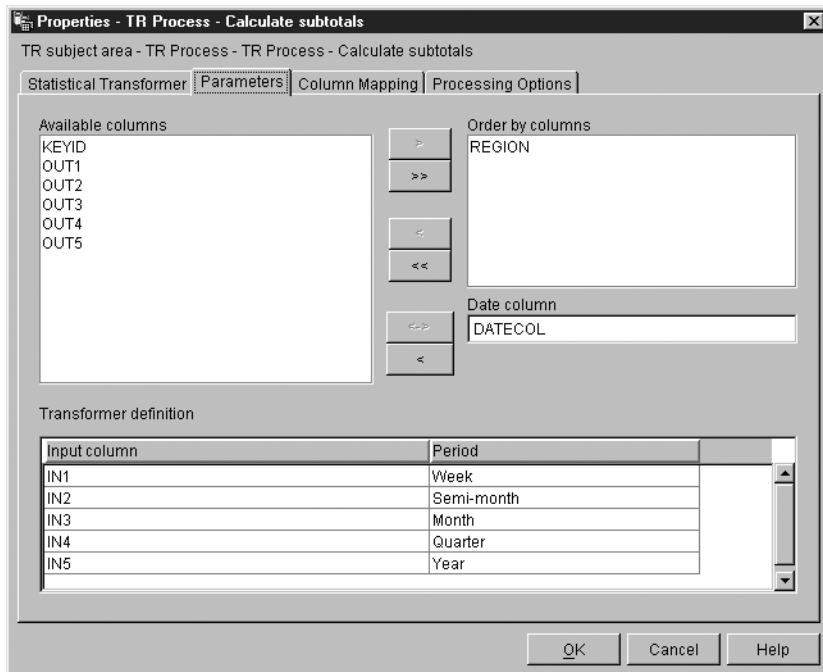
The Calculate Subtotals transformer uses a warehouse target table as a source. The table that you use as a source must contain a primary key. If you use a target table that was generated by the Data Warehouse Center, you must

Calculating statistics

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.

To define a Calculate Subtotals transformer:

1. Open the step notebook for the Calculate subtotals transformer.
2. Specify general information about the transformer. For more information, see “Providing general information about a step” on page 149.
3. Click the **Parameters** tab.



4. Optional: Click columns from the **Available columns** list to order the output data. Then, click > next to the **Order by columns** list.
5. From the **Available columns** list, click the date column in the input table. The column data type must be DATE or TIMESTAMP. Then, click > next to the **Date column** field.
6. Define a subtotal calculation:
 - a. Right-click anywhere on the Transformer definition table, and click **Add**. A row is added to the table.

- b. Under the **Input column** heading in the row you just added, left-click and select the name of the column on which to calculate a subtotal. Only available numeric columns are listed. A numeric column is not listed if it is being written to by another row or if it is selected as a grouping column. If you use a column as an input in two or more rows, you cannot use the column as an output column.
 - c. Under the **Period column** heading, left-click and select the period for which to calculate.
 - d. Repeat steps 6a-c to define additional subtotal calculations.
7. Use the Column Mapping page to map output columns from your transformer to columns on your target table. If you did not link the transformer to an output target table, you can create a target table using the Column Mapping page. For more information about the tasks you can perform using the Column Mapping page, see “Defining column mapping information” on page 150.

Restrictions for mapping columns:

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	
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

Calculating statistics

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.

If the Parameters page produces no output columns, or if this step is not linked to a target table and you did not create a default target table using the Column Mapping page, you cannot use this page to map your columns. Some steps will not allow you to change the column mapping.

8. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
9. Click **OK** to save your changes and close the step notebook.

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.

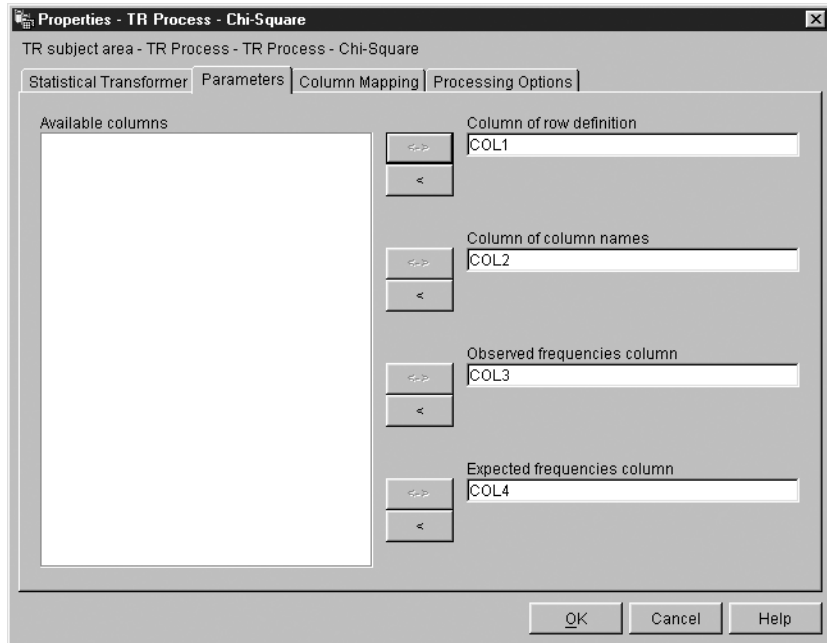
When you set up this process in the Process Model window, link the Chi-Square step to a warehouse target table. If you want the step to produce the Expected value output table, link the step to a second warehouse target table in the same database.

You can change the step definition only when the step is in development mode.

To define a Chi-square transformer:

1. Open the step notebook.
2. Specify general information about the transformer. For more information, see “Providing general information about a step” on page 149.
3. In the **Available columns list** on the Parameters page, click a column. Then, click > next to the **Column of row definition** field. This field is required for both goodness-of-fit calculation and chi-square calculations. If you want your step to run as a chi-square calculation, go to step 4. Otherwise, go to step 5. Your step will run as a goodness-of-fit calculation.

Calculating statistics



4. To define a chi-square calculation, click a column in the **Available columns list**, and click > next to the Column of column names field.
5. In the **Available columns list**, click a column that contains the observed frequencies data. This column must be of numeric type. Then, click > next to the **Observed frequencies column** field.
6. In the **Available columns list**, click a column that contains expected frequency data. This column must be of numeric type. Then, click > next to the **Expected frequencies column** field. This field is required for goodness-of-fit calculations and optional for chi-square calculations. If you specify an expected frequencies column, you cannot use the expected values output table.
7. In the **Expected values output table** list, select a target table for the expected values output table. Depending on certain conditions, this field is either optional or required:
 - This field is optional if only one target table is linked to the chi-square step in the Process Model window. If you want to create an expected values output table, select your target table. Then, click **OK** to save and close the step. Next, in the Process Model window, link a second table to the chi-square step to contain your regular chi-square output. Finally, open the chi-square step and continue defining values for the transformer.
 - This field is optional for chi-square calculations.

- This field is required if two tables are linked to the chi-square step in the Process Model window. Select one of the tables to be the expected values output table.
8. Optional: On the Column Mapping page, you can view the mappings between the output columns that result from the transformations that you defined on the Parameters page and the columns on your target table. You cannot change these mappings. If you do not have a target table linked to the step, you can create one. For more information about tasks you can perform using the Column Mapping page, see “Defining column mapping information” on page 150.
 9. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
 10. Click **OK** to save your changes and close the step notebook.

Correlation transformer

Use the Correlation transformer to determine the extent to which changes in the value of an attribute (such as length of employment) are associated with changes in another attribute (such as salary). The data for a correlation analysis consists of two input columns. Each column contains values for one of the attributes of interest. The Correlation transformer can calculate various measures of association between the two input columns. You can select more than one statistic to calculate for a given pair of input columns.

The data in the input columns also can be treated as a sample obtained from a larger population, and the Correlation transformer can be used to test whether the attributes are correlated in the population. In this context, the *null hypothesis* asserts that the two attributes are not correlated, and the *alternative hypothesis* asserts that the attributes are correlated.

The Correlation transformer calculates any of the following correlation-related statistics on one or more pairs of columns:

Correlation coefficient r

The correlation coefficient r is a measure of the linear relationship between two attributes or columns of data. The correlation coefficient is also known as the Pearson product-moment correlation coefficient. The value of r can range from -1 to +1 and is independent of the units of measurement. A value of r near 0 indicates little correlation between attributes; a value near +1 or -1 indicates a high level of correlation.

When two attributes have a positive correlation coefficient, an increase in the value of one attribute indicates a likely increase in the value of the second attribute. A correlation coefficient of less than 0 indicates a

Calculating statistics

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

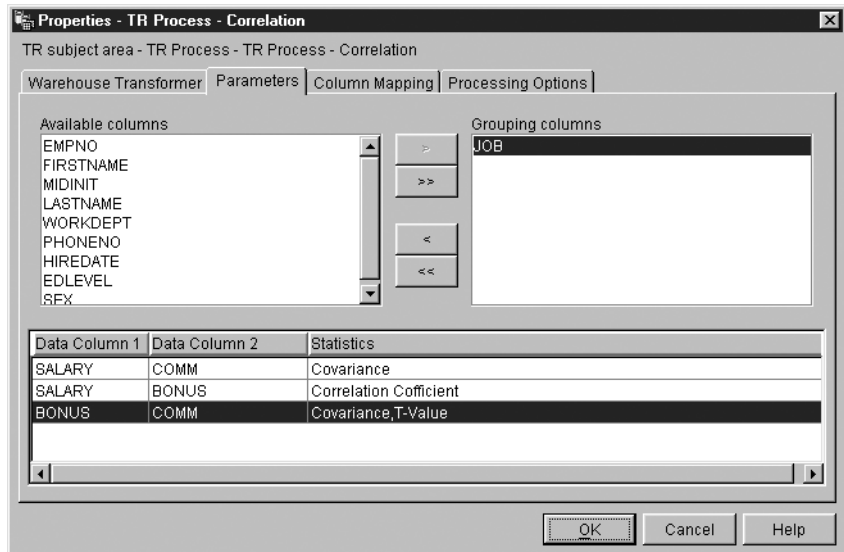
P-value is the probability, when the null hypothesis is true, that the absolute value of the T-statistic would equal or exceed the observed

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.

To define a Correlation transformer step:

1. Open the step notebook.
2. Specify general information about the transformer. For more information, see “Providing general information about a step” on page 149.
3. Click the **Parameters** tab.



4. Optional: Click columns that you want to use as grouping columns and click >. Grouping columns can contain character or numeric data.
5. Define correlation statistics:
 - a. Right-click anywhere on the Transformer definition table, and click **Add**. A row is added to the table.
 - b. Under the **Data Column 1** heading in the row that you just added, click and select the first column for the calculation. Only columns of numeric type are listed.
 - c. Under the **Data Column 2** heading, click and select the second column for the calculation. Only columns of numeric type are listed. You cannot use the same column in Data Column 2 that you use in Data Column 1.

Calculating statistics

- d. Double-click under the **Statistics** heading in the row you just created. The ... push button is available.
- e. Click the ... push button. The Correlation - Select Statistics window opens.
- f. Click one or more statistics from the **Available statistics** list. Then, click >. The statistics are moved to the **Selected statistics** list.
- g. Repeat steps 5a-f, as appropriate.
- h. Click **OK**. The Correlation - Select Statistics window closes.

The Correlation transformer supports partial data. For example, if you select a column to define statistics for, but don't select the statistics for it, the Correlation transformer will save your column selection.

However, you cannot map columns for a row that has a partial data selection, nor can you successfully run a step that has a partial data selection.

6. On the Column Mapping page, map the columns that result from your correlation statistics to columns in your target table. If you did not link the transformer to an output target table, you can create a target table.

The column names for your correlation statistics are based on the data column entries that you select on the Parameters page and the statistic that you select for it. A column is created for each statistic that is selected and its corresponding data columns. For example, if your data columns, Salary and Employment, have the correlation statistics Covariance and T-value defined to them, the columns Covariance_Salary_Employment and T-value_Salary_Employment will be displayed on the Column Mapping page. For more information about tasks you can perform using the Column Mapping page, see "Defining column mapping information" on page 150.

If the target column is mapped to a source column, the mapping remains intact. If the Parameters page produces no output columns, or if this step is not linked to a target table and you did not create a default target table using the Column Mapping page, you cannot use this page to map your columns. Some steps will not allow you to change the column mapping.

7. On the Processing Options page, provide information about how your step processes. For more information, see "Defining processing options" on page 151.
8. Click **OK** to save your changes and close the step notebook.

Moving Average transformer

Simple and exponentially smoothed moving averages can often predict the future course of a time-related series of values. Moving averages are widely used in time-series analysis in business and financial forecasting. Rolling sums have other widely used financial uses.

You can use the Moving Average transformer to calculate the following values:

- A simple moving average
- An exponential moving average
- A rolling sum for N periods of data, where N is specified by the user

An exponential moving average is also known as an exponentially smoothed moving average.

Moving averages redistribute events that occur briefly over a wider period of time. This redistribution serves to remove noise, random occurrences, large peaks, and valleys from time-series data. You can apply the moving average method to a time-series data set to:

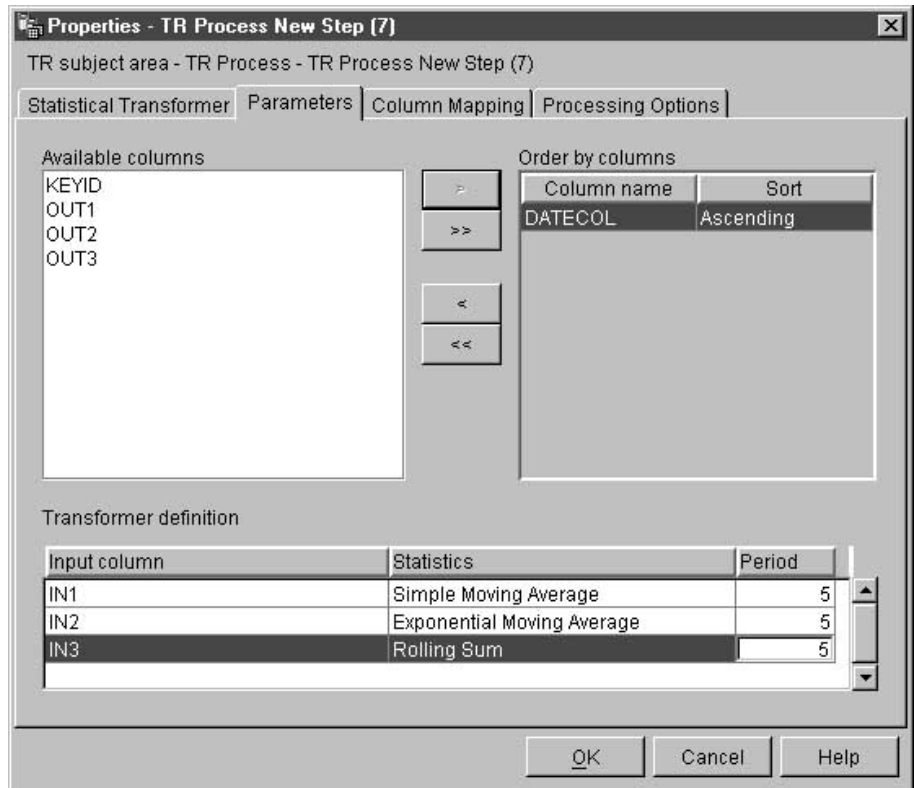
- Remove the effects of seasonal variations.
- Extract the data trend.
- Enhance the long-term cycles.
- Smooth a data set before performing higher-level analysis.

The Moving Average transformer uses a warehouse target table as a source. The table that you use as a source must contain a primary key. If you use a target table that was generated by the Data Warehouse Center, you must assign a primary key to the table before you use it as a source. The transformer writes to a table on the warehouse target. Before you define this step, link the warehouse target to the step in the Process Model window, with the arrow pointing towards the step.

To define a Moving Average transformer:

1. Open the step notebook.
2. Specify general information about the transformer. For more information, see “Providing general information about a step” on page 149.
3. On the Parameters page, in the **Available Columns** field, click columns that you want to use as grouping columns, then click >. The columns are moved to the **Order by columns** field. Grouping columns can contain character or numeric data.

Calculating statistics



4. For each column listed in the **Order by columns** field, click a sort order from the **Sort** list. Select **Ascending** for ascending order. Select **Descending** for descending order.
5. Define moving average calculations:
 - a. Right-click anywhere on the Transformer definition table, and click **Add**. A row is added to the table.
 - b. Under the **Input column** heading in the row that you just added, left-click and select a column for which to calculate a moving average. Only numeric columns are listed. A numeric column is not listed if it is being written to by another row, or if it is selected as a grouping column. If you use a column as an input in 2 or more rows, you cannot use the column as an output column.
 - c. Under the **Statistics column** heading, left-click and select the type of moving average to calculate.

In the Moving Average transformer, you can select one of three statistics:

- A simple moving average. A simple moving average (SMA) is defined as:

$SMA [row\ i] = \text{SUM} (\text{data values for last } N \text{ rows}) / N$

- An exponential moving average. An exponential moving average (EMA) is defined as:

$EMA [row\ i] = (\text{Value } [row\ i] * K) + (\text{EMA } [row\ i-1] * (1-K))$
 where: $K = 2/(N+1)$

For example: For a four-day EMA, $N = 4$, $K = 2/5$ and $1-K = 3/5$. An exponential moving average is also known as an exponentially smoothed moving average.

- A rolling sum for N periods of data, where N is specified by the user. A rolling sum (RS) is defined as:

$RS [row\ i] = \text{SUM} (\text{data values for the last } N \text{ rows})$

- d. Under the **Period column** heading, double-click the field, then type the number of days for which the moving average will be calculated. For example, if you want to calculate the moving average for one week, type 7. You can only type digits. Other keystrokes are ignored.
 - e. Repeat steps 5a-d, as appropriate.
6. On the Column Mapping page, map output columns from your transformer to columns on your target table. If you did not link the transformer to a target table, you can create a target table. For more information, see “Defining column mapping information” on page 150.

Columns that result from your transformation are listed on the left side of the page. These columns are named after the selections you made for the Input column, the Statistics column, and the Period column on the Parameters page. For example, if you selected SALARY as the input column, EMA as the calculation, and 7 for the period on the Parameters page, you will see SALARY_EMA_7 displayed on the left side of the Column Mapping page .

Restrictions:

These 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 moving average 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

Calculating statistics

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_EMA_7	
A_EMA_31	

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.

If the Parameters page produces no output columns, or if this step is not linked to a target table and you did not create a default target table using the Column Mapping page, you cannot use this page to map your columns. Some steps will not allow you to change the column mapping.

7. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
8. Click **OK** to save your changes and close the step notebook.

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.

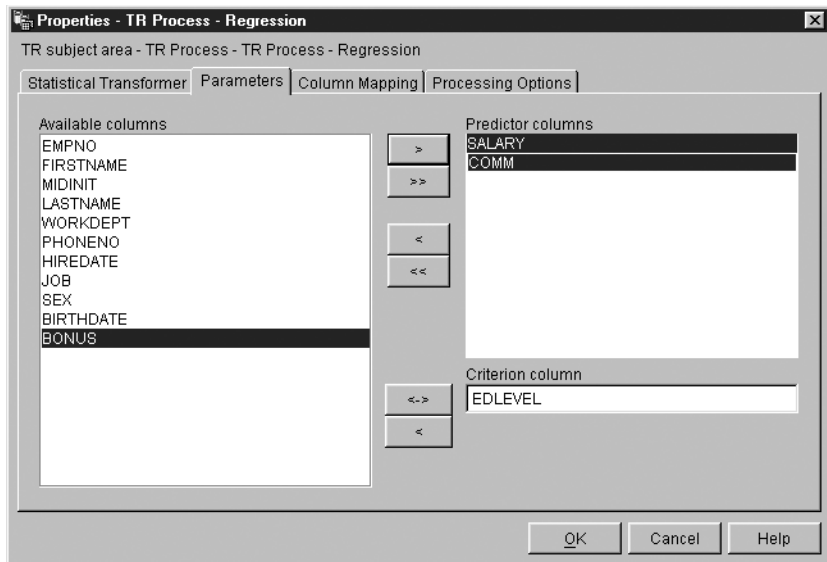
The Regression transformer produces two additional output tables: the ANOVA summary table and the Equation variable table.

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.

To define a Regression transformer:

1. Open the step notebook.
2. Specify general information about the transformer. For more information, see “Providing general information about a step” on page 149.
3. On the Parameters page, select columns from the **Available columns** list that contain independent variable data used for prediction in the regression calculations. Then, click > next to the **Predictor columns** list. The columns are added to the **Predictor columns** list. Only columns of numeric data type are available.

Calculating statistics



4. From the **Available columns** list, click one column that contains the dependent variable data to use for criteria in the regression calculations. Then, click > next to the **Criterion column** field. Only columns of numeric data type are available.
5. In the **Summary table** list, select a target table to be your ANOVA summary table.
6. In the **Equation variable table** list, select a target table to be your Equation variable table.
7. Optional: On the Column Mapping page, you can view the mappings between the output columns that result from the transformations that you defined on the Parameters page and the columns on your target table. You cannot change these mappings. If you have not linked the transformer to a target table, you can create one. For more information, see "Defining column mapping information" on page 150
8. On the Processing Options page, provide information about how your step processes. For more information, see "Defining processing options" on page 151.
9. Click **OK** to save your changes and close the step notebook.

Chapter 9. Cleansing name and address data

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.

Requirements

- 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.

Table 22 shows the software that is required to do name and address cleansing in the Data Warehouse Center.

Table 22. Software requirements for name and address cleansing

Operating system	Required software
UNIX	Trillium Software System Version 4.0 DB2 Universal Database Warehouse Manager Version 7.2 warehouse agent
Windows NT and Windows 2000	Trillium Software System Version 4.0 DB2 Universal Database Warehouse Manager Version 7.2 warehouse agent For remote access, the host must have ftpd and rexecd daemons.

Cleansing name and address data

Table 22. Software requirements for name and address cleansing (continued)

OS/390	Trillium Software System Version 4.0 installed on the remote OS/390 host
	DB2 Universal Database Warehouse Manager Version 7.2 warehouse agent installed on UNIX, Windows NT
	TCP/IP 3.2 or later must be installed
	The OS/390 operating system is supported as a remote host only

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.

Using the Trillium Batch System with the Data Warehouse Center

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.

Figure 14 through Figure 17 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 14. Sample Trillium script file

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

Figure 15. 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 16. 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 17. The Trillium Batch System step definition

Importing Trillium metadata

To import Trillium metadata into the Data Warehouse Center:

1. Create a Trillium Batch System script or JCL. You can use any script or JCL writing tool to create the script or JCL file.
2. Right-click **Warehouse**, and click **Import Metadata** —> **Trillium** to open the Trillium Batch System window.
3. In the **Script or JCL** field, type the name of the Trillium Batch System script or JCL file that you want to run.
4. In the **Input file** field, type the name of the input data file for the Trillium Batch System program that runs first in the specified script or JCL file.

Cleansing name and address data

5. In the **Input DDL** field, type the name of the input DDL file that describes the input data file. This file must be available on the warehouse agent site.
6. In the **Output file** field, type the name of the output data file for the last Trillium Batch System program in the script or JCL file.
7. In the **Output DDL** field, type the name of the output DDL file that describes the output data file. This file must be available on the warehouse agent site.
8. Optional: In the **Output error file** field, type the name of the output error file that you want to use. This error file captures the run-time errors from the Trillium Batch System program. These errors are recorded in the stderr log. For local hosts, a default output error file is created if you do not specify a name here. For more information about the output error file, see “Error handling” on page 268.
9. Click the **Connection** tab.
10. If the Trillium metadata that you are importing is on the warehouse agent site, click **Local host**.

If the Trillium metadata that you are importing is not on the warehouse agent site:

- a. Click **Remote host**, and type the TCP/IP host name of the remote system that contains the metadata that you are importing. If you select **Remote host**, the target file is created as a local file, because remote target files are not supported. You can add an FTP step to get the remote file to the specified local target file.
- b. In the **Remote operating system** list, click the operating system of the remote host that you are accessing.
- c. In the **Remote user ID** field, type the user ID for the remote host that you are accessing.
- d. In the **Password option** list, select the password option that you want to use for the remote host that you are accessing:

Password not required

Specifies that no password is required to access the metadata on the remote host.

Retrieve password

Specifies that the password will be retrieved from a user-defined program.

In the **Password program** field, type the name of the password program that will retrieve the password. The program must reside on the warehouse agent site and write the password to an output file in the first line.

In the **Program parameters** field, type the parameters for the password program. The first parameter must be the output file to which the password is written.

Enter password later

Specifies that the password will be entered later.

Enter the password in the Properties notebook for the step that runs the Trillium Batch System program.

- Click **OK** to import the Trillium metadata and close the notebook. If the script or JCL does not run from the default agent site, specify the warehouse agent site that you are using in the Properties notebook for the Trillium Batch System step.

The following warehouse objects are added to the Warehouse tree when the import operation is complete.

- Trillium Batch System.*scriptName* template, where *scriptName* is the name of the script or JCL file.
- Trillium Batch System process.
- Trillium Batch System step that runs the user-defined program.
- The warehouse file source and warehouse file target that you specified when you imported the metadata. The file source and file target are fixed files.
- Trillium Batch System program group.

Mapping the metadata

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 Table 23.

Table 23. Mapping DDL to Data Warehouse Center data types

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 OS/390 operating system.

Cleansing name and address data

The variable *n* is the number of characters in the string.

Restrictions

You can specify overlapping fields in the input and output DDL files with the Trillium DDL and the import metadata operation in the Data Warehouse Center. However, the corresponding warehouse source and warehouse target files cannot be used in the Data Warehouse Center with the 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.

Writing a Trillium Batch System JCL file

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 `SYSTEM 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. For more information about error handling and reporting, see “Error handling” on page 268.

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.

```
//SYSTEM DD UNIT=&UNIT,  
//          DISP=(MOD,CATLG,KEEP),  
//          SPACE=(400,(20,20),,ROUND),  
//          DSN=&PROJPREF.&TRILVER.&PROJECT.STDERR;
```

Figure 18. Example of a job step that includes a SYSTEM DD statement

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.

Defining a Trillium Batch System step

You must import the Trillium metadata that you want to use in the process before you define a Trillium Batch System step. To add a Trillium Batch System step to a process:

1. Open the process in the process modeler.
2. Click the **Trillium Batch System** icon on the palette.
3. Click **Trillium Batch System program** → *programName*, where *programName* is the name of the Trillium Batch System program that you want to use.
4. Click the place on the canvas where you want the step to appear.
5. Complete the steps in the topic, "Defining a step that runs a user-defined program," in the online help.

Using the Trillium Batch System user-defined program

The Trillium Batch System user-defined program is included with DB2 Data Warehouse Center Version 7.2 for Windows NT 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. Table 24 contains the parameters for the Trillium Batch System script or JCL.

Table 24. 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 OS/390 operating system • UNIX for AIX, Solaris Operating Environment, HP-UX, and NUMA/Q operating systems • WIN for the Windows NT and 2000 operating systems

Cleansing name and address data

Table 24. Parameters for the Trillium Batch System script or JCL (continued)

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.
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	The valid value is the password or password program name. The password program must be local to the warehouse agent site.
Program parameters	The parameters for the password program.
Output Error File	The name of the output error file.
Note: The data type for all of the parameters in this table is CHARACTER.	

Error handling

The Trillium Batch System programs write error messages to the standard error (stderr) file on the Windows NT and UNIX operating systems, and to the SYSTERM data set on the OS/390 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 OS/390 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

Table 25. Error numbers and descriptions

Error number	Description
0	Success
4	Warning. Either the password file could not be erased or an internal error occurred while the Trillium Batch System user-defined program was accessing a temporary file. Check the status of the password file, or all of the temporary files created under the directory that is specified by the environment variable VWS_LOGGING.
8	The amount or values of the parameters are not correct. Read the log file or the documentation for the correct syntax.
12	A problem occurred while the Trillium Batch System user-defined program was connecting to the remote host through FTP. Check the FTP connection or host name, user ID, and password.
16	The Trillium Batch System user-defined program cannot create the log or an internal file. Check to see if the user has the correct authorization. Check to see if the disk is full.

Cleansing name and address data

Table 25. Error numbers and descriptions (continued)

20	Either the OS/390 JCL cannot be executed, or an error occurred while the Trillium Batch System user-defined program was deleting or getting a file from OS/390 through FTP. Check the JESLogFile to identify the reason.
48	Environment variable VWS_LOGGING cannot be found or the log file cannot be created. Check the log file for more information.
56	Either the Windows NT or UNIX script cannot be 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.
500	The script or JCL file returns an error, or it does not return an error but the error file contains data. Check the log file for more information. On OS/390, also check the JESLogFile.

Log file

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 10. Data modeling

This chapter describes 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 Manager (formerly DataGuide) tag language file.

What is the 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 Manager tag language file. The logical model for the Information Catalog Manager, 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. For more information about mapping of ER1 attributes to Data Warehouse Center or Information Catalog Manager tags, see “ERwin to DB2 Data Warehouse Center mapping” on page 277 and “ERwin to Information Catalog Manager mapping” on page 278.

The metadata extract program supports all ER1 models with relational databases, including DB2, Informix, Oracle, Sybase, ODBC data sources and Microsoft SQL Server.

Software requirements

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

The following software is required to import the ERwin tag language file:

For Data Warehouse Center:

IBM DB2 Universal Database Version 7.2

For Information Catalog Manager:

IBM DB2 Universal Database Warehouse Manager 7.2

The template tag language files (.tag) must be in the directory that the VWS_TEMPLATES environment variable points to.

Data modeling

The type tag language files (.typ) must be in the directory that the DGWPATH environment variable.

Program files

The metadata extract program is installed in the sqllib\bin subdirectory of the IBM DB2 directory. The program installs the following files in your directory:

flgerwin.exe

Main migration program

erwext.dll

Tag language file generator DLL

cdmerwsn.dll

ERwin API wrapper class DLL

To start the extract program, issue the flgerwin command from a command prompt.

Creating tag language files

To create a Data Warehouse Center or Information Catalog Manager tag language file, run the flgerwin.exe program and provide two main parameters. The first parameter is either -dwc if you are creating a Data Warehouse Center tag language file, or -icm if you are creating an Information Catalog Manager tag language file. The first parameter is the name of the ER1 file from which to extract metadata. The second parameter is the name of the output tag language file. For a Data Warehouse Center tag language file, the second parameter can be -starschema or -m. For an Information Catalog Manager tag language file, the second parameter can be -m, -u, -a, or -d. By default, the extract program adds the MERGE parameter to the Data Warehouse Center tag language file.

The command syntax is:

```
flgerwin inputFile.er1 outputFile.tag [-dwc] [-icm] [ -m] [-u] [-a] [-d]
```

The syntax for the command if you want to create a star schema is:

```
flgerwin inputFile.er1 outputFile.tag [-dwc] [-starschema]
```

-dwc The first parameter. Creates a Data Warehouse Center tag language file. Optional second parameters are -m and -starschema.

-icm The first parameter. Creates an Information Catalog Manager tag language file. Optional second parameters are -m, -u, -a, and -d.

-starschema

Creates an ERwin model star schema tag language file.

- m** Specifies the action on the object as MERGE.
- u** Specifies the action on the object as UPDATE.
- a** Specifies the action on the object as ADD.
- d** Specifies the action on the object as DELETE.

The metadata extract program works with metadata, not data. After you complete the ERwin tag language file import and before you use the target table, you need to match passwords and user IDs.

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.

With the metadata extract program, you can import a tag language file as a target. In newly imported metadata, the tables are not yet populated. You can view these tables as logical or physical representations and then create a warehouse step to populate the table definitions imported from ERwin.

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.

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.

When you are creating star schemas by autojoining dimension tables to fact tables, processing can take a long time depending on how many tables you use. During processing, the autojoin lines are green. When the schema is saved, the lines change to black.

Use the automatically generated constraint name to ensure that the constraint name is unique.

Data modeling

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.

Importing a tag language file into the Data Warehouse Center

You can import a tag language file into the Data Warehouse Center in either of two ways. You can use the Data Warehouse Center or the command line.

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

1. Click **Start** → **Programs** → **IBM DB2** → **Control Center**. The DB2 Control Center opens.
2. Open the Data Warehouse Center and log on.
3. Right-click on **Warehouse**. The Import window opens.
4. Click **Import Metadata** → **ERwin**. The **Import Metadata** window opens.
5. In the **Input file** field, type the name of the input tag language file, and click **OK**.
6. Select the **Extract star schema** check box to define an ERwin star schema metadata model as a warehouse schema.

After the import is completed, you can click on **View** → **Refresh** to view the new step.

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

```
iwh2imp2 tag-filename log-pathname  
target-control-db userid password
```

tag-filename

The full path and file name of the tag language file.

log-pathname

The full path name of the log file.

target-control-db

The name of the target database for the import.

userid The user ID used to access the control database.

password

The password used to access the control database.

To change a DB2 database definition so that it is a source in the Data Warehouse Center, you can 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.

When the 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.

Importing a tag language file into the Information Catalog Manager

There are two ways to import a tag language file into the Information Catalog Manager. You can use the Information Catalog Administrator or use the command line.

To use the Information Catalog Administrator to import a tag language file:

1. Click **Start** → **Programs** → **DB2** → **Information Catalog Manager**.
2. Click **Catalog** → **Import**. The Import window opens.
3. Click **Find** to search for the tag language file, then click **Import**.

After the import is completed, you can double-click on the **Subjects** icon, which opens up a window that shows all the imported models and databases.

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

```
DGUIDE /USERID userid /PASSWORD password
  /DGNAME dgname /IMPORT filename /LOGFILE
filename /ADMIN /RESTART (B|C)
```

/USERID

The user ID used to access the control database.

/PASSWORD

The password for this user ID.

/DGNAME

The Information Catalog name.

Data modeling

/IMPORT

The full path and file name of the tag language file.

/LOGFILE

The full path name of the log file.

/ADMIN Specifies that the user has administrator authority.

/RESTART

Specifies that the import will start at the beginning of the tag language file (choice B) or start from the last committed point (choice C, the default).

Troubleshooting

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. Check that the Data Warehouse Center is installed.

Path to type files not found.

The environment variable DGWPATH is not set. Check 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. For more information, see “Software requirements” on page 271. Save the ER1 file.

Unknown ERwAPI error.

An ERwin API error occurred, and the program could not obtain more information on the error. Make sure that ERwin 3.5.2 is installed. You must register the ERwin API.

To register the ERwin API, run the following command from the directory where your ERwin program files are installed: **regsvr32 er2api32.dll**. You 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.

ERwin to DB2 Data Warehouse Center mapping

Table 26 through Table 28 on page 278 show how the main ERwin object attributes correspond to the Data Warehouse Center tags:

Table 26. Database - WarehouseDatabase.tag or SourceDatabase.tag

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

Table 27. Table - Table.tag

ERwin	Command line tag	Data Warehouse Center
Table Name	NAME	Table name
Table Name	TABLES	Table name

Data modeling

Table 27. Table - Table.tag (continued)

ERwin	Command line tag	Data Warehouse Center
Database Name	DBNAME	Not applicable
Table Owner	OWNER	Table schema
Table Comment	SHRTDESC	Description

Table 28. Column - Column.tag

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

ERwin to Information Catalog Manager mapping

Table 29 through Table 34 on page 280 shows how the main ERwin object attributes correspond to the Information Catalog Manager tags:

Table 29. Database - Database.tag

ERwin	Command line tag	Information Catalog Manager interface
Diagram Name	NAME	Database name
Diagram Author	RESPNSBL	Database owner
Database Name	DBNAME	Database name
Database Version	DBTYPE	Database type
Diagram Description	SHRTDESC	Short Description

Table 30. Table - TableOrView.tag

ERwin	Command line tag	Information Catalog Manager interface
Table Name	NAME	Table name
Table Name	TABLES	Table name

Table 30. Table - TableOrView.tag (continued)

ERwin	Command line tag	Information Catalog Manager interface
Database Name	DBNAME	Database name
Table Owner	OWNER	Table owner
Table Comment	SHRTDESC	Short Description
ERwin API	TABLVIEW	Definition represents a view

Table 31. Column - ColumnOrField.tag

ERwin	Command line tag	Information Catalog Manager interface
Column Name	NAME	Column name
Datatype	DATATYPE	Datatype of column
Length	LENGTH	Length of column
Scale	SCALE	Scale of column
Null Option	NULLS	Can column be null (?)
Position	POSNO	Column position
Primary Key	KEYPOSNO	Position of column in primary key
ERwin API	ISKEY	Is column part of key (?)
ERwin API	UNIKEY	Is column a unique key (?)
Database Name	DBNAME	Database name
Table Owner	OWNER	Table owner
Table Name	TABLES	Table name
Column Comment	SHRTDESC	Short Description
ERwin	ISTEXT	Is data text (?)
ERwin API	IDSRES	Resolution of data

Table 32. Model - Model.tag

ERwin	Command line tag	Information Catalog Manager interface
ER1 file name	NAME	Model name
Diagram Author	RESPNSBL	For further information...
Diagram Description	SHRTDESC	Short Description

Table 33. Entity - Entity.tag

ERwin	Command line tag	Information Catalog Manager interface
Entity Name	NAME	Entity name
Notes	SHRTDESC	Short Description

Data modeling

Table 33. Entity - Entity.tag (continued)

ERwin	Command line tag	Information Catalog Manager interface
Definition	LONGDESC	Long Description
Entity Owner	RESPNSBL	For further information...

Table 34. Attribute - Attribute.tag

ERwin	Command line tag	Information Catalog Manager interface
Attribute Name	NAME	Attribute name
Notes	SHRTDESC	Short Description
Definition	LONGDESC	Long Description
Datatype	DATATYPE	Datatype of member
Length	LENGTH	Length of member

Chapter 11. Refreshing a 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

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

1. Using the Essbase Administration 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 Administration 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 Administration Manager, write and test the load rules that will load the data sources into the Essbase database. Save the load rules into the database or as files on the warehouse agent site.

You can also define calculation scripts to run after the data is loaded. Save the calculation scripts in files on the warehouse agent site.

For information about defining load rules and calculation scripts, see the *OLAP Server Database Administrator's Guide*.

7. Define a step that uses one of the warehouse programs 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.

Refreshing a OLAP Server database

Figure 19 shows the data flow between the Data Warehouse Center and OLAP Server.

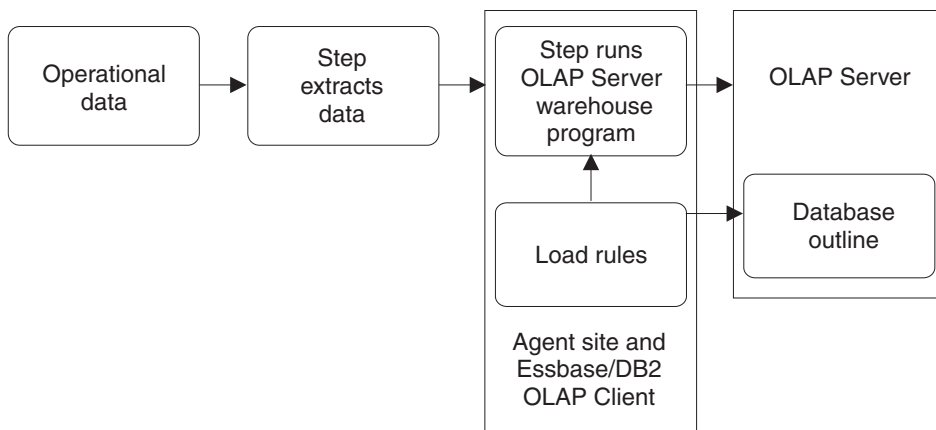


Figure 19. Data flow between the Data Warehouse Center and OLAP Server

For more information about completing this procedure, see the Data Warehouse Center online help and *OLAP Server: Using OLAP Server*.

Defining values for the OLAP Server: Default calc (ESSCALC1) warehouse program

Use the OLAP Server: Default calc (ESSCALC1) warehouse program to call the default calc script that is associated with the target database. The OLAP Server can be a OLAP Server or an Essbase server.

When you select a step that uses the OLAP Server: Default calc (ESSCALC1) warehouse program, do not link the step to a source or a target.

This warehouse program is available for the following operating systems:

- Windows NT
- AIX
- Solaris Operating Environment
- AS/400

The Column Mapping page is not available for this step.

To define values for a step that runs the warehouse program:

1. Open the step notebook.

2. Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149.
3. On the Parameters page, in the **OLAP Server system name** field, type the name of the OLAP server that you want to use.
4. In the **OLAP Server application name** field, type the name of the OLAP server application to use.
5. In the **OLAP Server database name** field, type the name of the OLAP server database that has the default calc script to call.
6. In the **OLAP Server user ID** field, type the user ID to use to connect to the OLAP server database.
7. In the **OLAP Server password** field, type the password for the user ID.
8. In the **Verify OLAP Server password** field, type the password again.
9. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
10. Click **OK** to save your changes and close the step notebook.

Defining values for the OLAP Server: Calc with calc rules (ESSCALC2) warehouse program

Use the OLAP Server: Calc with calc rules (ESSCALC2) warehouse program to apply the specified calc script to an OLAP server database. The OLAP server can be a OLAP Server or an Essbase server.

When you select a step that uses this warehouse program, do not link the step to a source or a target.

This warehouse program is available for the following operating systems:

- Windows NT
- AIX
- Solaris Operating Environment
- AS/400

The Column Mapping page is not available for this step.

To define values for a step that runs this warehouse program:

1. Open the step notebook.
2. Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149.

Refreshing a OLAP Server database

3. On the Parameters page, in the **OLAP Server system name** field, type the name of the OLAP Server to use.
4. In the **OLAP Server application name** field, type the name of the OLAP server application to use.
5. In the **OLAP Server database name** field, type the name of the OLAP server database that has the calc script to call.
6. In the **OLAP Server user ID** field, type the user ID to use to connect to the OLAP server database.
7. In the **OLAP Server password** field, type the password for the user ID.
8. In the **Verify OLAP Server password** field, type the password again.
9. In the **Calc script file name** field, type the name of the calc script file to use. If the calc script location flag indicates that the file is on an OLAP server system, the file name must follow the convention for specifying file names on the client or the server. For example, if you are using a file on an Essbase server, you must specify the file name without an extension.

You must also use the Essbase Application Manager to build the calc script file. To do this, you must store the calc script on the server that uses the Essbase Application Manager.

If the calc script file location flag in your program indicates that the file is on an agent site, you must specify a fully qualified file name using the file system of the agent site.

10. From the **Calc script file location** radio button group, select the location of the calc script file (indicated by the location flag).
11. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
12. Click **OK** to save your changes and close the step notebook.

Defining values for the OLAP Server: Free text data load (ESSDATA1) warehouse program

Use the OLAP Server: Free text data load warehouse program to load data from a comma-delimited flat file into a multidimensional OLAP Server database using free-form data loading. The OLAP server can be a OLAP Server or an Essbase server.

The OLAP Server: Free text data load warehouse program uses the file that is selected as a source for the step in the process. The step must have only one source file selected. The data in the source file must match the OLAP server outline values exactly. The selected source file must be on the agent site.

This warehouse program is available for the following operating systems:

- Windows NT
- AIX
- Solaris Operating Environment
- AS/400

The Column Mapping page is not available for this step.

To define values for a step that runs this warehouse program:

1. Open the step notebook.
2. Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149.
3. On the Parameters page, in the **OLAP Server system name** field, type the name of the OLAP server to use.
4. In the **OLAP Server application name** field, type the name of the OLAP server application to use.
5. In the **OLAP Server database name** field, type the name of the OLAP server database that has the default calc script to call.
6. In the **OLAP Server user ID** field, type the user ID to use to connect to the OLAP server database.
7. In the **OLAP Server password** field, type the password for the user ID.
8. In the **Verify OLAP Server password** field, type the password again.
9. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
10. Click **OK** to save your changes and close the step notebook.

Defining values for the OLAP Server: Load data from file with load rules (ESSDATA2) warehouse program

Use the OLAP Server: Load data from file with load rules (ESSDATA2) warehouse program to load data from a flat file into a multidimensional OLAP Server database using load rules. The OLAP server can be a OLAP Server or an Essbase server.

The warehouse program uses the selected file as a source for the step. The step must have only one source file selected. The source file name must follow the conventions for specifying file names on the client or server of your OLAP server system.

If the file location flag in your program indicates that the file is on an agent site, you must specify a fully qualified file name in the **Load rule file name** field on the Parameters page of the Step notebook.

Refreshing a OLAP Server database

If the file location flag indicates that the source file is on the OLAP server, you must specify the file name without an extension in the **Load rule file name** field on the Parameters page of the Step notebook.

This warehouse program is available for the following operating systems:

- Windows NT
- AIX
- Solaris Operating Environment
- AS/400

The Column Mapping page is not available for this step.

To define values for a step that runs this warehouse program:

1. Open the step notebook.
2. Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149.
3. On the Parameters page, in the **OLAP Server system name** field, type the name of the OLAP server to use.
4. In the **OLAP Server application name** field, type the name of the OLAP server application to use.
5. In the **OLAP Server database name** field, type the name of the OLAP server database that has the default calc script to call.
6. In the **OLAP Server user ID** field, type the user ID to use to connect to the OLAP server database.
7. In the **OLAP Server password** field, type the password for the user ID.
8. In the **Verify OLAP Server password** field, type the password again.
9. In the **Load rules file name** field, type the name of the load rules file that you want the program to use. If the file is on the OLAP server, the file name cannot have an extension. If the file is located on the agent site, you must specify the fully qualified file name.
10. In the **Load rules file location flag** area, click the location of the rules file.
11. In the **Source file location flag** area, click the location of the source file.
12. In the **OLAP Server utility abort flag** area, specify whether to stop the OLAP server utility if there is an error.
13. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
14. Click **OK** to save your changes and close the step notebook.

Defining values for an OLAP Server: Load data from SQL table with load rules (ESSDATA3) warehouse program

Use the OLAP Server: Load data from SQL table with load rules (ESSDATA3) warehouse program to load data from an SQL source into a multidimensional OLAP Server database using load rules. The OLAP server can be a OLAP Server or an Essbase server.

The OLAP Server: Load data from an SQL table with load rules warehouse program uses the user ID and password defined for the source database for the step in the Steps notebook.

When you select a step that uses this warehouse program, link a source table to the step, but do not link the step to a target.

This warehouse program is available for the following operating systems:

- Windows NT
- AIX
- Solaris Operating Environment
- AS/400

The Column Mapping page is not available for this step.

To define values for a step that uses this warehouse program:

1. Open the step notebook.
2. Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149 .
3. On the Parameters page, in the **OLAP Server system name** field, type the name of the OLAP server to use.
4. In the **OLAP Server application name** field, type the name of the OLAP server application to use.
5. In the **OLAP Server database name** field, type the name of the OLAP server database that has the default calc script to call.
6. In the **OLAP Server user ID** field, type the user ID to use to connect to the OLAP server database.
7. In the **OLAP Server password** field, type the password for the user ID.
8. In the **Verify OLAP Server password** field, type the password again.
9. In the **Load rules file name** field, type the name of the load rules file that you want the program to use. If the file is on the OLAP Server, the file name cannot have an extension. If the file is located on the agent site, you must specify the fully qualified file name.

Refreshing a OLAP Server database

10. In the **Load rules file location flag** area, click the location of the rules file.
11. In the **OLAP Server utility abort flag** area, specify whether to stop the OLAP server utility if there is an error.
12. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
13. Click **OK** to save your changes and close the step notebook.

Defining values for an OLAP Server: Load data from a file without using load rules (ESSDATA4) warehouse program

Use the OLAP Server: Load data from a file without using load rules (ESSDATA4) program to load data from a flat file into a multidimensional OLAP server database without using load rules. The OLAP server can be a OLAP Server or an Essbase server.

When you select a step that uses this warehouse program, link a source file to the step, but do not link the step to a target table.

This warehouse program is available for the following operating systems:

- Windows NT
- AIX
- Sun Solaris Operating Environment
- AS/400

The Column Mapping page is not available for this step.

To define values for a step that uses this warehouse program:

1. Open the step notebook.
2. Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149.
3. On the Parameters page, in the **OLAP Server system name** field, type the name of the OLAP server to use.
4. In the **OLAP Server application name** field, type the name of the OLAP server application to use.
5. In the **OLAP Server database name** field, type the name of the OLAP server database that has the default calc script to call.
6. In the **OLAP Server user ID** field, type the user ID to use to connect to the OLAP server database.
7. In the **OLAP Server password** field, type the password for the user ID.

8. In the **Verify OLAP Server password** field, type the password again.
9. In the **Source file location flag** area, click the location of the source file.
10. In the **OLAP Server utility abort flag** area, specify whether to stop the OLAP server utility if there is an error.
11. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
12. Click **OK** to save your changes and close the step notebook.

Defining values for an OLAP Server: Update outline from file (ESSOTL1) warehouse program

Use the OLAP Server: Update outline (ESSOTL1) warehouse program to update an OLAP server outline from a source flat file using load rules. The OLAP server can be a OLAP Server or Essbase server.

The OLAP Server: Update outline warehouse program uses the selected file as a source for the step. The step must have only one source file selected. The source file name must follow the conventions for your OLAP server for specifying file names on the client or server.

If the file location flag in your warehouse program indicates that the file is on an agent site, you must specify a fully qualified file name in the **Load rules file name** field on the Parameters page of the Step notebook.

If the file location flag indicates that the source file is on the OLAP server, you must specify the file name without an extension in the **Load rules file name** field on the Parameters page of the Step notebook.

For more information, see the documentation for your OLAP server.

When you select a step that uses this warehouse program, link a source file to the step, but do not link the step to a target table.

This warehouse program is available for the following operating systems:

- Windows NT
- AIX
- Solaris Operating Environment
- AS/400

The Column Mapping page is not available for this step.

Refreshing a OLAP Server database

To define values for a step that uses this warehouse program:

1. Open the step notebook.
2. Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149.
3. On the Parameters page, in the **OLAP Server system name** field, type the name of the OLAP server to use.
4. In the **OLAP Server application name** field, type the name of the OLAP server application to use.
5. In the **OLAP Server database name** field, type the name of the OLAP server database that has the default calc script to call.
6. In the **OLAP Server user ID** field, type the user ID to use to connect to the OLAP server database.
7. In the **OLAP Server password** field, type the password for the user ID.
8. In the **Verify OLAP Server password** field, type the password again.
9. In the **Load rules file name** field, type the name of the load rules file that you want the program to use. If the file is on the OLAP server, the file name cannot have an extension. If the file is located on the agent site, you must specify the fully qualified file name.
10. In the **Load rules file location flag** area, click the location of the rules file. In the **Source file location flag** area, click the location of the source file.
11. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
12. Click **OK** to save your changes and close the step notebook.

Defining values for an OLAP Server: Update outline from SQL table (ESSOTL2) program

Use the OLAP Server: Update outline from SQL table (ESSOTL2) warehouse program to update an OLAP server outline from an SQL source using load rules. The OLAP server can be a OLAP Server or an Essbase server.

When you select a step that uses this warehouse program, link a source table to the step, but do not link the step to a target table.

This warehouse program is available for the following operating systems:

- Windows NT
- AIX
- Solaris Operating Environment
- AS/400

Refreshing a OLAP Server database

The Column Mapping page is not available for this step.

To define values for a step that uses this warehouse program:

1. Open the step notebook.
2. Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149.
3. On the Parameters page, in the **OLAP Server system name** field, type the name of the OLAP server to use.
4. In the **OLAP Server application name** field, type the name of the OLAP server application to use.
5. In the **OLAP Server database name** field, type the name of the OLAP server database that has the default calc script to call.
6. In the **OLAP Server user ID** field, type the user ID to use to connect to the OLAP server database.
7. In the **OLAP Server password** field, type the password for the user ID.
8. In the **Verify OLAP Server password** field, type the password again.
9. In the **Load rules file name** field, type the name of the load rules file that you want the program to use. If the file is on the OLAP server, the file name cannot have an extension. If the file is located on the agent site, you must specify the fully qualified file name.
10. In the **Load rules file location flag** area, click the location of the rules file.
11. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
12. Click **OK** to save your changes and close the step notebook.

Refreshing a OLAP Server database

Chapter 12. Maintaining the warehouse database

This chapter describes how to use warehouse programs to maintain your warehouse database:

- You can use the DB2 reorganize warehouse program to rearrange a table in physical storage. This 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.
- You can use the DB2 runstats warehouse program to gather statistics about the physical and logical characteristics of a table and its indexes. DB2 Universal Database uses these statistics to determine the best way to access your data.

For information about tuning and configuring your database for optimal performance, see *DB2 Universal Database Administration Guide: Performance*.

Reorganizing data

You can reorganize data in a DB2 Universal Database table or in a DB2 UDB for OS/390 table space or index.

Defining values for a DB2 Universal Database REORG program

Use the Step Properties notebook for the DB2 UDB REORG warehouse program to create a step that can be used to reorganize data in a DB2 Universal Database table.

You can use a warehouse source or target as a source for this step subtype. The REORG program writes to the source table.

The Column Mapping page is not available in this step.

To define values for a step that runs a DB2 UDB REORG warehouse program:

1. Open the step notebook.
2. Specify general information about the program. For more information, see “Providing general information about a step” on page 149.
3. Optional: On the Parameters page, specify information for the REORG step:
 - In the **Using temporary table space** field, type the name of the temporary table space that should be used during the REORG step.

Maintaining the warehouse database

- In the **Using index** field, type the name of the index that should be used during the REORG step.
- 4. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
- 5. Click **OK** to save your changes and close the step notebook.

Defining values for a DB2 for OS/390 REORG program

Use the DB2 for OS/390 REORG Data Warehouse Center program to reorganize a table space to improve access performance and to reclaim fragmented space. In addition, you can reorganize a single partition of a partitioned table space.

To define values for a step that uses this warehouse program:

1. Open the step notebook.
2. Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149.
3. On the Parameters page, specify how you want the program to handle the data before, during, and after the reorganization. To specify reorganization parameters:
 - a. In the **Table Space** field, type the name of the table space that contains your table (and optionally the database to which it belongs).
 - b. Optional: Select the **Log records during reload** check box to log records during the reload phase of the reorganization program. If the records are not logged, the table space is recoverable only after an image copy is made. Logging records during the reload phase is not allowed for any table space in DSNDB01 or DSNDB06, or if you selected SHRLEVEL REFERENCE or CHANGE. Logging records is required for LOB table spaces.
 - c. Optional: Select the **Logically reset and reuse the data sets** check box to load data into a non-empty data set.
 - d. Optional: Specify options for improving performance after unloading the data:
 - Select the **Unload data by table space scan and sort data in clustering order** check box to unload the data using a sequential scan. If you do not specify this option, the reorganization program uses the clustering index to unload the data.
 - Select the **Sort index keys in parallel** check box to specify that index keys will be sorted in parallel during the SORTBLD phase to improve performance.
 - e. Optional: Specify the type of access allowed:
 - Select the **No application access** check box to unload data from the area being reorganized (while applications can read but cannot write

to the area), reload data into that area (while applications have no access), and then allow read/write access again. This selection is required for LOB table spaces.

- Select the **Read-only application access** check box to unload data from the area being reorganized (while applications can read but cannot write to the area), reload data into a shadow copy of that area (while applications can read but cannot write to the original copy), switch applications' future access from the original copy to the shadow copy by exchanging the names of the data sets, and then allow read/write access again.
 - Select the **Read-write application access** check box to unload data from the area being reorganized (while applications can read and write to the area), reload data into a shadow copy of that area (while applications have read/write access to the original copy of the area), apply the log of the original copy to the shadow copy (while applications can read and usually write to the original copy), switch applications' future access from the original copy to the shadow copy by exchanging the names of the data sets, and then allow read/write access again.
- f. Optional: Specify mapping table information:
- In the **Owner** field, type the name of the owner of the mapping table.
 - In the **Name** field, type the name of the mapping table that the reorganization program uses to map between the record identifiers (RIDs) of data records in the original copy of the area and the corresponding RIDs in the shadow copy. This table must already exist and include an index. The table space that contains the mapping table must be segmented and cannot be the table space to be reorganized.
- g. Optional: In the **Data set name** field for the **Unload data set** option, type the data set name for the data set that contains the data to be unloaded. The default name is SYSREC. The data set must be a sequential data set that is readable by the MVS BSAM access method.
- Specify a direct-access storage device type in which you are authorized to allocate space. Assign frequently accessed objects to fast devices and seldom-used objects to slower devices. You can also specify a tape device.
 - Allocate space compatible for your device. If your storage device is disk, the space allocation is in cylinders. If your device type is tape, you can leave this blank.
- h. Optional: In the **Data set name** field for the **Primary copy data set at local site** option, type a name for the primary copy data set. This data set contains image copy data sets. The default name is SYSCOPY.

Maintaining the warehouse database

- Specify a direct-access storage device type in which you are authorized to allocate space. Assign frequently accessed objects to fast devices and seldom-used objects to slower devices. You can also specify a tape device.
 - Allocate space that is compatible for your device. If your storage device is disk, the space allocation is in cylinders. If your device type is tape, you can leave this field blank.
- i. For the **Input sort data set name** and **Output sort data set** options, type the names of the temporary work files for the sort input and output (SYSUT1 and SORTOUT) data sets. These files are required when the tables have indexes, or SORTKEYS on the options page of this notebook is specified with no estimate or an estimate of 0.
 - Specify a direct-access storage device type in which you are authorized to allocate space. Assign frequently accessed objects to fast devices and seldom-used objects to slower devices. You can also specify a tape device.
 - Allocate space that is compatible for your device. If your storage device is disk, the space allocation is in cylinders. If your device type is tape, you can leave this field blank.
 - j. Optional: In the **Device type for DFSORT temporary data sets** field, type the device type for temporary data sets to be dynamically allocated by DFSORT. This can be any device type that is acceptable to the DYNALLOC parameter of the SORT or OPTION options for DFSORT.
 - k. Optional: Select the **Number of DFSORT temporary data sets** check box to use DFSORT temporary data sets. Type the number of data sets to be dynamically allocated by the sort application program. The SORTNUM field is used only if a device type is selected in the previous step.
 - l. Optional: Click the **Advanced** push button to open the Reorganize Table Space notebook. From this notebook you can specify additional options for reorganizing the table space.
4. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
 5. Click **OK** to save your changes and close the step notebook.

Defining values for a DB2 for OS/390 Utility program

Use the DB2 for OS/390 Utility warehouse program to run any utility supported by DSNUTILS. For more information about DSNUTILS, see *DB2 Universal Database for OS/390 and z/OS Utility Guide and Reference*.

To define a values for a step that uses this warehouse program:

1. Open the step notebook.

- Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149.
- On the Parameters page, double-click the **Parameter value** field to add values for the corresponding parameters shown in the **Parameter name** field. Here is a list of the parameters and their descriptions:

UTILITY_ID

Specifies a unique identifier (UID) for this utility within DB2. This is a string of 1-16 characters.

RESTART

Specifies whether this warehouse program 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. For more information about the utilities you can specify here, see the *DB2 Universal Database for OS/390 and z/OS Utility Guide and Reference*.

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

Maintaining the warehouse database

table spaces as a discard data set to hold records not reloaded. If you specify DISCDSN, it will be allocated to the SYSDISC DDNAME.

DISCDEVT

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set specified by DISCDSN resides.

DISCSpace

Specifies the number of cylinders to use as the primary space allocation for the data set specified by DISCDSN. The secondary space allocation will be 10% of the primary space allocation.

PNCHDSN

Specifies the cataloged data set name that is used when reorganizing table spaces with the keywords UNLOAD EXTERNAL or DISCARD. The data set is used to hold the generated LOAD utility control statements. If you specify a value for PNCHDSN, it will be allocated to the SYSPUNCH DDNAME.

PNCHDEVT

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set specified by PNCHDSN resides.

PNCHSPACE

Specifies the number of cylinders to use as the primary space allocation for the data set specified by PNCHDSN. The secondary space allocation will be 10% of the primary space allocation.

COPYDSN1

Specifies the name of the target (output) data set. If you specify COPYDSN1, it will be allocated to the SYSCOPY DDNAME.

COPYDEVT1

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set specified by COPYDSN1 resides.

COPYSPACE1

Specifies the number of cylinders to use as the primary space allocation for the data set specified by COPYDSN1. The secondary space allocation will be 10% of the primary space allocation.

COPYDSN2

Specifies the name of the cataloged data set used as a target (output) data set for the backup copy. If you specify COPYDSN2, it will be allocated to the SYSCOPY2 DDNAME.

COPYDEVT2

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set specified by COPYDSN2 resides.

COPYSPACE2

Specifies the number of cylinders to use as the primary space allocation for the data set specified by COPYDSN2. The secondary space allocation will be 10% of the primary space allocation.

RCPYDSN1

Specifies the name of the cataloged data set used as a target (output) data set for the remote site primary copy. If you specified RCPYDSN1, it will be allocated to the SYSRCPY1 DDNAME.

RCPYDEVT1

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the RCPYDSN1 data set resides.

RCPYSPACE1

Specifies the number of cylinders to use as the primary space allocation for the data set specified by RCPYDSN1. The secondary space allocation will be 10% of the primary space allocation.

RCPYDSN2

Specifies the name of the cataloged data set used as a target (output) data set for the remote site backup copy. If you specify RCPYDSN2, it will be allocated to the SYSRCPY2 DDNAME.

RCPYDEVT2

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set specified by RCPYDSN2 resides.

RCPYSPACE2

Specifies the number of cylinders to use as the primary space allocation for the data set specified by RCPYDSN2. The secondary space allocation will be 10% of the primary space allocation.

WORKDSN1

Specifies the name of the cataloged data set that is required as a work data set for sort input and output. If you specify WORKDSN1, it will be allocated to the SYSUT1 DDNAME.

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.

Maintaining the warehouse database

WORKSPACE1

Specifies the number of cylinders to use as the primary space allocation for the data set specified by WORKDSN1. The secondary space allocation will be 10% of the primary space allocation.

WORKDSN2

Specifies the name of the cataloged data set that is required as a work data set for sort input and output. It is required if you are using reorganizing non-unique type 1 indexes. If you specify WORKDSN2, it will be allocated to the SORTOUT DDNAME.

WORKDEVT2

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set specified by WORKDSN2 resides.

WORKSPACE2

Specifies the number of cylinders to use as the primary space allocation for the WORKDSN2 data set. The secondary space allocation will be 10% of the primary space allocation.

MAPDSN

Specifies the name of the cataloged data set that is required as a work data set for error processing during LOAD with ENFORCE CONSTRAINTS. It is optional for LOAD. If you specify MAPDSN, it will be allocated to the SYSMAP DDNAME.

MAPDEVT

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set specified by MAPDSN resides.

MAPSPACE

Specifies the number of cylinders to use as the primary space allocation for the MAPDSN data set. The secondary space allocation will be 10% of the primary space allocation.

ERRDSN

Specifies the name of the cataloged data set that is required as a work data set for error processing. If you specify ERRDSN, it will be allocated to the SYSERR DDNAME.

ERRDEVT

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set specified by ERRDSN resides.

ERRSPACE

Specifies the number of cylinders to use as the primary space

allocation for the ERRDSN data set. The secondary space allocation will be 10% of the primary space allocation.

FILTRDSN

Specifies the name of the cataloged data set that is required as a work data set for error processing. If you specify FILTRDSN, it will be allocated to the FILTER DDNAME.

FILTRDEVT

Specifies a unit address, a generic device type, or a user-assigned group name for a device on which the data set specified by FILTRDSN resides.

FILTRSPACE

Specifies the number of cylinders to use as the primary space allocation for the FILTRDSN data set. The secondary space allocation will be 10% of the primary space allocation.

4. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
5. Click **OK** to save your changes and close the step notebook.

Updating system catalog statistics

You can update system catalog statistics for a DB2 UDB or DB2 for OS/390 database.

Defining values for a DB2 UDB RUNSTATS program

Use the Step Properties notebook for the DB2 UDB RUNSTATS warehouse program 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.

The RUNSTATS program uses a warehouse target as a source and a target. Link a warehouse target to the step in the Process Model window before you define the values for the step.

The Column Mapping page is not available in this step.

To define values for a step that runs a DB2 UDB RUNSTATS warehouse program:

1. Open the step notebook.

Maintaining the warehouse database

2. Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149.
3. Optional: On the Parameters page, specify information for the RUNSTATS warehouse program:
 - Specify the level of statistics you want to gather for the table by clicking a radio button under **Statistics for the table**.
 - Specify the level of statistics you want to gather for the table’s indexes by selecting a radio button under **Statistics for the indexes**.
 - Use the **Share level** radio buttons to specify the type of access you want other users to have to the table while the statistics are being gathered.
4. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
5. Click **OK** to save your changes and close the step notebook.

Defining values for a DB2 OS/390 RUNSTATS program

Use the DB2 for OS/390 RUNSTATS warehouse program to gather summary information about the characteristics of data in table spaces. The output is recorded in the DB2 for OS/390 catalog tables. DB2 uses this information to select access paths to data during the bind process. The information is available to the database administrator for evaluating database design and to aid in determining when table spaces or indexes must be reorganized.

The Column Mapping page is not available for this step subtype.

To define values for a step that uses this warehouse program:

1. Define a warehouse target where the utility will run.
2. Add a RUNSTATS step to the process.
3. Connect the step to the target.
4. Open the step notebook.
5. Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149.
6. On the Parameters page, click one of the **Share level** radio buttons to specify the access that other programs can use on the table space while you are gathering statistics:
 - Specify **Reference only** if you want to allow read-only access by other programs. This is the default.
 - Specify **Allow changes** if you want to allow other programs to change the table space or index. Uncommitted data can be collected into statistical summaries.

7. Click one of the **Report statistics** radio buttons to specify if a set of messages is generated to report the collected statistics. Specify **Yes** if you want the set of messages sent to the SYSPRINT data set. Specify **No** to not send messages to SYSPRINT. This is the default.
8. Click one of the **Update options on catalog tables** radio buttons to specify whether the collected statistics are inserted into the DB2 for OS/390 catalog tables:

Access path and space statistics

The result of gathering the statistics updates all the statistics in the DB2 catalog.

Access path statistics only

The result of gathering the statistics updates only the catalog table columns used for access path selection.

Space statistics only

The result of gathering the statistics updates only the catalog table columns that help a database administrator assess the status of a particular table space or index.

Do not update catalog tables

The result of gathering the statistics does not update any catalog statistics. Available only when the Report Statistics radio button is **YES**.

9. In the **Percent rows sampled for non-index columns** field, specify a percentage of rows to sample when collecting non-indexed column statistics. Valid values are 1 to 100 percent. The default is 25 percent. Do not use this field if your table space is a LOB table space.
10. The columns in a table for which you are gathering statistics are listed in the **Available columns** list. Click on a column and click > to move a column from the **Available columns** list to the **Selected columns** list. You can move all the columns in the table by clicking >>. Click < or << to move columns from the **Selected columns** list to the **Available columns** list. All columns in the **Selected columns** list are the columns on which statistics are gathered. You can select a maximum of 10 columns with one table space statistics operation.
11. In the **Tablespace** field, type the name of the table space in uppercase letters. If the tablespace is not in database DSNDB04, you must qualify the tablespace name with the name of the database that contains it. For example, SAMPLE.EMPLOYEE.
12. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
13. Click **OK** to save your changes and close the step notebook.

Using MQSeries with the Data Warehouse Center

With the Data Warehouse Center you can access data from an MQSeries message queue as a DB2 database view. A wizard is provided to help you create a DB2 table function and the DB2 view through which you can access the data. Each MQSeries message is treated as a delimited string, which is parsed according to your specification and returned as a result row. In addition, MQSeries messages that are XML documents can be accessed as a warehouse source. Using the Data Warehouse Center, you can import metadata from an MQSeries message queue and a DB2 XML Extender Document Access Definition (DAD) file.

Creating views for MQSeries messages

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

Requirements

- DB2 Universal Database Version 7.2.
- DB2 Warehouse Manager Version 7.2
- MQSeries Support.

For more information about MQSeries requirements, see the topic, "MQSeries" in the release notes.

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.

See the SQL Reference section of the Release Notes for the maximum length of the MQ messages.

Creating a view for MQSeries messages

To create a view for MQSeries messages:

1. From the Data Warehouse Center window, expand the **Warehouse Sources** tree.
2. Expand the warehouse source that is to contain the view.
3. Right-click the **Views** folder, and click **Create for MQSeries messages...**

The MQSeries wizard opens. When you complete the wizard, a new view is created in the Data Warehouse Center. When you select the view, the

MQSeries queue is accessed and each message is parsed as a delimited string according to your specifications in the wizard.

Importing MQSeries messages and XML metadata

The following requirements and restrictions apply if you are importing MQSeries and XML metadata.

Requirements

- DB2 Universal Database Version 7.2.
- DB2 XML Extender Version 7.2.
- MQSeries Support.

For more information about MQSeries requirements, see the topic, "MQSeries," in the release notes. See the setup section for user-defined functions for information on setting up the warehouse source.

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.

Importing MQSeries messages and XML metadata

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. See the DB2 XML Extender Version 7.2 Release Notes for more information.
 - 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. See the DB2 XML Extender Version 7.2 Release Notes for more information.
2. Right-click **Warehouse**, and click **Import Metadata** → **MQSeries** to open the **Import Metadata** window.
3. In the **AMI service** field, type the service point that a message is sent to or retrieved from.
4. In the **AMI policy** field, type the policy that the messaging system will use to perform the operation.
5. In the **DAD file** field, type the name of the DB2 XML Extender DAD file, or search for a file to select by clicking the ellipsis (...). This file must be local.
6. In the **Warehouse Target** field, select the name of the warehouse target where the step will run. The warehouse target must already be defined.

Maintaining the warehouse database

- 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.
- 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**.
- Click **OK**.
The Import Metadata window closes.

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

If the warehouse target agent site is remote, you must change the step parameter:

- Right-click the step, and click **Properties** to open the Properties notebook for the step.
- Click the **Parameters** tab.
- Change the name of the DAD file parameter to the name of the same DAD file on the remote warehouse target agent site.
- Make sure the **Agent Site** field on the Processing Options page contains the correct agent site.

Using the MQSeries user-defined program

The MQSeries and XML stored procedure is called **MQXMLXF**, and is included with DB2 Data Warehouse Center Version 7.2 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 Table 35.

Table 35. Parameters for the MQXMLXF stored procedure.

Parameter	Values
MQSeries ServiceName	The name of the service point that a message is sent to or retrieved from.
MQSeries PolicyName	The name of the policy that the messaging system will use to perform the operation.

Table 35. Parameters for the MQXMLXF stored procedure. (continued)

DAD file name	The name of the DB2 XML Extender DAD file
TargetTableList	List of target tables of the step separated by commas
Option	REPLACE or APPEND
RUN ID	Step edition number (for logging purposes)

Note: The data type for all of the parameters in this table is CHARACTER.

If the **Option** parameter has a value of REPLACE, the stored procedure deletes all rows from the target tables. The stored procedure also calls the DB2 XML Extender stored procedure to populate the target tables for all existing MQSeries messages.

Error return codes

When the step runs, the stored procedure might return error code SQLCODE -443 and SQLSTATE 38600. To diagnose the error, see Table 36 for possible diagnostic texts.

Table 36. Diagnostic texts for error codes returned by the MQXMLXF user-defined program.

Error number	Description
AMIRC=xxxx;<log file name>	xxxx 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=xxxx;<log file name>	xxxx 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=xxxx;<log file name>	xxxx is the nonzero SQLCODE returned when an SQL request is performed. <log file name> indicates the location of the log file.

See the log file for more information about all errors.

Error log file

The Data Warehouse Center stores all diagnostic information in a log file when MQXMLXF runs. The name of the log file is mqxf<nnnnnnnnn>.log, where <nnnnnnnnn> 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

Maintaining the warehouse database

environment variable using the **db2set** command before the **db2start** command. The following example shows an environment command.

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

The log file is deleted if the step runs successfully.

Chapter 13. Extending the Data Warehouse Center with 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.

This chapter explains the following tasks:

- “Defining a user-defined program”
- “Using a user-defined program step” on page 313
- “Writing your own program for use with the Data Warehouse Center” on page 315

Defining 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.

For information about setting up a stored procedure for OS/390 or AS/400, see the *Warehouse Manager Installation Guide*.

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.

Defining a program group

A user program group is a logical group that contains related user-defined programs. You must create a user program group before you can define a user-defined program to the Data Warehouse Center.

To define a user program group:

Extending the Data Warehouse Center with user-defined programs

1. From the Data Warehouse Center, expand the **Administration** folder.
2. Expand the **Programs and Transformers** folder.
3. Right-click the **User-Defined Program and Transformers** folder and click **Define Group**.
4. Specify information for your program group. You can enter information into any of these fields.
 - In the **Name** field, type a new name for the step or you can keep the name that the Data Warehouse Center automatically supplied for the step.
 - Optional: In the **Administrator** field, type the name of the person who is responsible for the maintenance of this step.
 - Optional: In the **Description** field, type a business description for your agent definition. This description can be a maximum of 254 characters.
 - Optional: In the **Notes** field, type detailed information that might be helpful to users who can access this step.
5. Click **OK** to save your changes and close the notebook.

Specifying the program to use

To define a user-defined program to the Data Warehouse Center:

1. From the Data Warehouse Center, expand the **Administration** folder.
2. Expand the **Programs and Transformers** folder.
3. Expand the **User-Defined Program and Transformers** folder.
4. If you are defining a new program, right-click a program group, and click **Define Program**.
5. If you are editing an existing program definition, expand the program group folder. Then, right-click the program, and click **Properties**.
6. Specify information for your program:
 - In the **Name** field, type a new name for the step.
 - In the **Administrator** field, type the name of the person who is responsible for the maintenance of this program.
 - In the **Description** field, type a business description for your program. This description can be a maximum of 254 characters.
 - In the **Notes** field, type detailed information that might be helpful to users who can access this step.
 - In the **Program type** field, select the type of program that you are defining.
 - In the **Program Name** field, type the fully qualified name of the program.
 - If the program type is dynamic link library, type the function name in the **Function Name** field.

Extending the Data Warehouse Center with user-defined programs

- If the program type is DB2 stored procedure, type the schema and procedure name in the **Stored Procedure Name** field. Use the following syntax:

schema.procedurename

Specifying the agent site to use

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.

If you are defining a transformer, you must ensure that the warehouse agent that you associate with this program is associated with the database where your transformer exists. If you assign a warehouse agent to this program definition that is not associated with the target database specified on this page, you will receive a warning when you create a step that uses this program.

For example, you have the following warehouse agent, source, and targets:

Agent	Source	Target
Agent A	Source 1	Target 1
		Target 2

Suppose Agent A is associated with Source 1 and Target 1. Now, suppose that your transformer exists on Target 2. If you select Agent A in your transformer program definition, you will receive a warning when you try to create a step. This warning is issued because Agent A is not associated with the target that contains the transformer.

To use Agent A, you must associate it with Target 2, or you must store the transformer on Target 1 and add Target 1 to the target resources definition:

1. Depending on the type of program that you are defining, click the **Agent sites** tab or the **Targets** tab.
2. If you are defining a program, from the **Available Agent Sites** list, select the agent site on which the program is installed, and click **Add**. The name of the agent site moves to the **Selected Agent Sites** list. Repeat this step for each agent site on which the program is installed.

Extending the Data Warehouse Center with user-defined programs

If you are defining a transformer, from the **Available Targets** list, select the target on which the stored procedure exists and click **Add**. The name of the target moves to the **Selected Targets** list.

3. Click **OK** to save your changes and close the notebook.

For more information about defining agent sites, see “Defining agent sites” on page 13 and the online help.

Specifying parameters

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.

If you want your user-defined program to extract data from the sources that are linked to the step that uses the program, you can include the **&STMT** token in your parameter list. This token directs the Data Warehouse Center to pass the **SELECT** statement that is defined for the step to the program. The **SELECT** statement is passed to the program as a one command-line parameter, enclosed in double quotation marks. Any embedded quotation marks are enclosed in escape characters. If the **SELECT** statement is exceptionally long, it might conflict with operating system guidelines for the maximum number of parameters.

For a complete list of parameters with predefined tokens, see “Select Parameters fields and controls” in the online help.

To add a system parameter or system token to a program definition:

1. Open the Define User Program notebook.
2. Click the **Parameters** tab.
3. Click **Add**.
4. In the **Type** area, click **System parameters** or **System tokens**.
5. In the **Available parameters** field, select a parameter or token. You can select multiple parameters or tokens by pressing the **CTRL** key while you make your selections.
6. Click **OK**.

The parameters or tokens that you select are listed on the Parameters page of the Define User Program notebook.

Extending the Data Warehouse Center with user-defined programs

Example: You are defining a user-defined program that checks for a file at regular intervals on a Windows NT 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

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.

Using a user-defined program step

After you define your user-defined program, you define steps that use the program for processing.

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.

If your user-defined program uses tokens for a source or target, you must link this step to the source or target.

To define a step that runs a user-defined program:

1. Define a process.
2. Right-click the process. Click **Open**. The Process Model window opens.
3. Click the user-defined program step icon in the left pane of the Process Model window. To view descriptions of the step icons, click **View** → **Legend**.
4. Select a step subtype (if applicable). Then, move your mouse into the Process Model window and click.
5. Link the step to any needed sources or targets.
6. Right-click the step, and click **Properties**.

To edit an existing step within the Process Model window, click a process, and click **Open**. The Process Model window opens. Right-click the step, and click **Properties**.

Extending the Data Warehouse Center with user-defined programs

To edit an existing step outside of the Process Model window, click a process in the Warehouse tree view. A list of objects that are contained in the process is displayed in the right frame. Right-click a step, and click **Properties**.

7. Specify information for your step:
 - In the **Name** field, type a new name for the step, or you can keep the name that the Data Warehouse Center automatically supplied for the step.
 - Optional: In the **Administrator** field, type the name of the person who is responsible for the maintenance of this step. This field is optional.
 - Optional: In the **Description** field, type a business description for your step. This description can be a maximum of 254 characters. This field is optional.
 - Optional: In the **Notes** field, type detailed information that might be helpful to users who can access this step. This field is optional.
8. On the Parameters page, view or edit the parameters and the parameter values that the program uses. To edit a parameter value, double-click the **Parameter value** field for a parameter, and type the new parameter value. If you change a parameter value, the change affects only the instance of the program that is run by this step. The original program parameter values are not changed.
9. On the Processing Options page, in the **Agent Sites** list, select an agent site where you want your step to run. The selections in this list are agent sites that are associated with the source tables, the target table, and the transformer or program that you are defining.
10. If you want the option to run your step at any time, select the **Run on demand** check box. Your step must be in test or production mode before you can run it.
11. Optional: Select the **Populate externally** check box if the step is populated externally, meaning that it is started in some way other than by the Data Warehouse Center. The step does require any other means of running in the Data Warehouse Center in order for you to change the mode to production.

If the **Populate externally** check box is not selected, then the step must have a schedule, be linked to a transient table that is input to another step, or be started by another program in order for you to change the mode to production.
12. In the **Retry** area, specify how many times you want the step to run again if it needs to be retried, and the amount of time that you want to pass before the next run of the step.
13. Click **OK** to save your changes and close the step notebook.

Extending the Data Warehouse Center with user-defined programs

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.

Testing steps that use user-defined programs

After you define a step that uses a user-defined program, promote the step to test mode and run it.

To promote a step to test mode:

1. Open the Process Model window.
2. Right-click the step to promote.
3. Click **Mode** → **Test**.

After the step is promoted, **Test** is selected when you right-click **Mode**.

To test a step:

1. Open the Process Model window.
2. Right-click the step.
3. Click **Test**.
4. Click **OK**.

The step is displayed on the Work in Progress window with a status of **Populating**. The processing should complete successfully. However, if the processing fails, see “Testing a step” on page 154 for information about debugging what went wrong.

After you test the step, you can schedule it in the same way as any other step. For more information about scheduling steps, see “Scheduling warehouse processes” on page 157

Writing your own program for use with the Data Warehouse Center

You can write programs in any language that supports one of the following program types: executable, batch program, dynamic link library, or stored procedure.

If the user-defined program has a program type of executable, batch program, batch command file, or dynamic link library, it must reside on the agent site. The warehouse agent starts the program at the scheduled time. On Windows NT, 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.

Extending the Data Warehouse Center with user-defined programs

If you write user-defined programs using Object REXX for Windows, complete the following procedure to enable these programs to run under Windows NT:

1. Define the warehouse agent or server service as a system process that can interact with the Windows NT desktop:
 - a. Select the agent or server service from the **Service** list.
 - b. Click **Startup**.
 - c. Click **System Account**.
 - d. Select the **Allow Service to Interact with Desktop** check box.
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.

Passing parameters

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 workstation. It uses the following parameters:

- File name
- Polling interval
- Timeout interval

Extending the Data Warehouse Center with user-defined programs

The program gets the parameters as shown in Figure 20:

```
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;
        }
    }
}
```

Figure 20. Reading parameters from the command line

Returning status information

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:

Extending the Data Warehouse Center with user-defined programs

- 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>
```

Extending the Data Warehouse Center with user-defined programs

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.

Figure 21 shows an example of 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>
```

Figure 21. Example of the feedback file

Extending the Data Warehouse Center with user-defined programs

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

Table 37 lists the possible combinations of these values and the results that they produce.

Table 37. Feedback file conditions and results

Conditions				Results	
				Step status ¹	Values of Error RC1 and RC2
The user-defined program return code is 0	No feedback file exists ²			Successful	RC1 = 0; RC2 = 0
	A feedback file exists ²	The value of <RC> in the feedback file is 0 ³	<WARNING> is not set in the feedback file	Successful	RC1 = 0; RC2 = 0
			The value of <WARNING> in the feedback file is 1	Warning	RC1 = 0; RC2 = 0
		The value of <RC> in the feedback file is non-0 ³	<WARNING> is not set in the feedback file	Failed	RC1 = 8410 (the user-defined program failed); RC2 = the value of <RC> in the feedback file
			The value of <WARNING> in the feedback file is 1	Warning	RC1 = 0; RC2 = the value of <RC> in the feedback file

Extending the Data Warehouse Center with user-defined programs

Table 37. Feedback file conditions and results (continued)

Conditions			Results		
			Step status ¹	Values of Error RC1 and RC2	
The user-defined program return code is nonzero	No feedback file exists ²		Failed	RC1 = 8410 (the user-defined program failed); RC2 = the code returned by the user-defined program	
	A feedback file exists ²	The value of <RC> in the feedback file is 0 ³	<WARNING> is not set in the feedback file	Successful	RC1 = 0; RC2 = 0
			The value of <WARNING> in the feedback file is 1	Warning	RC1 = 0; RC2 = 0
		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
			The value of <WARNING> in the feedback file is 1	Warning	RC1 = 0; RC2 = the value of <RC> in the feedback file

Extending the Data Warehouse Center with user-defined programs

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.

Chapter 14. Maintaining the Data Warehouse Center

This chapter explains concepts and procedures that are associated with maintaining the Data Warehouse Center. This chapter covers the following topics:

- “Backing up the Data Warehouse Center” on page 323
- “Expanding your warehouse” on page 324
- “Exporting and importing Data Warehouse Center metadata” on page 324
- “Publishing Data Warehouse Center metadata” on page 331
- “Changing the active warehouse control database” on page 337
- “Configuring the Data Warehouse Center” on page 339

Backing up the Data Warehouse Center

Back up your warehouse control database, as well as all your warehouse databases, at regular intervals.

To back up the warehouse control database, use the standard procedures for DB2 backup and recovery.

You need to stop the Data Warehouse Center before backing up the control database. To shut down the Data Warehouse Center:

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.

The warehouse logger and server are linked; 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**.

To back up your warehouse databases, use the backup and recovery procedures provided by your database manager.

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, or delete agents and administrative interfaces from your existing configuration.

You can also create a warehouse control database. If you create a warehouse control database, you must run the initialization process again. For more information, see “Changing the active warehouse control database” on page 337.

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.

For more information about installing and removing Data Warehouse Center components, see the *DB2 Universal Database Quick Beginnings* for your operating system and the *DB2 Warehouse Manager Installation Guide*.

Exporting and importing Data Warehouse Center metadata

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

- Windows NT
- AIX
- Solaris Operating Environment

Export processes and import processes use a large amount of system resources. You might want to limit the use of other programs while you are exporting and importing object definitions.

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. If you want to migrate the Data Warehouse Center, see the *DB2 Universal Database Quick Beginnings* for your operating system.

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.

Exporting metadata to another Data Warehouse Center

When you export metadata to a tag language file, the Data Warehouse Center finds the objects that you want to export and produces tag language statements to represent the objects. It then places the tag language statements into files that you can import into another Data Warehouse Center.

Planning to export metadata

If you export a process, you might export a large volume of metadata. The following objects are exported with a process:

- All steps that are contained within a process. You cannot export individual steps.
- The agent site for the process.
- Any user-defined programs that are associated with steps in the process.
- Internal objects that are required by the Data Warehouse Center, such as column maps for SQL steps.
- Schedule information for processes and steps. You can specify not to export this information.
- Any warehouse source or warehouse target definitions that are used by the process and underlying steps. You can choose not to export the warehouse source table definitions.
- Any other processes and steps whose output is used by the process that is being exported.
- All table definitions in a selected warehouse source or warehouse targets, if you export a warehouse source or warehouse target.
- All program definitions within a program group, if you export a program group.
- All table definitions within a warehouse schema, if you export a warehouse schema.

Exporting the metadata to a tag language file

To export a tag language file:

1. From the Data Warehouse Center, right-click the Warehouse node, and click **Export Metadata** → **Interchange file**.

The Export Metadata window opens.

2. Specify the tag language file to which you want to export the metadata. You can:

- Use the default tag language file name shown in the **File name** field.

On Windows NT systems, the file is written to the path and directory from which you start the DB2 Control Center. If you do not have write access to that directory (for example, if the Control Center is installed

Maintaining Data Warehouse Center

on a network drive), the file is written to the home path and directory specified by your Windows NT environment variable, usually the C drive and root directory.

On AIX systems, the file is written to the path and directory from which you invoke the Data Warehouse Center. If you do not have write access to that directory, the file is written to your home directory.

- Type a path and file name. You must include the correct file extension.
 - Click ... to open a window that you can use to view the available files and select one. Any existing file that you select is overwritten with the exported metadata.
3. In the **Available objects** list, select the objects that you want to export, and click >. The object moves to the **Selected objects** list. Repeat this step for all objects that you want to export.
If you want to export all available objects in a tree, click >>. The objects move to the **Selected objects** list. If a node on the tree does not contain objects within it, it is not moved.
 4. If you do not want to export all definitions of warehouse sources, tables, and columns that are related to exported steps, clear the **Export dependent source properties** check box.
 5. If you do not want to export schedule information related to the processes that you are exporting, clear the **Include schedules** check box.
 6. Click **OK**.

The Export Information window opens and displays a message that tells you whether the export process completed successfully. If errors occurred, the Export Metadata window remains open so that you can try the export procedure again. You can find more information about a particular error message in the *IBM DB2 Universal Database Message Reference*.

Several files can be created during a single export process. For example, when you export metadata definitions for BLOB data, multiple tag language files are created. The first file created in the export process has an extension of .tag. 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 tag language 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.

Importing metadata

You can import object definitions for use in your Data Warehouse Center system. You might want to import sample data into a warehouse or you might want to import data if you are creating a prototype for a new warehouse. You can import the following types of metadata:

- Interchange File
- Erwin
- i2
- MQSeries
- Trillium

Planning to import objects

When you import a tag language file, the metadata that it contains is stored in the warehouse control database. The Data Warehouse Center will accept a tag language file that is not generated by another Data Warehouse Center system. For example, you can import a tag language file that is generated by an ETI product.

Any object with an unknown OBJTYPE in a non-Data Warehouse Center tag language file will be ignored, and processing will continue. If an object is of a known OBJTYPE, but does not contain all the properties required by the Data Warehouse Center, an error will occur.

Initializing the control database: If you are using the import utility to establish a new Data Warehouse Center, you must initialize a new warehouse control database in the target system. After you complete this task, you can import as many tag language files as you want.

For more information about how to initialize a new control database, see “Changing the active warehouse control database” on page 337.

Importing from one Data Warehouse Center system to another: After you complete the export process, and a tag language file is created, the file is ready to import to a target system.

Before you import the file, consider the following issues:

- Naming considerations

If you are using the import utility to move a warehouse source from a test environment to a production environment, make sure that the production environment does not already have a warehouse source with the same warehouse source name unless you want to over write the definition of the warehouse source.

Maintaining Data Warehouse Center

If you import a step into a system that contains a step with the same name, then you must either delete the step that you want to overwrite, or change the step to development mode. Otherwise, the step cannot be updated, and an error will occur.

- Ordering the import of objects

You use a logical order to import objects. An object that is referred to but not defined in a tag language file must be defined in the destination warehouse control database. You can do this by first importing the object definition that is referred to in a separate tag language file, or you can define the object in the Data Warehouse Center tree view.

For example, you can import warehouse source definitions first. When the warehouse source definitions are ready for use, you can import subject areas, warehouse target definitions, and any corresponding steps individually.

- If you move a tag language file from one system to another, you must move all the associated files along with it and they must reside in the same directory. See “Exporting the metadata to a tag language file” on page 325 for more information on multiple files that can be created during the export process.

Do not use the import function to migrate from a previous version of the Data Warehouse Center. You must always import and export tag language files using the same version of the Data Warehouse Center. For information about migrating from a previous version of the Data Warehouse Center, see the *DB2 Universal Database Quick Beginnings* for your operating system.

Importing a tag language file into Data Warehouse Center

To import a tag language file into a Data Warehouse Center:

1. From the Data Warehouse Center, right-click on the Warehouse node, and click **Interchange File**.
The Import Metadata window opens.
2. In the **File name** field, type the name of the tag language file that you want to import. If you want to search for a file to import, click The File window opens.
3. Select the tag language file that you want to import, and click **OK**. The file name that you selected is displayed in the **File name** field of the Import Metadata window.
4. In the **Check-point ID** field, type the last check-point ID recorded in the log file if a previous import of the specified tag file failed. If there was no previous import of the tag file that failed, set the check-point ID field to 0.
5. Click **OK**. The metadata that is represented by the tag language file is imported into the warehouse control database.

After a successful import procedure, you receive a message with return code 0 or a warning return code, and the Import Metadata window closes. No changes are committed to the warehouse control database when an error occurs. Refresh the tree view to display the objects that you imported.

In addition to importing tag language files, you can import ERwin, MQSeries, Trillium, and i2 metadata.

- For information about importing ERwin metadata, see “Importing a tag language file into the Data Warehouse Center” on page 274.
- For information about importing MQSeries metadata, see “Importing MQSeries messages and XML metadata” on page 305.
- For information about importing Trillium metadata, see “Importing Trillium metadata” on page 263.
- For information about importing i2 metadata, see “Importing metadata for i2 sources” on page 441.

Import errors: If an error occurs during the import process, you receive an error message with an error return code and the Import Metadata window remains open.

Common errors include:

- Importing an object that is dependent on another object that is not present in your warehouse.
- Importing an object that already exists in the warehouse and has other objects dependent on it.
- Importing a step that already exists and is not in development mode.
- Importing a tag language file that was not exported properly.

Check the log file to determine what object or relationship was being imported when the error occurred. If the object is not listed in the message, it should be the last object that is listed before the return code and error message information at the end of the log file. Fix the error, then import the object definitions again. If you cannot determine the error or you cannot fix it, contact IBM Software Support or the vendor who provided the tag language file.

Post-import considerations

After a successful import, you receive a message with a return code of 0, and the Import Metadata window closes.

When you import metadata, all of the objects are assigned to the *default security group*. An imported subject area, warehouse source, or warehouse target will not be displayed in the tree view unless you belong to a warehouse group that has access to the object. A warehouse group that is associated with

Maintaining Data Warehouse Center

exported warehouse sources or warehouse targets must be related to a process that uses those objects; some warehouse groups might not be exported. If the warehouse groups that you want your warehouse sources and warehouse targets to be related to are not imported, you need to add the warehouse groups to the Data Warehouse Center system. Metadata for individual users is not exported. You must add users to the Data Warehouse Center.

The Data Warehouse Center assigns development mode to all imported steps so that you can verify that the steps are complete and accurate before you promote them to test or production mode.

Before you use imported objects: Update database information for warehouse sources and warehouse targets (database name, user ID, and password), if necessary. If any new agent site definitions were imported, their passwords also need to be updated. If you imported any programs that have password type parameters, you must reset those values.

Add your user ID to a warehouse group that has access to the imported processes and warehouse targets.

Before you run imported steps:

1. Ensure that all the warehouse source and warehouse target databases that are referred to by the imported steps are defined and configured in your Data Warehouse Center system. Verify all the definitions that you imported and change any parameters to match the new environment.
After you import a tag language file that includes steps that include joins, you might notice the tables are not placed as you expected in the Process Model or Warehouse Schema Model. Some information cannot be imported from one system to another. You will need to reposition your steps and star schemas using the Process or Warehouse Schema Model window.
2. Promote steps as necessary.

Importing tag language files again If you import an existing step again, it must be in development mode, or an error will occur. You must demote steps to development mode before importing them again.

Most tag language files can be imported more than once without errors. If a tag language file includes primary and foreign keys, the keys must be deleted before the file can be imported again.

If the names or order of columns in a table or file have changed (for example, a warehouse source or warehouse target table is redefined), delete the warehouse source table, file, or step definition (including the target table in

the warehouse target) from the target system before you import the same table or step again. At a minimum, verify that the column or field information is imported as you expected.

Publishing Data Warehouse Center metadata

Publishing metadata is the process of transferring metadata from the Data Warehouse Center to the Information Catalog Manager. An *information catalog* is the set of tables managed by the Information Catalog Manager that contains business metadata that helps users identify and locate data and information available to them in the organization. When you publish metadata to the information catalog, you provide end users and system analysts with a fast path to seeing what is in the warehouse.

From the Data Warehouse Center, you can publish:

- Metadata that describes Data Warehouse Center objects.
- Metadata that describes DB2 OLAP Server or Hyperion Essbase Server objects.

This chapter discusses publishing Data Warehouse Center metadata only. For information about publishing OLAP server metadata, see the *Information Catalog Manager Administration Guide*.

To publish metadata, you must install the DB2 Warehouse Manager package with the Information Catalog Manager. The Information Catalog Manager runs on Windows 32-bit operating systems. For information about the DB2 Universal Database platforms on which an information catalog can reside, see the *Information Catalog Manager Administration Guide*.

Planning to publish metadata

Before you publish metadata, you need to ensure that the correct environment is established.

1. Ensure that you installed and configured the necessary warehouse components on the correct workstations:

The Information Catalog Manager administrator function must be installed both on the warehouse server site and on the Data Warehouse Center administrative interface component if they are on different workstations.

The Data Warehouse Center administrative interface must have DB2 connectivity to the information catalog APIs.

2. Ensure that both the administrator user IDs for the Information Catalog Manager and Data Warehouse Center have Windows NT administrator privileges.

On AIX and the Solaris Operating Environment, the installation process sets the language to publish for the information catalog, and export to the OLAP

Maintaining Data Warehouse Center

Integration Server. If you want to use these functions in a language other than the language set during installation, create the following soft link by entering the following command on one line:

On AIX

```
/usr/bin/ln -sf /usr/lpp/db2_07_01/msg/locale/flgnxolv.str  
/usr/lpp/db2_07_01/bin/flgnxolv.str
```

locale The locale name of the language in xx_yy format

On the Solaris Operating Environment

```
/usr/bin/ln -sf /opt/IBMdb2/V7.1/msg/locale/flgnxolv.str  
/opt/IBMdb2/V7.1/bin/flgnxolv.str
```

locale The locale name of the language in xx_yy format

Selecting metadata to publish

Table 38 provides the mapping between object types in the Data Warehouse Center and information catalog. The Data Warehouse Center uses this mapping when you export the metadata to the information catalog. See “Appendix B. Metadata mappings” on page 363 for a detailed mapping of object types and object type properties.

Table 38. Mapping between Data Warehouse Center and information catalog object types

Data Warehouse Center object type	Information catalog object type
Process	DWC Process
Step	Transformation (at the table or column level)
Column or field	Columns or fields
Warehouse source or warehouse target	Databases, IMS database definitions
Subject area	Business subject areas
Table or file	Relational tables and views, Files, IMS segments
Warehouse schema	Star Schema

When you publish a process, the following objects are published with it:

- All steps that are contained within a process. You cannot publish individual steps.
- Any user-defined programs that are associated with steps in the process.
- Schedule information for processes and steps.
- Any warehouse source or warehouse target definitions that are used by the process and underlying steps.

- Any other processes and steps whose output is used by the process that is being published.
You can limit the number of underlying processes that are published by selecting the **Limit the levels of objects in the tree** option in the Publish Data Warehouse Center Metadata notebook.
- All table definitions within a warehouse schema, if you publish a warehouse schema.
- All table definitions in a selected warehouse source or warehouse target are published.

Requirement: Before you publish metadata, make sure that you are not logged on to the Information Catalog Manager as an administrator.

To select the metadata that you want to publish:

1. From the Data Warehouse Center, right-click on the Warehouse node, and click **Publish metadata**. The Publish Metadata window opens.
2. Right-click on the white space in the list, and click **Data Warehouse Center metadata**. The Publish Data Warehouse Center Metadata notebook opens.
3. In the **Available objects** list, select the objects that you want to publish, and click **>**
The object moves to the **Selected objects** list. Repeat this step for all the objects that you want to publish.
If you want to publish all available objects, click **>>**. If a node on the tree does not contain objects within it, it is not moved.
4. Click the **Synchronization Settings** tab.
5. In the **Catalog name** field, type the name of the information catalog to which you want to publish metadata.
6. In the **Administrator user ID** field, type a user ID that has access to the information catalog. The user ID can be the ID of the information catalog administrator or that of an information catalog user who has certain administrative privileges.
7. In the **Administrator password** field, type the password that corresponds to the user ID.
8. Type the password again in the **Verify password** field.
9. Select one of the options from the **Map source tables to target warehouse** list to specify how you want to map processes and their source tables when they are published in the information catalog:
 - Click **At the table level** to generate an information catalog transformation object between an information catalog source table and a target table. In the Information Catalog Manager, you will see the

Maintaining Data Warehouse Center

target table, the transformation object, and the source table when you open the Tree view for a database object. This option is selected by default.

- Click **At the column level** to generate an information catalog transformation object between an information catalog source column and a target column. When you open the Tree view for a database object, you will see the target table, target column, the transformation object, and source table and source column.

You can map processes and their source tables at the column level to get a more detailed view of the data.

10. Optional: Select one of the following check boxes to determine how published metadata will be updated:
 - Select **Generate object names and descriptions in the information catalog on update** to replace existing object names and descriptions with the names and descriptions of the same objects in the Data Warehouse Center. Do not select this check box if you want to keep existing objects and descriptions in the information catalog. For example, if you specified additional business information for an object in the information catalog and do not want to overwrite the information with updated values, do not select this check box. This check box is not available if you select **Delete existing objects in information catalog before updating**.
 - Select **Delete existing objects in information catalog before updating** to delete existing objects before you publish updated objects from the Data Warehouse Center. Do not select this check box if you want to keep and update existing objects in the information catalog. If you select this check box, the Information Catalog Manager deletes all objects that are related to a deleted object, such as database and table objects. The tree view is updated so that there are no objects that are unrelated to other objects.
11. Optional: You can make the tree structure shown in the Information Catalog Manager less detailed by limiting the number of levels of objects published and displayed in the Tree view:
 - a. Select **Limit the levels of objects in the tree** to limit the number of levels of objects that are published and displayed in the Tree view. If you do not select this check box, all objects under the selected object are published and displayed. If you specify a limit, only a subset of the objects under a selected object is published and displayed.
 - b. Click the up or down arrow to specify a limit for the number of levels of objects that are published and displayed in the tree view. The default limit value is 1. The lower the limit that you specify, the less detailed the tree structure will be.

If you set a limit, the number of levels of objects that are published and displayed in the tree view is limited to the number of levels that you specify.

12. Click **OK**.

The Publish Information window opens and displays a message that tells you whether the publish process completed successfully. The Publication Information window displays the contents of a log file that lists which metadata objects were published successfully. If errors occurred, the notebook remains open when you close the Publication Information window so that you can try to publish objects again. You can find more information about a particular error message in the *IBM DB2 Universal Database Message Reference*.

If the publication is successful, the notebook closes when you close the Publication Information window, and two publication objects are added to the list of publications in the Publish Metadata window. One publication updates control metadata, the other updates definition metadata. *Control metadata* is metadata that describes changes to data in the warehouse. Examples of control metadata are the date and time that a table is updated by the processing of a step. *Definition metadata* is metadata that describes the objects in the warehouse, the sources of data, and the transformations that are applied to the data. Examples of definition metadata are column names, table names, and database names. The first time that you publish metadata, both control and definition metadata is registered in the information catalog. When you update metadata, you can choose to update the control metadata, the definition metadata, or both types.

Updating published metadata

After you publish metadata, you can use the publication objects (created during the publish process) periodically to propagate to the information catalog the changes that are made in the Data Warehouse Center. This process is called *metadata synchronization*.

When you synchronize metadata, the metadata for an object that is registered in the information catalog is updated either when you run the publication or based on a schedule that you create for the publication. Metadata is not updated in the information catalog in the following situations:

- When a new object is created in the Data Warehouse Center. If you add a new object to the Data Warehouse Center, you must publish the object first. The update process only updates objects that were previously published.
- When the name of an object that you previously published to the information catalog changes

Maintaining Data Warehouse Center

Requirement: If you plan to synchronize metadata, you must use the Data Warehouse Center warehouse control database as your information catalog database.

To update published metadata:

1. In the **List of Publications** in the Publish Metadata window, right-click on the publication that you want to run.
2. Click **Run**.

Scheduling regular updates of published metadata

You can automate updates to previously published metadata by creating a schedule.

To schedule regular updates of metadata to the information catalog:

1. Open the Publish Metadata window.
2. In the **List of publications**, right-click the publication that you want to schedule, and click **Schedule**. The Schedule page of the Schedule notebook opens so that you can create a schedule on which to run the publication.

See “Scheduling warehouse processes” on page 157 for information about using the Schedule notebook.

Viewing log files for the publication updates

You can view the log files that show the result of the update to the publication. The files are located in the directory that is specified by the VWS_LOGGING environment variable. The default value of the VWS_LOGGING variable for Windows NT is \sql1ib\logging, where x is the drive where the DB2 Universal Database is installed. The following log file names are used:

Type of metadata	Purpose	Log file name
Update of Data Warehouse Center control metadata	Describes changes to objects in the warehouse	ICMDWCD.OUT
Update of Data Warehouse Center definition metadata	Describes objects in the warehouse, the sources of data, and the transformations that are applied to the data	ICMXCHG.OUT

Propagating deleted objects to the information catalog

When an object is deleted in Data Warehouse Center, information about the deleted object is stored in the warehouse control database. When the metadata is updated during the synchronization process, the Data Warehouse Center propagates these deletions to the information catalog before importing other changes into the information catalog. When metadata synchronization

completes successfully, the Data Warehouse Center removes the entries in the warehouse control database. Because the Data Warehouse Center removes the entries, deletions can be propagated to only one information catalog. If you need to make the deletions to a second information catalog, you must delete those items manually.

If you change the name of a warehouse object that you previously published to the information catalog, you must publish the object again to update the information catalog. The object with the old name is not overwritten, so both objects exist in the information catalog after metadata synchronization; you must delete the old object manually.

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.

To initialize a new warehouse control database:

1. Stop the warehouse server. See “Starting the warehouse server and the logger” on page 7 for information about how to start and stop the warehouse server.

Maintaining Data Warehouse Center

2. Click **Start** → **Programs** → **IBM DB2** → **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. Restart the warehouse server.
11. Start the Data Warehouse Center administrative interface.
12. Click **Advanced** in the Data Warehouse Center Logon window, and type the name of the new warehouse control database in the **Control database field**.
13. 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 perform a custom installation and change the name in the Define a Local Warehouse Control Database window. 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.

Note: If you apply a fixpak or install a new release of DB2 or the DWC, you must migrate the existing control database to update the objects that it contains.

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 on page 337. 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.

Configuring the Data Warehouse Center

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.

Changing the Data Warehouse Center properties

Use the Data Warehouse Center Properties notebook to change the Data Warehouse Center configuration parameters. For example, you can change the parameters for the default notification messages to suit your particular business situation.

You must have administrative privilege to change Data Warehouse Center properties.

To change the Data Warehouse Center properties:

Open the Data Warehouse Center Properties notebook:

1. Right-click the **Warehouse** object, and select **Properties** from the pop-up menu.
2. Change any of the following settings:
 - On the Trace Level page:
 - Control database
 - ODBC
 - Server
 - Agent
 - Logger
 - On the Server page:
 - Agent start/stop timeout
 - Purge log when total records equal
 - Restart interrupted jobs
 - On the Processing Options page:
 - Retry
 - No rows returned

Maintaining Data Warehouse Center

- SQL warning
 - On the Notification Messages page:
 - Condition
 - Sender
 - Subject
 - System parameters
 - Message
 - Default mail server
- The mail server must support ESMTP.

3. Click **OK** to save your changes.

See the online help for the Data Warehouse Center Properties notebook for detailed descriptions of the settings in the notebook.

You can also use the Data Warehouse Center Properties notebook to run a Data Warehouse Center component trace. For more information, see “Component trace data” on page 358.

Chapter 15. 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. 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*.

Table 39 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.

Table 39. Tasks to create a star schema and populate a multidimensional cube

Task	Complete task from	
	Data Warehouse Center	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	

Table 39. Tasks to create a star schema and populate a multidimensional cube (continued)

Task	Complete task from	
	Data Warehouse Center	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 automatically generated.		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).

Defining the warehouse schema

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.

See “Chapter 4. Setting up access to a warehouse” on page 105 for more information about defining a warehouse target.

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. See “Data Warehouse Center security” on page 19 for more information.

To define a warehouse schema:

1. From the Data Warehouse Center, right-click the **Warehouse Schemas** folder.
2. Click **Define**.
The Define Warehouse Schema notebook opens.
3. In the **Name** field, type the schema name.
4. Optional: In the **Administrator** field, type the name of a contact for the warehouse schema.
5. Optional: In the **Description** field, type a description of the warehouse schema. If you publish the metadata for the warehouse schema to an information catalog, the information in this field is used as a value for the Short description property for the warehouse schema.
6. Optional: In the **Notes** field, type any notes that you want to add. If you publish the metadata for the warehouse schema to an information catalog, the information in this field is used as a value for the Long description property for the warehouse schema.

7. Optional: To show tables in only one database, select the **Use only one database** check box, and select the database name from the **Warehouse Target Database** list.

Only those warehouse schemas that are made up of tables from one database can be exported to the DB2 OLAP Integration Server.

8. Click **OK** to define the warehouse schema.

The new warehouse schema is added to the tree under the **Warehouse Schemas** folder.

After you define your warehouse schema, you can update it by opening the Warehouse Schema Modeler window.

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 as described in “Exporting a warehouse schema to the DB2 OLAP Integration Server” on page 346.

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.
 - 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.

Autojoining tables

Use the Warehouse Schema Model window to autojoin tables. Selected tables are autojoined if the warehouse primary key and warehouse foreign key are defined and if you click **Autojoin**. For more information about defining keys on target tables, see “Defining a warehouse target” on page 115.

1. Expand the object tree until you find the **Warehouse Schemas** folder.
2. Right-click the warehouse schema, and click **Open**. The Warehouse Schema Modeler window opens.
3. Select the tables that you want to autojoin holding down the Ctrl key and clicking each table.

The primary key for each table is shown with an icon for the columns that are part of the key. Use the **View** menu to hide or show columns on the tables. Columns must be visible in both tables to create the link. All tables do not need to be in the same state when you select them in a group.

4. Click the **Autojoin** icon on the toolbar, or click **Warehouse schema** → **Autojoin**. Primary and foreign key relationships are shown in green.
5. Click the **Save icon** on the toolbar, or click **Warehouse schema** → **Save** to save your work.

Adding join relationships between non-keyed columns

Use the Warehouse Schema Model window to add join relationships between non-keyed columns in tables that are displayed on the canvas. You can join any two columns (multi-column joins are not allowed).

1. Expand the object tree until you find the **Warehouse Schemas** folder.
2. Right-click the warehouse schema, and click **Open**. The Warehouse Schema Modeler window opens. Ensure that there are at least two tables defined on the canvas.
3. Click the **Join** icon on the toolbar, and then click the column in the first table, while holding the mouse button down.
4. Move the cursor to the column of the second table to establish the join relationship. The join relationship lines are shown in black.
5. Click the **Save icon** on the toolbar, or click **Warehouse schema** → **Save** to save the join relationships as part of the warehouse schema.

Figure 22 on page 346 shows what your warehouse schema might look like after you have defined it.

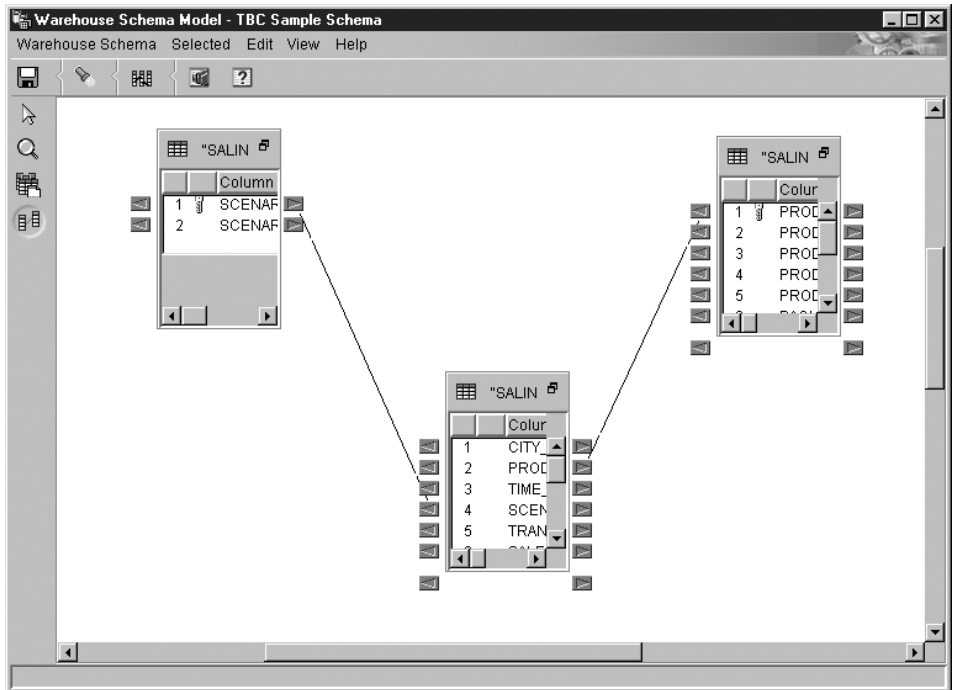


Figure 22. Warehouse Schema Model window

Exporting a warehouse schema to the DB2 OLAP Integration Server

Use the Export Warehouse Schema notebook to export warehouse schemas to the DB2 OLAP Integration Server (as OLAP models).

To export the warehouse schema:

1. From the Data Warehouse Center, right-click the **Warehouse** node and click **Export Metadata** → **OLAP Integration Server**.
The Export Warehouse Schema notebook opens.
2. From the **Available** list, select the warehouse schemas that you want to export. Only warehouse schemas defined with the **Use only one database** option are available to export.
3. Click **>**.
The selected schemas move to the **Selected** list.
4. Click the **Integration Server** tab.

5. In the **Catalog name** field, type the name of the DB2 OLAP Integration Server catalog to which you want to export the warehouse schema. The default name is the last catalog that was used to export warehouse schemas.
6. In the **Catalog table schema name** field, type the name of the catalog table schema that is used by the DB2 OLAP Integration Server catalog.
7. Optional: In the **Time dimension table** field, select the name of the Time dimension table. If you do not select a name, or if there are no tables in the list, the Regular dimension type will be used. If more than one schema is selected to be exported, this field is available only when the selected schemas have common dimension tables.
You cannot change the dimension type after the schema is stored in the DB2 OLAP Integration Server catalog.
8. Optional: In the **Accounts dimension table** field, select the name of the Accounts dimension table. If you do not select a name, or if there are no tables in the list, the Regular dimension type will be used. If multiple schemas are selected, this field is available only when the selected schemas have common dimension tables.
You cannot change the dimension type after the schema is stored in the DB2 OLAP Integration Server catalog.
9. In the **User ID** field, type the user ID that you will use to access the DB2 OLAP Integration Server catalog.
10. In the **Password** field, type the password for the user ID.
11. In the **Verify password** field, type the password again.
12. Select one of the following options to specify the action you want to take if the OLAP model already exists in the DB2 OLAP Integration Server catalog.
 - Click **Add the new data to the existing OLAP model**, if the OLAP model name is the same as the warehouse schema name.
If you select this option, only new information from the warehouse schema is added to the existing OLAP model. Existing information is not modified. For example, the fact table remains the same and you cannot change any dimensions if you select this option. You can add new dimensions.
 - Click **Display a message and not perform an export**, if the OLAP model name is the same as the warehouse schema name.
If you select this option, an error message is displayed and the existing model is not modified. This option is selected by default.
 - Click **Overwrite the existing model**, if the OLAP model name is the same as the warehouse schema name.

If you select this option, all the existing metadata about the OLAP model is deleted and replaced with the new metadata from the warehouse schema.

13. Click **OK** to export the selected warehouse schemas to the DB2 OLAP Integration Servercatalog.

The notebook closes, and a progress indicator is displayed until the export is complete. When all the specified warehouse schemas are exported, the Export Information window opens, displaying success information or failure information about the export. Click **OK** to close this window.

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 `\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

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.

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. Figure 23 on page 349 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.

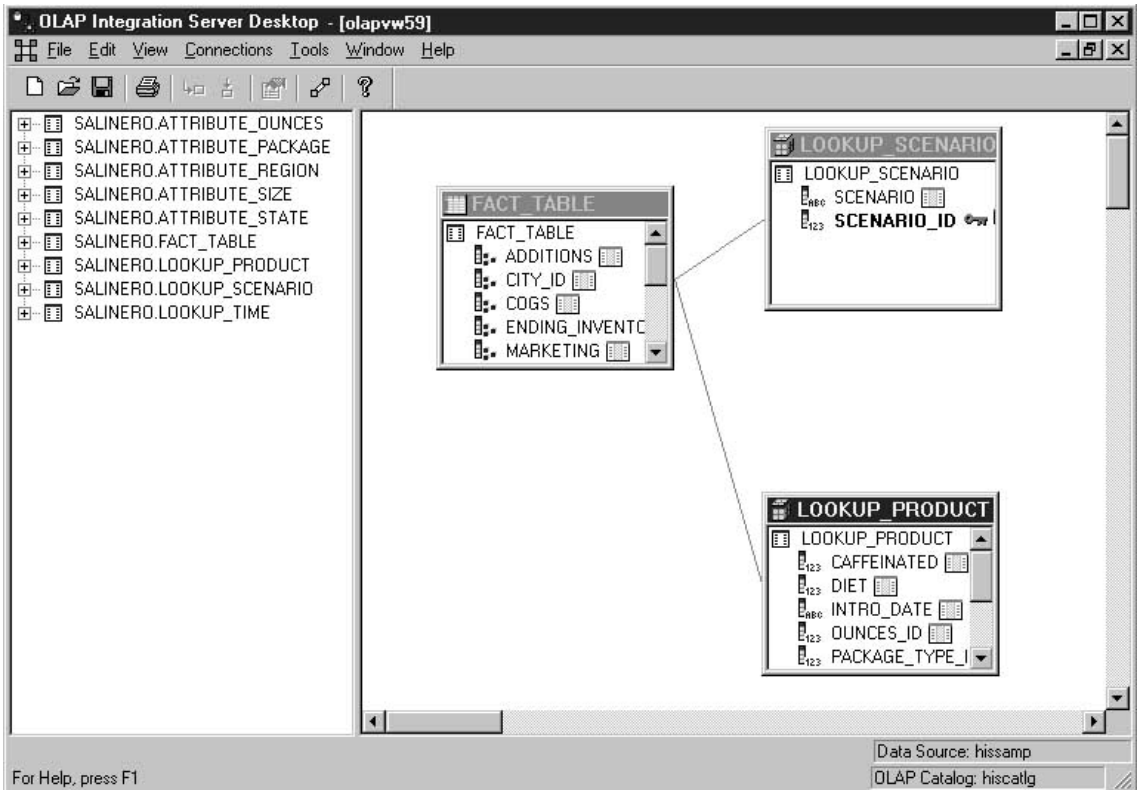


Figure 23. Warehouse schema displayed as an OLAP model

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

1. 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 Model and User's Guide*.
2. 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. See "Creating an outline and loading the data for the multidimensional cube in the DB2 OLAP Integration Server" on page 350 for more information.

3. 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. See “Exporting the metadata to the Data Warehouse Center” on page 352.

Creating an outline and loading the data for the multidimensional cube in the DB2 OLAP Integration Server

This section describes how to create an outline and associate it with a 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.

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

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.

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.

Figure 24 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"
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

```

Figure 24. OLAP command script: my_script.script

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.

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 Figure 25 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"

```

Figure 25. Bat file that invokes command script: my_script.bat

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 DB2 OLAP Integration Server Administration Manager 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.

Before you export the metadata, make certain you have already defined the tables for your warehouse schema as described in “Designing the warehouse schema in the Data Warehouse Center” on page 343.

To export metadata to the Data Warehouse Center, start from the DB2 OLAP Integration Server desktop:

1. Click **Tools** → **Administration Manager**. The Administration Manager window opens.

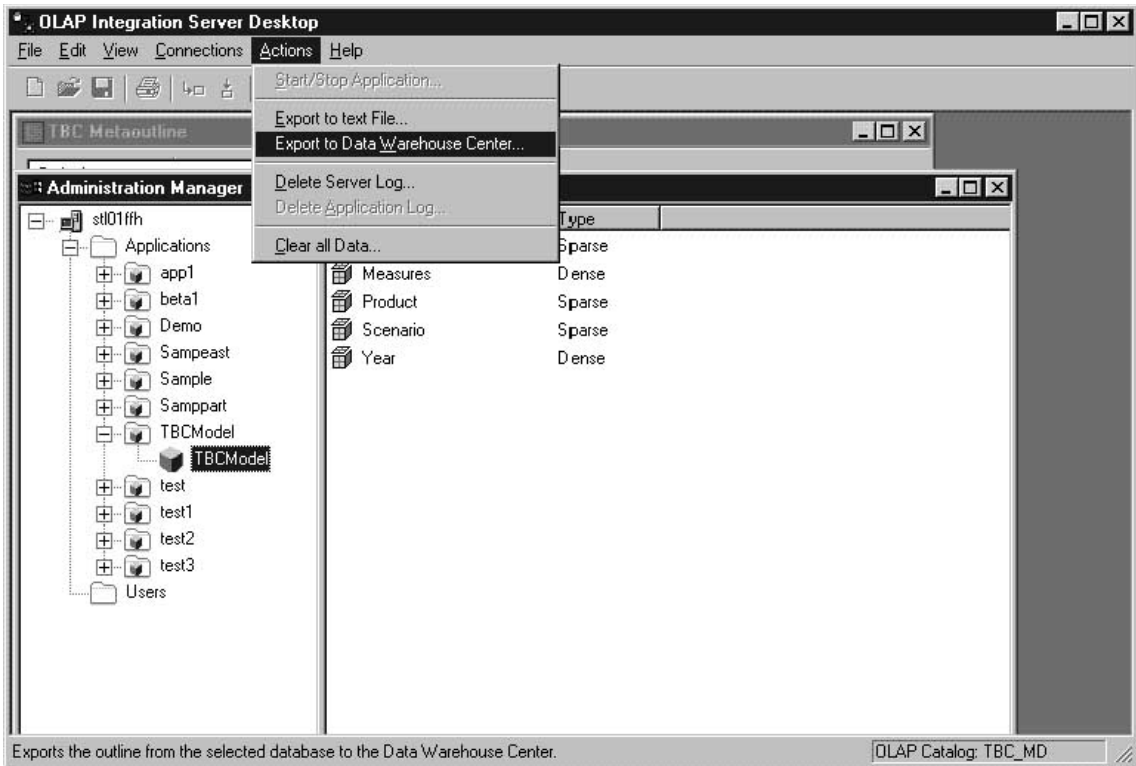


Figure 26. Administration Manager window

2. Expand the **Applications** folder, and right-click the database whose metadata you want to export.

3. Click **Actions** —> **Export to Data Warehouse Center**. The Export to Data Warehouse Center window opens.
4. In the **OLAP Model Name** field, type the name of the model (warehouse schema) on which your metaoutline is based.
5. In the **Catalog DSN** field, type the name of the catalog database source name in the DB2 OLAP Integration Server that contains the model on which your metaoutline is based.
6. In the **User Name** field, type the user ID that you use to connect to the catalog database.
7. In the **Password** field, type the password for the user ID.
8. In the **Table Schema** field, type the table schema that is used for tables in DB2 OLAP Integration Server catalog.
9. In the **Control Database** field, type the name of the warehouse control database to which you want to export the metadata.
10. In the **User Name** field (for Data Warehouse Center), type the user ID that you use to connect to the warehouse control database.
11. In the **Password** field, type the password for the user ID.
12. In the **Table Schema** field, type the table schema that you use for the tables in your warehouse schema.
13. In the **Batch file to invoke OLAP Integration Server script** field, type the fully qualified name for the batch file that runs the command script to load the multidimensional cube.
For example, type `c:\is\batch\my_script.bat`.
14. Click **OK**.

The metadata for the batch file is exported to the Data Warehouse Center.

See the log file for information about the metadata that is exported to the Data Warehouse Center.

See “Loading the multidimensional cube from the Data Warehouse Center” for information about the objects that are created in the Data Warehouse Center.

Loading the multidimensional cube from the Data Warehouse Center

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. The value in the **Parameter value** column matches the batch file you specified in the **Batch file to invoke OLAP Integration Server script** field of the DB2 OLAP Integration Server Export to Data Warehouse Center window. 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).

See "Metadata mappings between the DB2 OLAP Integration Server and the Data Warehouse Center" on page 375 for detailed information on how DB2 OLAP Integration Server metadata maps to Data Warehouse Center metadata.

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. 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 (described in “Designing the warehouse schema in the Data Warehouse Center” on page 343). For more information about scheduling a step to run based on the success or failure of another step, see “Scheduling warehouse processes” on page 157.

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. Figure 27 shows the Work in Progress window as a multidimensional cube is being populated.

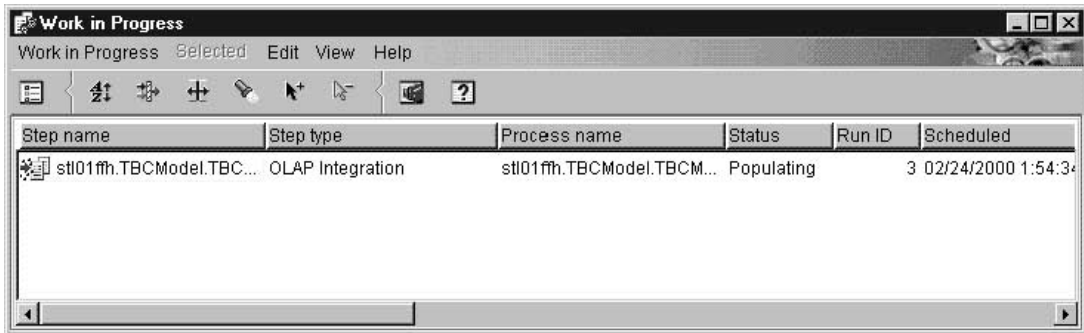


Figure 27. Work in Progress window with a step that is populating a cube

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. See “Publishing Data Warehouse Center metadata” on page 331 for more information.

A warehouse schema maps to a star schema in the Information Catalog Manager. You use a command line interface to publish DB2 OLAP Integration Server metadata to the information catalog. When you publish DB2 OLAP Integration Server metadata objects, a linked relationship is created between a “dimensions within a multi-dimensional database” object type and a table object in the information catalog. If you publish your warehouse schema to the information catalog and publish the DB2 OLAP Integration Server metadata, you will have a complete depiction of the metadata for the OLAP model. See the *Information Catalog Manager Administration Guide* for more information about publishing DB2 OLAP Integration Server metadata.

Appendix A. 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 appendix 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.

To view run-time (step processing) errors:

1. Open the Data Warehouse Center desktop.
2. Click **Data Warehouse Center** → **Work in Progress**.
The Operations Work in Progress window opens.
3. Select the step for which you want to view errors.
4. Click **Log**.

The Log Viewer window opens and displays the run-time errors for the selected step.

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**.

The Log Viewer window opens and displays the build-time errors for the Data Warehouse Center.

For detailed explanations of return codes and error messages that the log viewer displays, see *DB2 Messages and Reason Codes*.

Data Warehouse Center logging and trace data

The log holds records until a designated count limit is reached. When the count limit is reached, the Data Warehouse Center automatically deletes the log.

To change the count limit for the log, see the online help for the Data Warehouse Center Properties notebook.

Note: 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.

You can also view error messages with the Windows NT Application Events, which is available through the Event Viewer application.

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 AS/400 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. For more information, see the *Warehouse Manager Installation Guide*.

The default value of *VWS_LOGGING* is:

The component trace data function provides information about the following components:

Windows NT and OS/2

x:\program files\sqllib\logging

UNIX and OS/390

/var/IWH

AS/400

/QIBM/UserData/IWH

The Data Warehouse Center writes these files on Windows NT:

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.

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.

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.

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.

Data Warehouse Center logging and trace data

For more information, see the online help for the specific warehouse program.

Transformers

You can log errors that are generated when you use the transformers. To enable logging, specify a log table name on the Processing Options page of the Step notebook, 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.

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.

Apply program

You can trace errors that are generated when you use the Apply program. 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.

Data Warehouse Center logging and trace data

Appendix B. Metadata mappings

This appendix lists object types and object type properties for the following metadata:

- Information Catalog Manager metadata to Data Warehouse Center metadata, described in “Metadata Mappings between the Information Catalog Manager and the Data Warehouse Center” below.
- Information Catalog Manager metadata to OLAP server metadata, described in “Metadata mappings between the Information Catalog Manager and OLAP server” on page 373.
- Data Warehouse Center metadata to DB2 OLAP Integration Server metadata described in “Metadata mappings between the DB2 OLAP Integration Server and the Data Warehouse Center” on page 375.

Metadata Mappings between the Information Catalog Manager and the Data Warehouse Center

The following tables show the metadata mappings between the Information Catalog Manager and the Data Warehouse Center for each object type. The Information Catalog Manager column shows object type properties as they are displayed in the Description view for an object. The Data Warehouse Center column shows the names of object properties as they are displayed in the various object property notebooks. In some cases, Data Warehouse Center property information (such as processing time stamps for steps) is taken from the Work in Progress window.

Table 40. Metadata mappings between the Information Catalog Manager and the Data Warehouse Center for database objects

Information Catalog Manager metadata	Data Warehouse Center metadata
Name	Warehouse source or Warehouse target name
Short description	Description
Long description	Notes
URL to access data	N/A
Actions	N/A
Database or subsystem name	Database name

Table 40. Metadata mappings between the Information Catalog Manager and the Data Warehouse Center for database objects (continued)

Database type	<p>The value for this property can be either RELATIONAL or FILE.</p> <p>The mapping is derived from the warehouse source or warehouse target type.</p>
Agent type	N/A
Database location	N/A
Database host server name	System name
System code page	This metadata is internal to the Data Warehouse Center. ¹
Database server type and database extended type.	<p>The mapping is derived from the warehouse source or warehouse target type.</p> <p>For example, if your warehouse target is a DB2 Universal Database for Windows NT database, the database server type is DB2 Family. The database extended type is DB2 NT.</p>
Database owner	N/A
Timestamp source definition last changed	<p>Last update time stamp for the database definition.</p> <p>This metadata is internal to the Data Warehouse Center.¹</p>
Database status	N/A
Database extended type	<p>Database subtype and database version.</p> <p>The mapping is derived from the warehouse source or warehouse target type. For example, if your warehouse target is a DB2 Universal Database for Windows NT database, the database extended type is DB2 NT.</p>
Timestamp source definition created	N/A
For further information	Administrator
Note:	
<p>1. If a property is specified as internal to the Data Warehouse Center, it is not displayed.</p>	

Table 41. Metadata mappings between the Information Catalog Manager and the Data Warehouse Center for IMS DBD (database description definition) objects

Information Catalog Manager metadata	Data Warehouse Center metadata
Name	Warehouse source name
Short description	Description
Long description	Notes
Actions	N/A
Database last refreshed	N/A
For further information	Administrator
Database owner	N/A
Database host server name	System name
Database server type	Database type and database version. The mapping is derived from the warehouse source type. The property value for IMS DBDs is IMS.
Database or subsystem name	Datasource name
Database type	This property is set to HIERARCHICAL. The mapping is derived from the warehouse source type.
Database extended type	Database subtype and database version. The mapping is derived from the warehouse source type. The property value for IMS DBDs is IMS.
Database status	N/A
IMS access method	N/A
Operating system access method	N/A
Shared index names	N/A
URL to access data	N/A
Timestamp source definition created	N/A
Timestamp source definition last changed	Last update time stamp for the IMS DBD. This metadata is internal to the Data Warehouse Center. ¹
Note:	
1. If a property is specified as internal to the Data Warehouse Center, it is not displayed.	

Table 42. Metadata mappings between the Information Catalog Manager and the Data Warehouse Center for relational table or view objects

Information Catalog Manager metadata	Data Warehouse Center metadata
Name	Table name
Short description	Description
Long description	N/A
URL to access data	N/A
Actions	N/A
Catalog remarks	N/A
Local database alias	N/A
Table data last refreshed	Last completed time stamp for a step that ran and used the table as a target table. This information is displayed in the Work in Progress window.
Transformation program type	The value for this property is Data Warehouse Center. There is no specific metadata in the Data Warehouse Center for this property.
Database or subsystem name	Database name of the warehouse source or warehouse target database that contains the table
Table owner	Table schema
Table name	Table name
Timestamp source definition last changed	Last update time stamp for a table definition. This metadata is internal to the Data Warehouse Center. ¹
Base table owner name	N/A
Base table name	N/A
Transformation program run mode	N/A
Transformation program last run	N/A
Transformation program run frequency	N/A
Partial or full table copy/update	N/A
Copied/updated data is in a consistent state	N/A
Catalog refresh/update frequency	N/A
Transformation program last changed	N/A

Table 42. Metadata mappings between the Information Catalog Manager and the Data Warehouse Center for relational table or view objects (continued)

Transformation program compiled	N/A
Table type	The mapping is derived from the warehouse source or warehouse target subtype of the database that contains the table. For example, if your warehouse source or target is a DB2 Universal Database for Windows NT database, the database table type is DB2 NT.
Definition represents a view	N/A
Internal IDS name of table	N/A
Table is used as a dimension table	Dimension table
Database host server name	System name of the warehouse source or warehouse target database that contains the table.
Time stamp source definition created	N/A
For further information	Administrator of the warehouse source or warehouse target database that contains the table.
Note:	
1. If a property is specified as internal to the Data Warehouse Center, it is not displayed.	

Table 43. Metadata mappings between the Information Catalog Manager and the Data Warehouse Center for column or field objects

Information Catalog Manager metadata	Data Warehouse Center metadata
Name	Column or field name
Short description	Description
Long description	N/A
URL to access data	N/A
Actions	N/A
Catalog remarks	N/A
Datatype of column or field	Data type
Position of column or field in the primary key	N/A
Length of the column or field	Length or precision (depending on the data type)

Table 43. Metadata mappings between the Information Catalog Manager and the Data Warehouse Center for column or field objects (continued)

Scale of the column or field	Scale
Can column or field be null	Nullable
Column or field position	Position in the list of columns or fields displayed in the table or file notebook for a warehouse source or warehouse target.
Database or subsystem name	Database name of the warehouse source or warehouse target that contains the table that contains the column.
Table owner	Table schema of the table that contains the column.
Table name	Name of the table that contains the column.
Containing dimension	N/A
Column or field name	Column name
File name	File name of the file that contains the field (Data Warehouse Center files only)
Byte offset of column or field from start	Offset of this field in a file of fixed type. This metadata is internal to the Data Warehouse Center. ¹
Is column or field part of a key	N/A
Is column or field a unique key	N/A
Is data a before or after image, or computed	N/A
Source column/field name or expression used to populate column	N/A
Timestamp source definition last changed	Last update time stamp for the column definition. This metadata is internal to the Data Warehouse Center. ¹
String used to represent null values	N/A
Resolutions of dates	N/A
Precision of column or field	N/A
Is data text	Is text The value for this property is Y or N.

Table 43. Metadata mappings between the Information Catalog Manager and the Data Warehouse Center for column or field objects (continued)

Database host server name	System name of the database that contains the table that contains the column.
Column or field last refreshed	N/A
Timestamp source definition created	N/A
For further information	Administrator for the database that contains the table that contains the column.
Column ordinality	N/A
Note:	
1. If a property is specified as internal to the Data Warehouse Center, it is not displayed.	

Table 44. Metadata mappings between the Information Catalog Manager and the Data Warehouse Center for file objects

Information Catalog Manager metadata	Data Warehouse Center metadata
Name	The value for this property is derived from the file name.
Short description	Description
Long description	N/A
URL to access data	N/A
Actions	N/A
Information last refreshed	Last completed time stamp for a step that ran and used the file as a target file.
Transformation program type	The value for this property is Data Warehouse Center. There is no specific metadata in the Data Warehouse Center for this property.
Database host server name	System name of the warehouse source or warehouse target that contains the file.
Database or subsystem name	Database name of the warehouse source or warehouse target that contains the file.
File owner	N/A
File path or directory	The property value for the file path or directory is derived from the file name.
File name	The property value is derived from the file name.

Table 44. Metadata mappings between the Information Catalog Manager and the Data Warehouse Center for file objects (continued)

File class or type	File type
Source definition last changed	Last update time stamp for a file definition. This metadata is internal to the Data Warehouse Center. ¹
Transformation program last run	N/A
Transformation program run frequency	N/A
Partial or full file copy/update	N/A
Copied/updated data is in a consistent state	N/A
Transformation program last changed	N/A
Transformation program last compiled	N/A
Timestamp source definition created	N/A
For further information	Administrator of the warehouse source or warehouse target that contains the file.
Note:	
1. If a property is specified as internal to the Data Warehouse Center, it is not displayed.	

Table 45. Metadata mappings between the Information Catalog Manager and the Data Warehouse Center for IMS segment objects

Information Catalog Manager metadata	Data Warehouse Center metadata
Name	Table name
Short description	Description
Long description	N/A
URL to access data	N/A
Actions	N/A
Database or subsystem name	Datasource name
Segment name	N/A
Segment maximum length	N/A
Segment minimum length	N/A
Real logical child segment source	N/A
Logical parent concatenated key source	N/A
Transformation program last run	N/A

Table 45. Metadata mappings between the Information Catalog Manager and the Data Warehouse Center for IMS segment objects (continued)

Transformation program run frequency	N/A
Timestamp source definition last changed	Last update time stamp for a segment definition. This metadata is internal to the Data Warehouse Center. ¹
Database host server name	System name for the IMS database definition (DBD)
Segment owner	N/A
Segment last refreshed	N/A
Timestamp source definition created	N/A
For further information	Administrator for the IMS DBD that contains the segment.
Note:	
1. If a property is specified as internal to the Data Warehouse Center, it is not displayed.	

Table 46. Metadata mappings between the Information Catalog Manager and the Data Warehouse Center for transformation objects

Information Catalog Manager metadata	Data Warehouse Center metadata
Name	Step name
Short description	Description
Long description	N/A
URL to access data	N/A
Actions	N/A
Transformation Identifier	Unique identifier for the transformation. This metadata is internal to the Data Warehouse Center. ¹
Transformation program name	Program name
Transformation class or type	Program type
Source column/field name, expression or parameters	For SQL steps, the value of this property is SQL statement. For non-SQL steps, the value is the concatenation of any Parameter values for the step.

Table 46. Metadata mappings between the Information Catalog Manager and the Data Warehouse Center for transformation objects (continued)

Source definition last changed	Last update time stamp for step definition. This metadata is internal to the Data Warehouse Center. ¹
Database host server name	Target database system name
Transformation owner	N/A
Source sequence	N/A
Transformation ordinality	N/A
Transformation bidirectionality	N/A
Timestamp source definition created	N/A
For further information	Administrator
Note:	
1. If a property is specified as internal to the Data Warehouse Center, it is not displayed.	

Table 47. Metadata mappings between the Information Catalog Manager and the Data Warehouse Center for business subject area objects

Information Catalog Manager metadata	Data Warehouse Center metadata
Name	Subject area name
Short description	Description
Long description	Notes
Actions	N/A
Data refresh frequency	N/A
URL to access data	N/A
Filename	N/A
For further information	Administrator

Table 48. Metadata mappings between the Information Catalog Manager and the Data Warehouse Center for star schema objects

Information Catalog Manager metadata	Data Warehouse Center metadata
Name	Warehouse schema name
Short description	Description
Long description	Notes
Actions	N/A

Table 48. Metadata mappings between the Information Catalog Manager and the Data Warehouse Center for star schema objects (continued)

For further information	Administrator
URL to access data	N/A
Timestamp source definition last changed	Last update time stamp for warehouse schema definition. This metadata is internal to the Data Warehouse Center. ¹
Note:	
1. If a property is specified as internal to the Data Warehouse Center, it is not displayed.	

Table 49. Metadata mappings between the Information Catalog Manager and the Data Warehouse Center for Data Warehouse Center process objects

Information Catalog Manager metadata	Data Warehouse Center metadata
Name	Process name
Short description	Description
Long description	Process notes
Actions	N/A
For further information	Administrator
URL to access data	N/A
Timestamp source definition last changed	Last update time stamp for the process definition. This metadata is internal to the Data Warehouse Center. ¹
Note:	
1. If a property is specified as internal to the Data Warehouse Center, it is not displayed.	

Metadata mappings between the Information Catalog Manager and OLAP server

Table 50 on page 374 shows the mapping of OLAP server metadata to the Information Catalog Manager common object types. OLAP server metadata refers to metadata for DB2 OLAP Server, DB2 OLAP Integration Server, or Hyperion Essbase Server.

When you publish DB2 OLAP Integration Server metadata to the information catalog, a linked relationship is created between a "dimensions within a multi-dimensional database" object type and a table object.

The left column of the table shows the name of the Essbase API structure. The right column shows the Information Catalog Manager object and object type properties.

Table 50. Mapping of OLAP server metadata to the Information Catalog Manager common object types

OLAP server metadata	Information Catalog Manager metadata
<i>Outline</i>	<i>Multidimensional databases</i>
Four part name of the OLAP object in the following format: server.application.database.outline	Name
Message indicating width and depth limits	Long description
OLAP server (part 1 of Name)	Database host server name
OLAP database (part 3 of Name)	Database or subsystem name
N/A	Database type The value of this property is MULTIDIMENSIONAL.
usOutlineType in ESB_OUTLINEINFO_T	Database extended type The value for this property can be NORMAL or CURRENCY.
N/A	Database status The value for this property is PRODUCTION.
<i>Dimensions in an outline</i>	<i>Dimension within a multidimensional database</i>
Dimension alias from EssOtlGetMemberAlias or name	Name
OLAP server	Database host server name
OLAP database	Database or subsystem name
OLAP application	Using application name
Dimension name	Dimension name
usCategory in ESS_MBRINFO_T	Dimension class or type
<i>Members in a dimension</i>	<i>Members within a multidimensional database</i>
Member alias from EssOtlGetMemberAlias or name	Name
OLAP server	Database host server name
OLAP database	Database or subsystem name
OLAP application	Using application name

Table 50. Mapping of OLAP server metadata to the Information Catalog Manager common object types (continued)

Dimension name	Dimension name
Member name	Member name
last calc string or calc string from EssGetMemberCalc	Derived from
usShare in ESS_MBRINFO_T	This property is treated as a shared member (a member with multiple parents).

Metadata mappings between the DB2 OLAP Integration Server and the Data Warehouse Center

Table 51 shows the mapping of DB2 OLAP Integration Server metadata to Data Warehouse Center metadata.

Table 51. Mapping of DB2 OLAP Integration Server metadata to the Data Warehouse Center objects and tag language

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.
This property is based on a query of the DB2 OLAP Integration Server catalog (OM_INFO table) based on the following columns: Columns MODEL_NAME and MODEL_IDget MODEL_DATA_SOURCE	DatabaseName

Table 51. Mapping of DB2 OLAP Integration Server metadata to the Data Warehouse Center objects and tag language (continued)

<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	VWPProgramInstanceParameterData
N/A	VWP Group – OLAP Integration

Appendix C. How the Data Warehouse Center migrates Version 5.2 definitions

The Data Warehouse Center has a different object model from Visual Warehouse Version 5.2. Read this appendix to understand how the Data Warehouse Center will migrate your Version 5.2 definitions to the Data Warehouse Center.

Mapping between Visual Warehouse and Data Warehouse Center objects

Table 52 shows the mapping between the Visual Warehouse objects and the Data Warehouse Center objects.

Table 52. Mapping between Visual Warehouse objects and Data Warehouse Center objects

Visual Warehouse Version 5.2	Data Warehouse Center
Subject	Subject area
Information resource or information source	Warehouse source
Warehouse	Subject area and warehouse target
Business view	Step

In addition to these objects, the Data Warehouse Center adds a new process object, which contains a set of steps that accomplish a certain warehouse task. After migration, look at the processes that are created, and determine the other processes that you need to create. You can move steps between processes, and you can move processes between subject areas. Move the steps and processes to group the steps into smaller processes that make sense as a single task.

Subjects

Visual Warehouse Version 5.2 subjects will have a corresponding subject area created for it in the Data Warehouse Center. The subject area will have the same name as the Version 5.2 subject. Each subject area will contain one process. The process will be named *subject area name* - Process 1, where *subject area name* is the name of the subject area that contains the process.

Warehouses

Visual Warehouse Version 5.2 warehouses will have a corresponding subject area created for it in the Data Warehouse Center. The subject area will have the same name as the Version 5.2 warehouse. Each subject area will contain

How the Data Warehouse Center migrates Version 5.2 definitions

one process. The process will be named *subject area name* - Process 1, where *subject area name* is the name of the subject area that contains the process.

Business views

Each business view will have a corresponding step in the subject area and process that corresponds to the subject or warehouse that contained the business view. If a business view is included in both a subject and warehouse, then the step that corresponds to the business view will be included in the **VW to DWC Migration Subject** subject area and **VW to DWC Process** process. You must look through these steps and determine the appropriate subject area and process for the steps.

Each step will have a type and subtype associated with it. If the business view used a program or transformer, the step will have the type and subtype for the program or transformer. If the business view used SQL, the step will have a type of SQL.

Trigger programs and concurrent cascade relationships no longer exist. Trigger programs have been migrated to separate steps. Concurrent cascade relationships have been converted to schedules.

If you want to migrate Visual Warehouse metadata synchronization business views to the Data Warehouse Center, promote the business views to production status before you migrate the warehouse control database. If the business views are in production mode, their schedules are migrated to the Data Warehouse Center. If the business views are not in production mode, they will be migrated in test mode without their schedules. You cannot promote the migrated steps to production mode. You must create the synchronization steps again in the Data Warehouse Center and delete the migrated steps.

Visual Warehouse programs

Visual Warehouse programs have changed in the following ways in the Data Warehouse Center:

- The metadata synchronization Visual Warehouse programs are replaced by the Data Warehouse Center Publish facility.
- There are new versions of the DB2 programs for DB2 Universal Database, DB2 for AS/400, and DB2 for OS/390. The versions of the DB2 programs that existed in Visual Warehouse Version 5.2 also still exist for your use in a program group named Visual Warehouse Version 5.2 DB2 Programs.
- The DB2 OLAP Programs program group is now called OLAP Server Programs.

Security changes

The following Visual Warehouse security privileges no longer exist:

- Business View Definition
- Business View Maintenance
- Resource Definition
- Visual Warehouse Program Definition

Only the Administration and Operations privileges remain in the Data Warehouse Center.

There is no longer an Update Security Group for a business view. The security groups that were associated with a Visual Warehouse warehouse or subject will also be associated with the processes that contain the migrated business views from the warehouse or subject.

Appendix D. Defining values for Visual Warehouse 5.2 compatible programs

This appendix provides information for users of Visual Warehouse 5.2 who want to run certain 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.

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.

If the export process generates warnings, the program returns as successfully completed.

This program is available for the following operating systems:

- Windows NT
- AIX
- Solaris Operating Environment
- OS/2

The Column Mapping page is not available for this step.

To define values for a step that uses this program:

1. Open the step notebook.
2. Specify general information about the program. For more information, see "Providing general information about a step" on page 149.
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

Defining values for Visual Warehouse 5.2 compatible programs

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. For more information, see “Defining processing options” on page 151.
6. Click **OK** to save your changes and close the step notebook.

Defining values for a Visual Warehouse 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.

Connect the step to warehouse source and a warehouse target in the Process Model window.

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.

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

Defining values for Visual Warehouse 5.2 compatible programs

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.

To create a tablespace:

```
CREATE TABLESPACE tablespace-name MANAGED BY SYSTEM USING ('d:directory')
```

where *directory* is the directory that is to contain the databases. DB2 creates the directory for you.

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.

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
- AIX
- Solaris Operating Environment
- OS/2

The Column Mapping page is not available for this step.

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. For more information, see "Providing general information about a step" on page 149.
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. 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

Defining values for Visual Warehouse 5.2 compatible programs

comma. Make sure that you enclose this parameter in double quotation marks. Otherwise, the command line processor interprets some characters as file redirection characters.

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"
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.
Deopt <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.

See the *DB2 Utilities* book for more information about these modifiers.

5. On the Processing Options page, provide information about how your step processes. For more information, see "Defining processing options" on page 151.

Defining values for Visual Warehouse 5.2 compatible programs

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

Defining values for a Visual Warehouse 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.

Connect the step to warehouse source and a warehouse target in the Process Model window.

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. 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. 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.

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.

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.

Defining values for Visual Warehouse 5.2 compatible programs

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.

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
- AIX
- Solaris Operating Environment
- AS/400
- OS/2

To define a values for a step that uses this program:

1. Open the step notebook.
2. Specify general information about the warehouse program. For more information, see "Providing general information about a step" on page 149.
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.

The file type modifier can use the following modifiers:

Defining values for Visual Warehouse 5.2 compatible programs

Modifier	Description
Chardel <i>x</i>	<i>x</i> is a single-character string delimiter. The default value is a double quotation mark ("). The character that you specify is used in place of double quotation marks to enclose a character string. You can specify a single quotation mark (') as a character string delimiter as follows: Modified by chardel "
Coldel <i>x</i>	<i>x</i> is a single-character column delimiter. The default value is a comma (,). The character that you specify is used instead of a comma to signal the end of a column. Do not insert a space between coldel and the comma. Enclose this parameter in double quotation marks. Otherwise, the command line processor interprets some characters as file redirection characters. In the following example, coldel ; causes the export utility to interpret any semicolon (;) it encounters as a column delimiter: Db2 "export to temp of del modified by coldel; select * from staff where dept = 20"
Dateisiso	Dateisiso causes all date data values to be exported in ISO format.
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.

See the *DB2 Utilities* book for more information about these modifiers.

4. On the Processing Options page, provide information about how your step processes. For more information, see "Defining processing options" on page 151.
5. Click **OK** to save your changes and close the step notebook.

Defining values for Visual Warehouse 5.2 compatible programs

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.

You schedule this step to run on the target table of a process, after the process completes. In the Process Model window, draw a data link from the step to the warehouse target.

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.

To define a values for a step that uses this program:

1. Open the step notebook.
2. Specify general information about the warehouse program. For more information, see “Providing general information about a step” on page 149.
3. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
4. Click **OK** to save your changes and close the step notebook.

Define 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.

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. In the Process Model window, draw a data link from the step to the warehouse target.

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:

Defining values for Visual Warehouse 5.2 compatible programs

- 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.

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. For more information, see “Providing general information about a step” on page 149.
3. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
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 UDB EEE (VWPLDPR) program (AIX only)

Use this step to run the Visual Warehouse 5.2 Load flat file into DB2 UDB EEE (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.

Before you use this warehouse program, you must be familiar with parallel system concepts and parallel load.

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

Defining values for Visual Warehouse 5.2 compatible programs

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.

The Load flat file into DB2 UDB EEE 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.

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.

This program runs on AIX.

To define a values for a step that uses this warehouse program:

1. Open the step notebook.
2. Specify general information about the program. For more information, see "Providing general information about a step" on page 149.
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 "\"

Defining values for Visual Warehouse 5.2 compatible programs

- c. Double-click the **Parameter value** field for the Decimal delimiter parameter, and type the decimal delimiter. The most common values are a period (.) or a comma (,).
 - d. Double-click the **Parameter value** field for the local, non-shared file directory parameter, and type the local, non-shared file directory.
 - e. Double-click the **Parameter value** field for the Path name and prefix for the parameter, and type the path name and prefix for the split files. The name of each file will consist of the prefix plus a numeric identifier.
 - f. Double-click the **Parameter value** field for the Partition key parameter, and type a parameter for each partition key. The partition key must be in the format used by the db2split database utility. The format generally is as follows: *col1,1,,,N,integer* followed by *col3,3,,5N,character*
4. On the Processing Options page, provide information about how your step processes. For more information, see “Defining processing options” on page 151.
 5. Click **OK** to save your changes and close the step notebook.

Defining values for Visual Warehouse 5.2 compatible programs

Appendix E. The data warehousing sample

DB2 Universal Database provides a data warehousing sample that you can use to familiarize yourself with the Data Warehouse Center. It includes sample data and metadata that you can run to create tables in a warehouse database.

Install and run the sample to learn about the relationships between source and target data, and the definitions in the Data Warehouse Center that move and transform the data.

The sample defines a star schema in the Data Warehouse Center. The star schema has three dimension tables:

- PRODUCT
- SCENARIO
- TIME

It has one fact table, named FACT TABLE.

Installing the sample

You can install the Data Warehousing sample on Windows NT only. The sample is installed as part of the typical installation of DB2 Universal Database on Windows NT. You can also install the sample by selecting the **Getting Started** component, or the **First Steps** and **Sample Databases** subcomponents of the **Getting Started** component, in a custom installation.

You must install the sample on the same workstation as the warehouse server.

Creating the sample databases

You must create the sample databases after you install the files for the sample. To create the databases:

1. Open the First Steps window.
2. Click **Create Sample Databases**.
The Create SAMPLE Databases window opens.
3. Select the **Data Warehousing sample** check box.
4. Click **OK**.
5. If you are installing the data warehousing sample, a window opens for the DB2 user ID and password to use to access the sample.

- a. Type the user ID and password that you want to use. You must specify a valid DB2 user ID and password.
- b. Click **OK**.

A progress window opens. When the databases are created, click **OK**.

The databases that you created are registered with ODBC.

The following databases are created:

DWCTBC

Contains the operational source tables that are required for the Data warehousing section of the tutorial.

TBC_MD

Contains metadata for the Data Warehouse Center.

Creating the warehouse database

You need to create a database for the data that is generated when you run the sample.

To create the database:

1. Start the DB2 Control Center:
 - On Windows NT, click **Start** → **Programs** → **IBM DB2** → **Control Center**.
 - On AIX or the Solaris Operating Environment, type the following command:

```
db2jstrt 6790
db2cc 6790b
```
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:
SAMPWHS

If you use a different name, you must change the name of the database in the TBC Sample Targets warehouse target. Otherwise the sample will not run.

4. From the **Default drive** list, select a drive for the database.
5. In the **Comments** field, type a description of the database:
Sample warehouse database
6. Click **Finish**. All other fields and pages in this wizard are optional. The SAMPWHS database is created and is listed in the DB2 Control Center.

There are several ways that you can register a database with ODBC. You can use the Client Configuration Assistant on Windows NT, the DB2 Command Line Processor, or the ODBC32 Data Source Administrator on Windows NT. The following instructions apply to the Client Configuration Assistant.

For more information about the Command Line Processor, see the *DB2 Universal Database Command Reference*. For more information about the ODBC32 Data Source Administrator, see the online help in the Administrator.

To register the SAMPWHS database with ODBC on Windows NT:

1. Start the Client Configuration Assistant by clicking **Start** —> **Programs** —> **IBM DB2** —> **Client Configuration Assistant**. The Client Configuration Assistant window opens.
2. Select **SAMPWHS** from the list of databases.
3. Click **Properties**. The Database Properties window opens.
4. Select **Register this database for ODBC**. Use the default selection of **As a system data source**, which means that the data is available to all users on the system.
5. Click **OK**. All other fields are optional. The SAMPWHS database is registered with ODBC.

Using Sample Contents

Use Sample Contents to view a sample of the data contained in a warehouse table. Run Sample Contents from the Tables folder that contains the table you want to sample, or from the table icon in the process modeler.

When you run Sample Contents on a warehouse target from the process modeler, the warehouse source, warehouse target, and step must have a common agent site, or the request will fail. To avoid this problem, select the step on the Programs page in the Properties notebook for the agent site you want to use.

When you run Sample Contents against a target file, the first agent site on the selected list is used. However, database maintenance operations might affect the order of the agent sites listed. Sample contents will fail if the agent site selected does not reside in the same system as the source or target file.

A Sample Contents request that uses the AS/400 agent on a flat file source will fail.

Viewing the sample data

The DWCTBC database contains the source tables for the sample warehouse. It contains the following tables:

- SALES

- INVENTORY
- PRODUCTION_COSTS
- GEOGRAPHIES
- SCENARIO
- TIME
- PRODUCT

To view the data in these tables:

1. From the DB2 Control Center, expand the objects in the DWCTBC database until you see the **Tables** folder.
2. Click the folder. In the right pane, you see all the tables for the database.
3. Find the table that you want to view. Right-click it, and click **Sample Contents**.

Up to 200 rows of the table are displayed. The column names are displayed at the top of the window. You might need to scroll to the right to see all the columns and scroll down to see all the rows.

Viewing and modifying the sample metadata

To access the sample, you must log on to the Data Warehouse Center, specifying TBC_MD as the warehouse control database.

If the TBC_MD database is not local to the workstation that contains the warehouse server, you must catalog it as a remote database on the workstation. If it is not local to the workstation that contains the Data Warehouse Center administrative client, you must catalog it on that workstation as well.

To log on to the Data Warehouse Center:

1. Click **Tools** —> **Data Warehouse Center** in the DB2 Control Center. The system will automatically try to log on to the Data Warehouse Center using your DB2 Control Center user name and password. The Data Warehouse Center Logon window opens.
2. Click **Advanced**.
The Advanced window opens.
3. In the **Control database** field, type TBC_MD, the name of the warehouse control database that is included in the sample.
4. In the **Server host name** field, type the TCP/IP host name for the workstation where the warehouse server is installed.
5. Click **OK**.

The Advanced Logon window closes.

The next time that you log on, the Data Warehouse Center will use the settings that you specified in the Advanced Logon window. If you first log on to the DB2 Control Center with a user ID that is defined to the Data Warehouse Center, then the Data Warehouse Center will use the same user ID to log you on automatically when you click **Tools** → **Data Warehouse Center**.

6. In the **User ID** field of the Logon window, type the user ID that you specified when you created the Data Warehousing sample databases.
7. In the **Password** field, type the password for the user ID.
8. Click **OK**.

The Data Warehouse Center Logon window closes, and you are now logged on to the Data Warehouse Center.

After you log on to the Data Warehouse Center, you can view and modify the properties of the sample metadata:

1. Expand the **Warehouse Sources** folder.
The TBC Sample Sources warehouse source is displayed.
2. Right-click the **TBC Sample Sources** warehouse source and click **Properties** to view the properties of the warehouse source.
3. In the **System Name** field of the Database page, specify the host name of the workstation on which the sample database exists.
This step is optional and is used only if you are using a database or file that has the same name as another database or file on a different workstation.

4. In the **User ID** field, type the user ID that you specified when you created the sample database.
5. In the **Password** field, type the password for the user ID.
6. Type the password again in the **Verify Password** field.
7. Click **OK** to close the notebook.
8. Click the **Tables** folder. The tables in the warehouse source are displayed on the right-hand side of the window.

The warehouse source contains the following tables:

- GEOGRAPHIES
- INVENTORY
- PRODUCT
- SCENARIO
- TIME
- PRODUCTION_COSTS
- SALES

9. Right-click a table, and click **Properties** to view its properties.

10. Click **OK** or **Cancel** to close the notebook.
11. Expand the **Warehouse Targets** folder.
The TBC Sample Targets warehouse target is displayed.
12. Right-click the **TBC Sample Targets** warehouse target and click **Properties** to view the properties of the warehouse target.
13. In the **System Name** field of the Database page, specify the host name of the workstation on which the sample database exists.
This step is optional and is used only if you are using a database or file that has the same name as another database or file on a different workstation.
14. In the **User ID** field, type the user ID that you specified when you created the sample database.
15. In the **Password** field, type the password for the user ID.
16. Type the password again in the **Verify Password** field.
17. Click **OK** to close the notebook.
18. Expand the **Subject Area** folder.
An icon for the **TBC Sample** subject area is displayed.
19. Right-click the **TBC Sample** subject area and click **Properties** to view the properties of the subject area.
20. After you have finished viewing the properties, click **OK** or **Cancel** to close the notebook.
21. Expand the **TBC Sample** subject area tree.
22. Expand the **Processes** folder.
Four processes are displayed under the folder:
 - Build Sample Fact Table
 - Build Sample Product Dimension
 - Build Sample Scenario Dimension
 - Build Sample Time Dimension
23. Right-click the **Build Sample Fact Table** process, and click **Open**.
The Process Model window opens.
The process contains the following objects:
 - Three sources:
 - SALES
 - PRODUCTION_COSTS
 - INVENTORY
 - An SQL step named Fact Table SQL. The SQL step joins the sources in the process, and adds three columns: CITY_ID, TIME_ID, and SCENARIO_ID.
 - A target table named TARGET FACT TABLE.

- A shortcut to the Select Time step. This shortcut is used in scheduling the sequence of steps to run in this sample.

To view the metadata for an object in a process, right-click the object and click **Properties**. Click **OK** or **Cancel** to close the notebook.

24. Right-click the **Build Sample Product Dimension** process, and click **Open**.

The Process Model window opens.

The process contains the following objects:

- A source named PRODUCT.
- An SQL step named Select Product. The SQL step selects all the columns from the source.
- A generated target table named TARGET PRODUCT.
- A shortcut to the Select Scenario step. This shortcut is used in scheduling the sequence of steps to run in this sample.

To view the metadata for an object in a process, right-click the object, and click **Properties**. Click **OK** or **Cancel** to close the notebook.

25. Right-click the **Build Sample Scenario Dimension** process, and click **Open**.

The Process Model window opens.

The process contains the following objects:

- A source named SCENARIO.
- An SQL step named Select Scenario. The SQL step selects all the columns from the source. You cannot use SQL Assist to edit the SQL in this step. The SQL was edited after it was created by the SQL Assist.
- A generated target table named TARGET SCENARIO.
- A shortcut to the Select Time step. This shortcut is used in scheduling the sequence of steps to run in this sample.

To view the metadata for an object in a process, right-click the object and click **Properties**. Click **OK** or **Cancel** to close the notebook.

26. Right-click the **Build Sample Time Dimension** process, and click **Open**.

The Process Model window opens.

The process contains the following objects:

- A source named TIME.
- An SQL step named Select Time. The SQL step selects all the columns from the source.
- A generated target table named TARGET TIME.
- A shortcut to the Fact Table SQL step. This shortcut is used in scheduling the sequence of steps to run in this sample.

To view the metadata for an object in a process, right-click the object, and click **Properties**. Click **OK** or **Cancel** to close the notebook.

27. Expand the **Warehouse Schemas** folder.
28. Right-click the warehouse schema and click **Open**.

The Warehouse Schema Model window opens. It contains a join of the following tables:

- TARGET FACT TABLE
- TARGET PRODUCT
- TARGET SCENARIO
- TARGET TIME

Promoting the steps

Before you run the steps, you must promote them to test mode, and then to production mode. You must promote them in the order in which they will run:

1. Select Product
2. Select Scenario
3. Select Time
4. Fact Table SQL

To promote the steps to test mode:

1. Right-click the **Build Sample Product Dimension** process, and click **Open**.
2. Right-click the Select Product step, and click **Mode** → **Test**.

The Data Warehouse Center displays a progress window. Wait until the Data Warehouse Center finishes processing before you start the next procedure.

3. Repeat steps 1 and 2 for the rest of the steps to promote the steps to test mode:

- The Select Scenario step in the Build Sample Scenario Dimension process.
- The Select Time step in the Build Sample Time Dimension process.
- The Fact Table SQL step in the Build Sample Fact Table process.

To promote the steps to production mode:

1. Right-click the **Build Sample Product Dimension** process, and click **Open**.
2. Right-click the Select Product step, and click **Mode** → **Production**.

The Data Warehouse Center starts to create the target table, and displays a progress window. Wait until the Data Warehouse Center finishes processing before you start the next procedure.

3. Repeat steps 1 on page 400 and 2 on page 400 for the rest of the steps to promote the steps to production mode:
 - The Select Scenario step in the Build Sample Scenario Dimension process.
 - The Select Time step in the Build Sample Time Dimension process.
 - The Fact Table SQL step in the Build Sample Fact Table process.

Running the steps

To run the steps, you run the Sample Product step. The rest of the steps start, in sequence, after the Sample Product step finishes processing.

To run the Sample Product step:

1. From the main Data Warehouse Center window, click **Warehouse** —> **Work in Progress**.

The Work in Progress window opens. You use the Work in Progress window to monitor the progress of all steps in the Data Warehouse Center that are running or scheduled.

2. Click **Work in Progress** —> **Run New Step**.

The Run New Step window opens.

3. Select the Sample Product step, and click >.
4. Click **OK**.

You should see an entry for the step that is running. While the step is running, it has a status of Populating. When it finishes running, it should have a status of Successful. As one step in the sequence finishes processing, the next step starts to run and has a status of Populating.

Viewing the sample warehouse data

The SAMPWHS database that you created contains the results of the step processing. It contains the following tables:

- TARGET FACT TABLE
- TARGET PRODUCT
- TARGET SCENARIO
- TARGET TIME

To view the data in these tables:

1. From the DB2 Control Center, expand the objects in the SAMPWHS database until you see the **Tables** folder.
2. Click the folder. In the right pane, you see all the tables for the database.
3. Find the table that you want to view. Right-click it, and click **Sample Contents**.

Up to 200 rows of the table are displayed. The column names are displayed at the top of the window. You might need to scroll to the right to see all the columns and scroll down to see all the rows.

Appendix F. 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 OS/390. 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?” on page 404
- “What are its components?” on page 404

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.

What are its components?

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

- “Warehouse agents” on page 405
- “CROSS ACCESS ODBC driver” on page 406
- “Classic Connect data server” on page 406
- “Enterprise server” on page 408
- “Data mapper” on page 410

Figure 28 on page 405 shows how Classic Connect and its components fit into the overall Data Warehouse Center architecture.

Using Classic Connect with the Data Warehouse Center

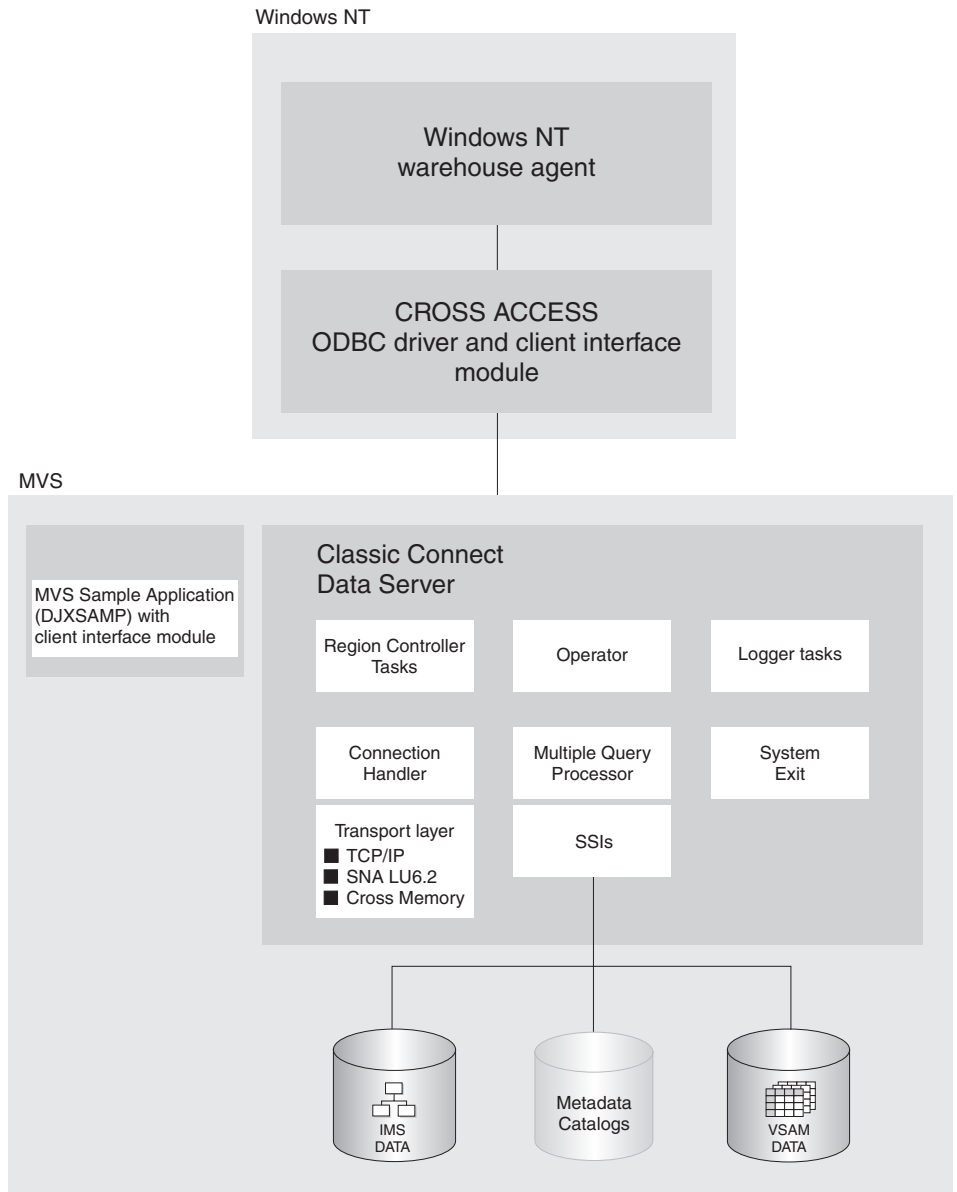


Figure 28. Classic Connect architecture

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. For information about setting up the ODBC interface for z/OS and OS/390, see *Warehouse Manager Installation Guide*.

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 OS/390.

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

Using Classic Connect with the Data Warehouse Center

- Query processor services
- Logger services

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 OS/390 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 OS/390 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 OS/390 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 OS/390 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
- OS/390 cross memory services.

Using Classic Connect with the Data Warehouse Center

The OS/390 client application, DJXSAMP, can connect to a data server using any of these methods; however, the recommended approach for local clients is to use OS/390 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.

Using Classic Connect with the Data Warehouse Center

Figure 29 shows how the enterprise server fits into a Classic Connect configuration:

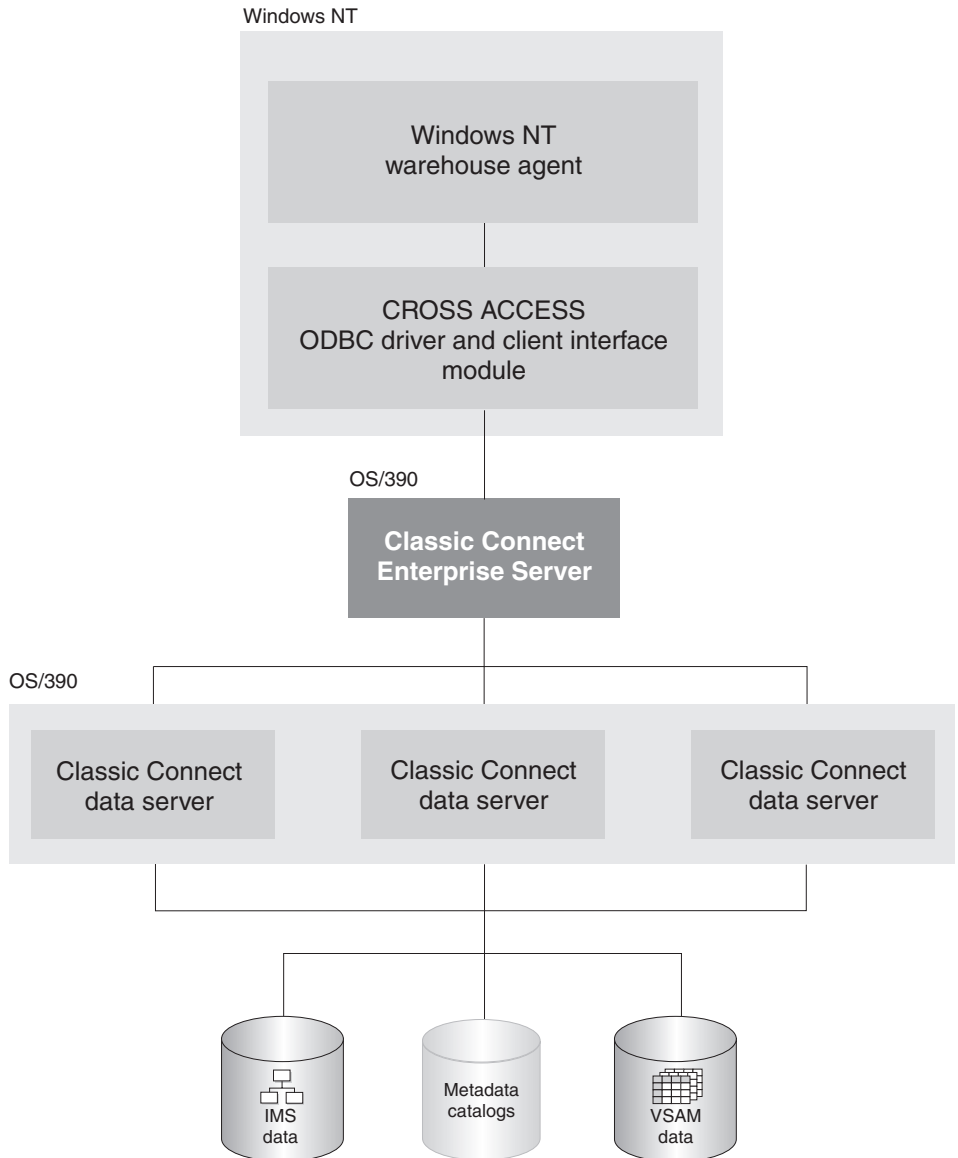


Figure 29. Classic Connect architecture with the enterprise server implemented

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

Using Classic Connect with the Data Warehouse Center

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.

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

Using Classic Connect with the Data Warehouse Center

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

Figure 30 shows the data administration workflow with data mapper.

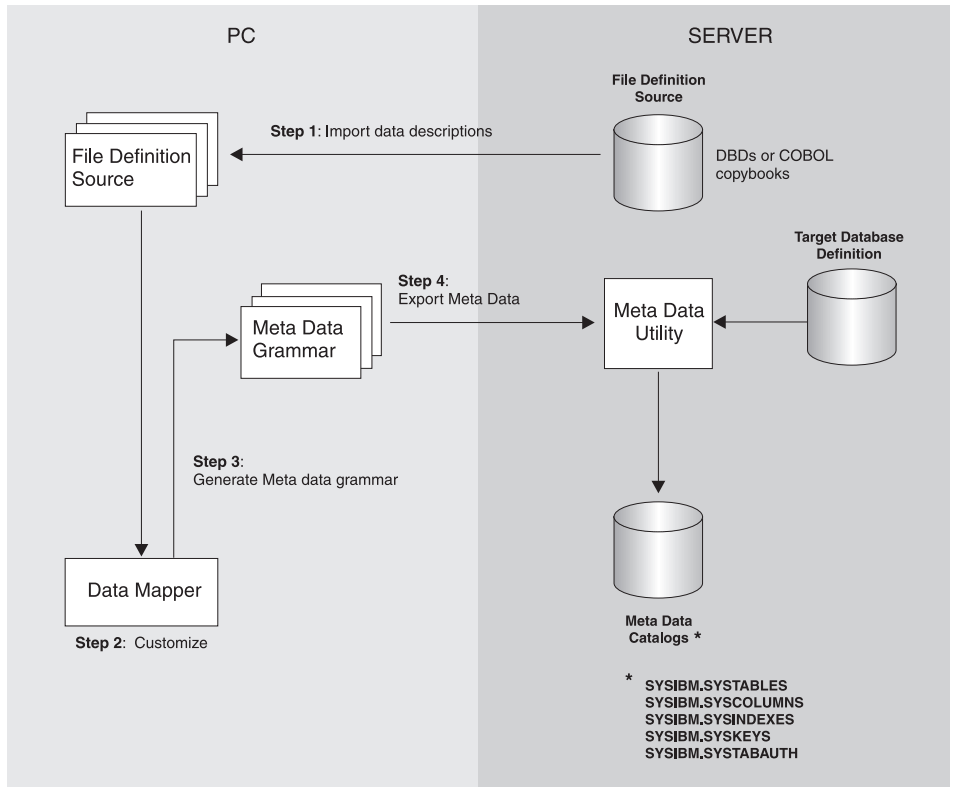


Figure 30. Data mapper workflow

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

The steps in Figure 30 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.

Using Classic Connect with the Data Warehouse Center

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.

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 OS/390.
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.

Setting up the environment

This section summarizes the requirements for setting up integration between Classic Connect and the Data Warehouse Center.

Hardware and software requirements

The integration requires the following software:

- DataJoiner Classic Connect Version 2 Release 1
- DB2 Universal Database Version 7

Optionally, you can use the DataJoiner Classic Connect data mapper to generate metadata grammar for you. You can obtain the data mapper from the following Web site:

<http://www.software.ibm.com/data/datajoiner/news.html#newcxa>

Installing and configuring prerequisite products

Complete the tasks summarized in Table 53 on page 413 to set up integration between Classic Connect and the Data Warehouse Center. See the documentation listed in each task for more information.

Using Classic Connect with the Data Warehouse Center

Table 53. Summary of installation and configuration tasks

Task	Content	Location
Learning about the integration	What is Classic Connect?	"What is Classic Connect?" on page 403
	Concepts and terminology	Chapter 2 of <i>DataJoiner Classic Connect: Installation, Configuration, and Reference Guide</i>
Installing and configuring the data server	System requirements and planning	Chapter 3 of <i>DataJoiner Classic Connect: Installation, Configuration, and Reference Guide</i>
	Installing Classic Connect on OS/390	Chapter 4 of <i>DataJoiner Classic Connect: Installation, Configuration, and Reference Guide</i>
	Data server installation and verification procedure	Chapter 6 of <i>DataJoiner Classic Connect: Installation, Configuration, and Reference Guide</i>
	Introduction to data server setup	Chapter 6 of <i>DataJoiner Classic Connect: Installation, Configuration, and Reference Guide</i>
	Configuring communication protocols between OS/390 and Windows NT	"Configuring communications protocols between OS/390 and Windows NT" on page 414
Installing and configuring the client workstation	Configuring a Windows NT client	"Configuring a Windows NT client" on page 424
	Defining an agent site	"Defining agent sites" on page 13

Using Classic Connect with the Data Warehouse Center

Table 53. Summary of installation and configuration tasks (continued)

Using an IMS or VSAM warehouse source	Mapping nonrelational data and creating queries	Chapter 13 of <i>DataJoiner Classic Connect: Installation, Configuration, and Reference Guide</i> and <i>DataJoiner Classic Connect: Data Mapper Installation and User's Guide</i>
	Optimization	Chapter 14 of <i>DataJoiner Classic Connect: Installation, Configuration, and Reference Guide</i>
	Defining a warehouse source	"Defining a non-DB2 database warehouse source in the Data Warehouse Center" on page 97
Migrating from the Visual Warehouse Host Adapters	Migrating from the Visual Warehouse Host Adapters to Classic Connect	"Migrating from the Visual Warehouse Host Adapters to Classic Connect" on page 435

Configuring communications protocols between OS/390 and Windows NT

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 OS/390.

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
- TCP/IP

Cross memory

Cross memory should be used to configure the local OS/390 client application (DJXSAMP) to access a data server. Unlike for SNA and TCP/IP, there are no setup requirements to use the OS/390 cross memory interface. This interface

Using Classic Connect with the Data Warehouse Center

uses OS/390 data spaces and OS/390 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 OS/390 system and on your Windows NT 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.

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" on page 419.

Using Classic Connect with the Data Warehouse Center

Both interfaces allow Classic Connect to communicate with the OS/390 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 OS/390

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. For more detailed information about IBM's or Interlink's TCP/IP, refer to the appropriate product's documentation.

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.

Determine the following values for the OS/390 system on which Classic Connect is being installed, and enter these values in the worksheet portion of Figure 31 on page 419.

IP address or hostname

Obtain either the hostname or IP address of the OS/390 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 address. However, it is recommended that you use hostnames in remote client configuration files for readability and ease of future configuration changes.

Using Classic Connect with the Data Warehouse Center

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 OS/390 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 OS/390 TCP/IP on off-loaded devices, specify the address of the TCP/IP stack on the OS/390 image, not the address of an off-load gateway's IP stack.

Port number

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 OS/390 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 Figure 31 on page 419 to complete the OS/390 configuration of your TCP/IP communications.

Configuring TCP/IP on Windows NT

You must configure your Windows NT machine to locate the data server on OS/390.

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 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.
- 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.

Using Classic Connect with the Data Warehouse Center

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 Figure 31 provides you with an example set of TCP/IP values for an OS/390 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	

Figure 31. TCP/IP Communications Template and Worksheet

Configuring the LU 6.2 communications protocol

This section describes the values you must determine and steps you must perform, both on your OS/390 system and on your Windows NT system, to configure LU 6.2 (SNA/APPC) communications for Classic Connect.

Requirement:

For APPC connectivity between Classic Connect and DataJoiner for Windows NT, 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 parameter relationships on OS/390 and Windows NT and assist you with your LU 6.2 configuration.

Using Classic Connect with the Data Warehouse Center

Configuring LU 6.2 on OS/390

If you will be using LU 6.2 to access Classic Connect from DataJoiner, you need to configure VTAM® table definitions on your OS/390 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 Windows NT

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 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 OS/390 discussed in “Configuring LU 6.2 on OS/390”.

- APPC Mode

See the mode, CX62R4K in Figure 32 on page 422. It has a maximum RU size of 4096.

- CPIC Symbolic Name

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 OS/390 system.

Using Classic Connect with the Data Warehouse Center

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 OS/390 respectively), are 'Active,' the connection is ready to test with an application.

APPC communications template and worksheet

Figure 32 on page 422, which provides you with an example set of VTAM[®] and SNA values, is included for reference. Use Figure 33 on page 423, which is a duplicate of Figure 32 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

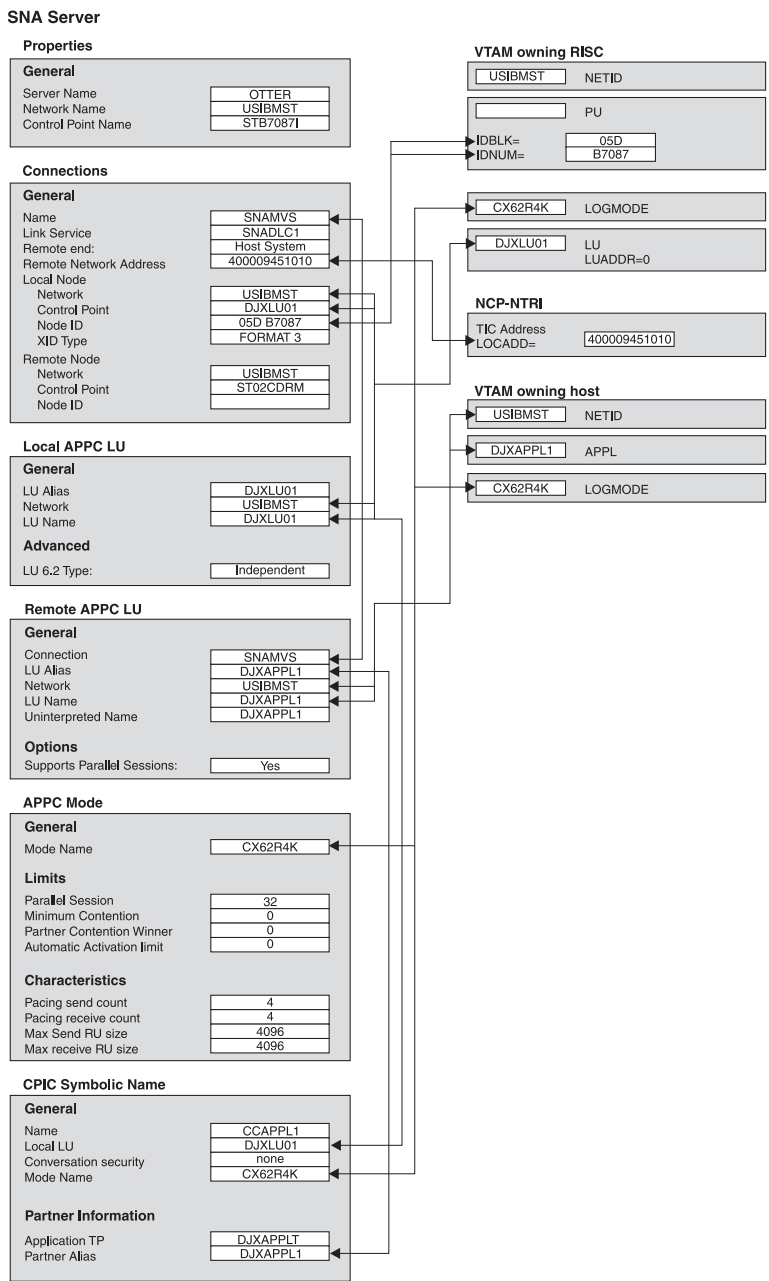


Figure 32. LU 6.2 Configuration Template

Using Classic Connect with the Data Warehouse Center

SNA Server

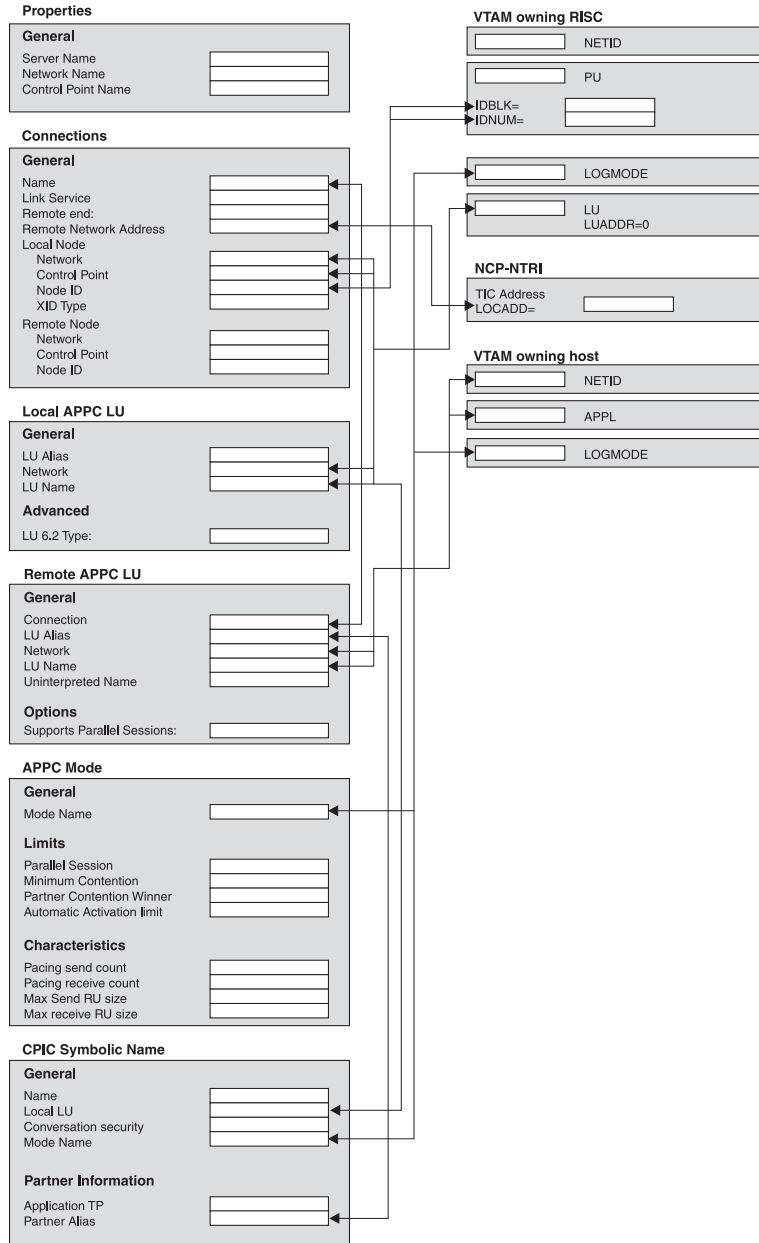


Figure 33. LU 6.2 Configuration Worksheet

Configuring a Windows NT 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

Installing the Classic Connect ODBC Driver:

1. Insert the Warehouse Manager CD-ROM into your CD-ROM drive.
2. Click **Install** from the launchpad.
3. In the Select Products window, verify that the **DB2 Warehouse Manager** check box is selected, then click **Next**.
4. In the Select Installation Type window, select **Custom**, then click **Next**.
5. In the Select Components window, select **Classic Connect Drivers** and **Warehouse Agent**, clear all other check boxes, and then click **Next**.
6. In the Start Copying Files window, review your selections.

If you want to change any of your selections, click **Back** to return to the window where you can change the selection.

7. Click **Next** to begin the installation.

Installing the CROSS ACCESS ODBC driver

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

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 DataJoiner 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.

To open the ODBC Data Source Administrator window:

1. Click **Start** on your desktop and click **Settings**.
2. Click **Control Panel**.
3. Open the ODBC icon. The ODBC Data Source Administrator window opens.

Using Classic Connect with the Data Warehouse Center

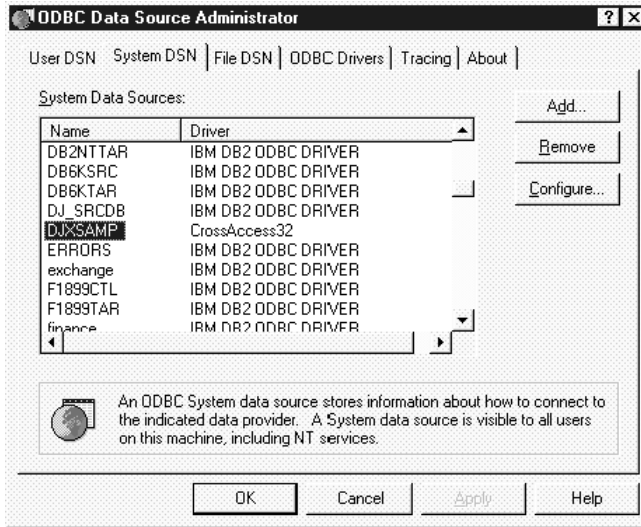


Figure 34. ODBC Data Source Administrator window

This window displays a list of data sources and drivers on the System DSN page.

Adding and configuring a data source

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.

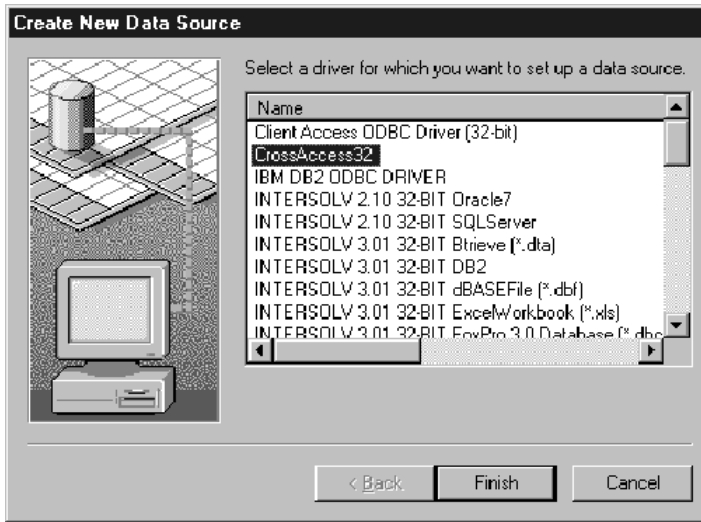


Figure 35. Create New Data Source window

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.

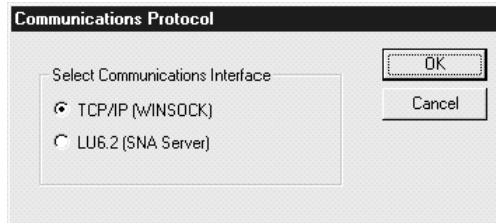


Figure 36. Communications Protocol window

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

- For the TCP/IP communications interface, see “Configuring TCP/IP communications”.
- For the LU 6.2 communications interface, see “Configuring LU 6.2 communications” on page 430.

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 screenshot shows the 'CROSS ACCESS ODBC Data Source Configuration' dialog box. It has a title bar with a close button. The 'General' tab is active, showing the following fields and options:

- Data Source Name:** CLASSAMP
- Description:** Classic Connect
- Communications Configuration:**
 - Host IP Address:** xxx.xxx.xxx.xxx
 - Host Port Number:** 1035
- Authorizations:**
 - OS Login ID Required
 - DB User ID Required
 - File ID Required
- Read Only

Buttons at the bottom: OK, Cancel, Apply.

Figure 37. The CROSS ACCESS ODBC Data Source Configuration window for TCP/IP

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. (For an example, see Chapter 9 of *DataJoiner Classic Connect: Installation, Configuration, and Reference Guide*.)
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. (For an example, see Chapter 9 of *DataJoiner Classic Connect: Installation, Configuration, and Reference Guide*.)
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.

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

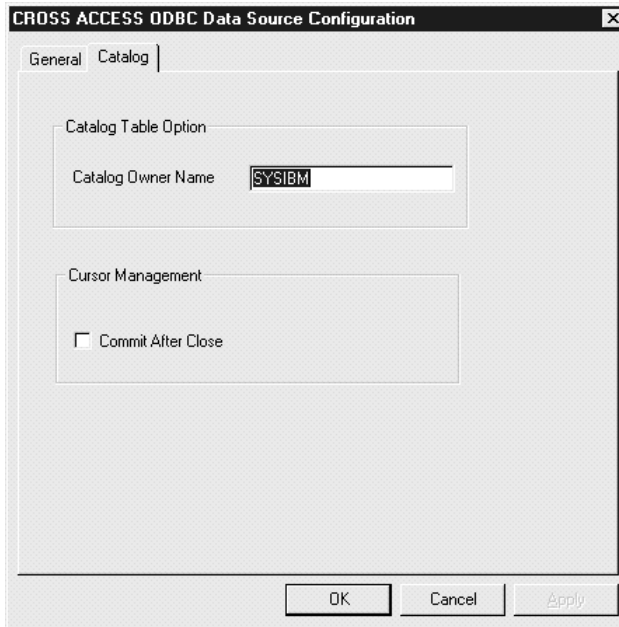


Figure 38. CROSS ACCESS ODBC Data Source Configuration window

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.

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.

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.

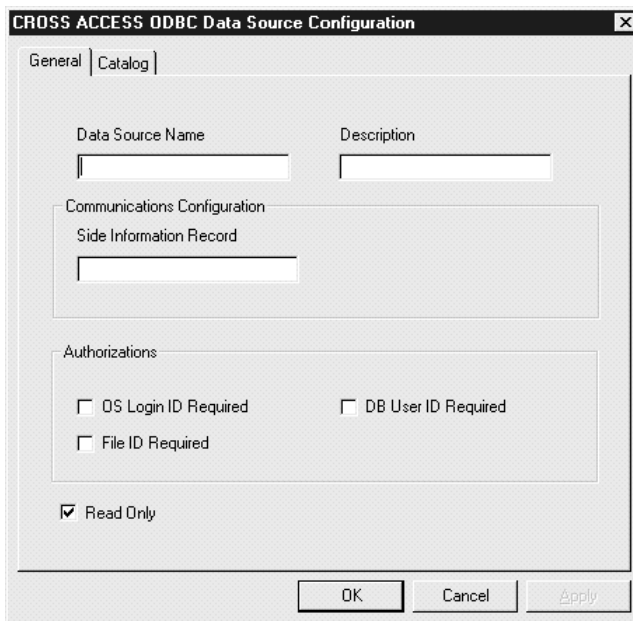


Figure 39. The CROSS ACCESS ODBC Data Source Configuration window for LU 6.2

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. (For an example, see Chapter 9 of *DataJoiner Classic Connect: Installation, Configuration, and Reference Guide*.)
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 Figure 33 on page 423) 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.
 - **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.

Using Classic Connect with the Data Warehouse Center

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: Use the Catalog page to perform the following tasks:

- Specify the catalog table option.
- Specify cursor management.

To set the database catalog options:

1. Click the **Catalog** tab in the Configuration window.

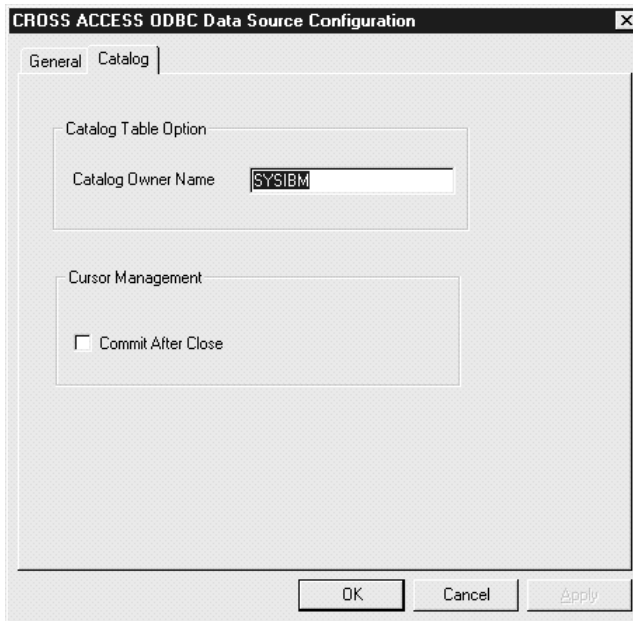


Figure 40. Database Catalog Options window

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.

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.

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.

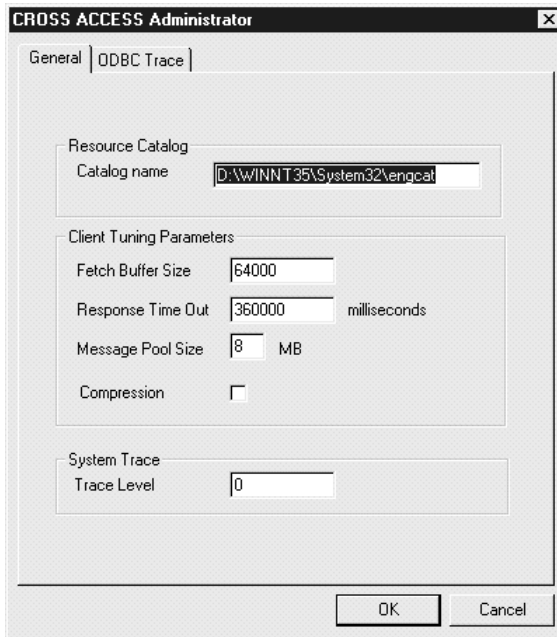


Figure 41. General page of the CROSS ACCESS Administrator window

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.

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

Using Classic Connect with the Data Warehouse Center

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:

- 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.

Using Classic Connect with the Data Warehouse Center

7. Click the **ODBC Trace** tab in the CROSS ACCESS Administrator window.

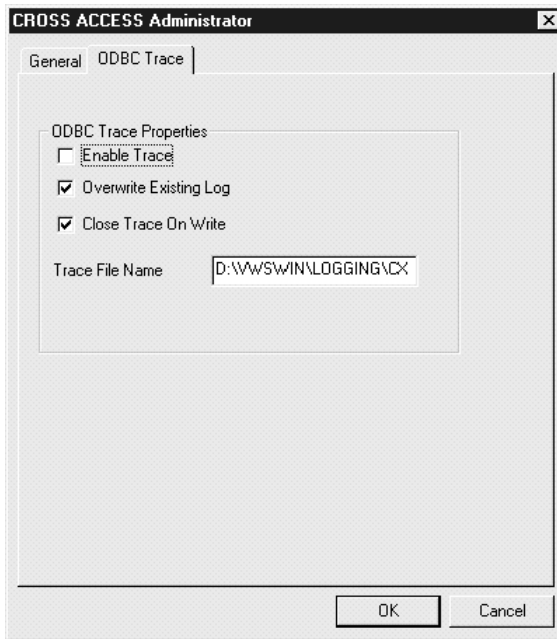


Figure 42. ODBC Trace page of the CROSS ACCESS Administrator window

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**.

Migrating from the Visual Warehouse Host Adapters to Classic Connect

To migrate from the Visual Warehouse V1.3 Host Adapters to Classic Connect:

1. Edit the Meta Data utility. The sample Meta Data Utility JCL is found in the SDJXSAMP member DJXMETAU. See Appendix A of the *DataJoiner Classic Connect: Installation, Configuration, and Reference Guide* for more information.
2. Supply a valid job card and modify the DJX high-level qualifier.

Using Classic Connect with the Data Warehouse Center

3. If you are mapping IMS data, modify the IMS high-level qualifier and the DBDLIB DD statement.
4. If you are referring to VSAM files by DD names, add DD statements with the same DD names in the Meta Data utility JCL.
5. The first time that you run the Meta Data utility, you need to create the metadata catalog files. Unlike the Visual Warehouse Host Adapters, both IMS and VSAM metadata can exist in the same catalog.
6. Define the metadata grammar files that were used with the Visual Warehouse Host Adapters as input to the Meta Data utility.
7. Verify execute and access authority for the data sets referred to in the JCL.
8. Submit the Meta Data utility JCL for execution and review the output.
9. If necessary, install and configure the CROSS ACCESS ODBC driver.
10. Adapt the Visual Warehouse Host Adapter setup to use Classic Connect:
 - a. Open the Warehouse Source notebook.
 - b. From the **Warehouse source type** list, select **IMS** or **VSAM**.
 - c. Select the **Database** tab.
 - d. Change the data source name to be the same as the data source name that you used when configuring the CROSS ACCESS ODBC driver.
 - e. Close the Warehouse Source notebook.
11. Run a step that uses the modified warehouse source to ensure it runs successfully.

Appendix G. 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 Table 54, the drive letter C indicates a local drive.

Table 54. Data Warehouse Center environment variable updates for Windows

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

Data Warehouse Center registry updates

The following values are added to HKEY_LOCAL_MACHINE\Software\IBM\DB2\DataWarehouseCenter\ServiceParms in the Windows NT or Windows 2000 registry:

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 variables

Data Warehouse Center configuration files

The following updates are made to TCP/IP configuration information:

TCP/IP SERVICES:

vwkernel	11000/tcp
vwlogger	11002/tcp

Appendix H. 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 Data Transformation Services (DTS) allows you to import, export, and transform data between OLE DB sources and targets to build data warehouses and datamarts. 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.

Software requirements

The following software requirements apply.

- DB2 Universal Database for Windows NT Version 7.2 as the warehouse target database
- DB2 Warehouse Manager Version 7.2

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.
- 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

Microsoft OLE DB and Data Transaction Services support

is not successful, a warning message is displayed and you must either catalog or choose a different database alias in the wizard.

- 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.
- 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.
- The DTS package connection string has the same syntax as the `dtsrun` command.

Creating views for OLE DB table functions

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**.
The OLE DB Assist wizard opens. The wizard steps you through the task of creating a new view in the warehouse source database.

Creating views for DTS packages

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.

Appendix I. Creating an i2 step

With the Connector for i2 TradeMatrix BPI, you can schedule updates to i2 business models, distributed datamarts, and OLAP cubes. You must install the DB2 Warehouse Manager Connectors. For more information, see *DB2 Warehouse Manager Installation Guide*.

The i2 TradeMatrix Business Process Intelligence (BPI) products are an enterprise-wide analytical and reporting solution from i2 Technologies, Inc. With BPI, you create Online Analytical Processing (OLAP) definitions for dimensions, measures (facts), hierarchies, levels, and so on, using the i2 TradeMatrix BPI Architect. These definitions are stored in a corporate model in the i2 Business Repository. You create business models from the definitions that are stored in the corporate model in the i2 Business Repository. You can deploy business models to create star schema dimension and fact tables in a distributed data mart (DDM). From these star schema tables, you can create OLAP cube structures for reporting and analysis.

With the Connector for i2 TradeMatrix BPI, you can:

- Import changes to the metadata of a deployed BPI business model.
- Update the BPI distributed datamart and an i2 OLAP cube for the business model.

The Connector for i2 TradeMatrix BPI generates the steps that manage the loading of the i2 dimension and fact tables from existing input files (in standard BPI comma-separated variable (CSV) format), as well as the loading of the OLAP cube. After you import the metadata for i2 sources, the Data Warehouse Center creates and populates the **Processes** folder for the i2 load steps. Because the i2 load steps are controlled by the Data Warehouse Center, you can control when and how often they run, or run them on demand.

The Connector for i2 TradeMatrix BPI runs on Microsoft Windows NT or Windows 2000. The i2 TradeMatrix business-model source can be on any supported platform.

Importing metadata for i2 sources

To import metadata for i2 sources into the Data Warehouse Center:

1. Right-click the **Warehouse** folder, click **Import Metadata**, and click **i2**.

You can only import metadata from deployed i2 business models into the Data Warehouse Center.

Creating an i2 step

2. On the Connection page of the Import Metadata window, type the database name of the DB2 database that contains the i2 TradeMatrix BPI Business Repository, the schema name for the i2 Business Repository, and the user ID and password you use to connect to the database.
3. On the Target page of the Import Metadata window, select the agent site, the subject area, the process, the warehouse target, and the schema name for target. Select the deployed business model from which you want to import metadata.

The warehouse agent site is the workstation where the Connector for i2 TradeMatrix BPI runs, where the i2 TradeMatrix BPI Engine is installed, and where the DDM (warehouse target) is cataloged.

Creating the i2 steps

After you import the metadata for i2 sources, the Data Warehouse Center creates and populates the **Processes** folder for the i2 load steps. When you open the Process Model window, the steps are already added to the canvas and serve as a template for running the i2 load steps.

Restriction: The i2 step to update the BPI Business Model fact table and OLAP cube can only refer to deployed BPI Business Models that are associated with a single fact table and OLAP cube. The i2 steps do not support Business Models that have been deployed to multiple fact tables and OLAP cubes.

To define the properties for the steps, right-click one of the i2 Steps on the canvas, and click **Properties**, or double-click one of the i2 Steps.

In the Properties window for the step that loads the i2 dimension and fact tables, you must specify values for the following parameters:

Dimension name

The name of the dimension.

Dimension type

The type of dimension. It can be either Conformed or Non-Conformed.

Member File Name

The name of the file in the BPI Import directory that contains changes to members of the dimension.

Hierarchy File Name

The name of the file in the BPI Import directory that contains hierarchy changes for the dimension.

Alias File Name

The name of the file in the BPI Import directory that contains alias changes for the dimension.

Attribute File Name

The name of the file in the BPI Import directory that contains changes to the attributes of the dimension.

Business Model

The name of the business model that is associated with the step.

Identity Type

How the member data is imported:

- 0 By code
- 1 By unique name
- 2 By family

Hierarchy Format

Relationships among members of the hierarchy:

- 0 Parent-child relationship
- 1 Flat hierarchy relationship
- 2 Snow flake relationship

Quotes for String

Whether the data in the input file has quotation marks:

- True** Strings have quotation marks.
- False** Strings do not have quotation marks.

Fixed Length

Whether the data in the input file has a fixed length:

- True** Data has a fixed length.
- False** Data does not have a fixed length.

Delimiter

How data in the input file is separated:

- 0 The data is separated by commas (,).
- 1 The data is separated by pipe characters.
- 2 The data is separated by tab characters.

In the Properties window for the step that loads the i2 OLAP cube, you must specify values for the following parameters:

Business Model

The name of the business model that is associated with the step.

Definition File Name

The name of the file in the BPI Import directory that describes the dimensions and measures of the data contained in the data file.

Creating an i2 step

Data File Name

The name of the file in the BPI Import directory that contains changes to the fact table data.

Identity Type

How the member data is imported:

- 0 By code
- 1 By unique name
- 2 By family

Data File Format

The format for the input file:

- 0 Data is for a mono-column fact table.
- 1 Data is for a multi-column fact table.

Quotes for String

Whether the data in the input file has quotation marks:

- True** Strings have quotation marks.
- False** Strings do not have quotation marks.

Fixed Length

Whether the data in the input file has a fixed length:

- True** Data has a fixed length.
- False** Data does not have a fixed length.

Delimiter

How data in the input file is separated:

- 0 The data is separated by commas (,).
- 1 The data is separated by pipe characters.
- 2 The data is separated by tab characters.

Important: If you do not want to assign a value to a particular parameter, you must assign it the value NULL. You cannot leave it blank.

The Connector for i2 TradeMatrix BPI does not create the comma-separated variable (CSV) files that are used as source data for the i2 steps. The CSV files must exist in the BPI Import directory at the warehouse agent site before you can run the i2 steps.

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 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 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 which 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 information 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 illustrates 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.

ACF/VTAM	IBM
AISPO	IMS
AIX	IMS/ESA
AIX/6000	LAN Distance
AIXwindows	MVS
AnyNet	MVS/ESA
APPN	MVS/XA
AS/400	Net.Data
BookManager	OS/2
CICS	OS/390
C Set++	OS/400
C/370	PowerPC
DATABASE 2	QBIC
DataHub	QMF
DataJoiner	RACF
DataPropagator	RISC System/6000
DataRefresher	RS/6000
DB2	S/370
DB2 Connect	SP
DB2 Extenders	SQL/DS
DB2 OLAP Server	SQL/400
DB2 Universal Database	System/370
Distributed Relational Database Architecture	System/390
DRDA	SystemView
eNetwork	VisualAge
Extended Services	VM/ESA
FFST	VSE/ESA
First Failure Support Technology	VTAM
	WebExplorer
	WIN-OS/2

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

Microsoft, Windows, and Windows NT are trademarks or registered trademarks of Microsoft Corporation.

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

Tivoli and NetView are trademarks of Tivoli Systems Inc. in the United States, other countries, or both.

UNIX is a registered trademark in the United States, other countries or both and is licensed exclusively through X/Open Company Limited.

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

Index

A

adding data sources 426
administrative interface, starting 12
agent, warehouse 2
agent site 2
 configuration 14
 defining 13
agent site, default 3
agents 405
aggregating columns 235
analysis of variance 241
ANOVA transformer 241

B

backing up Data Warehouse Center 323
backward regression 259
Base Aggregate step 197
basic statistics 243
Berkeley sockets 416
business metadata 5
BVBESTATUS table
 and DataJoiner 112
 creating 123
 for OS/390 database 110

C

Calculate Statistics transformer 243
Calculate Subtotals transformer 245
calculated column
 definition 218
calculation scripts 281
CASE statement 214
catalog name 433
Change Aggregate step 200
changing your configuration 324
Chi-square transformer 248
choosing warehouses 105
Classic Connect
 data server 406
 Data Warehouse Center step 403
 nonrelational data mapper 410
 warehouse agents 405
Clean Data transformer 220
clean types 221
Client Access/400 107
client connection requests 409
close trace on write 435
code transformation 214

communications compound address field 415
communications options 414
compression 434
configuration files 438
configuration prerequisites 425
configurations 14
configuring
 data sources 424
 IMS 87
 Informix
 OS/2 93
 UNIX 87
 Windows NT 63
 local OS/390 client 414
 LU 6.2 communications 430
 LU 6.2 on OS/390 420
 LU 6.2 on Windows NT 420
 Microsoft Access 80
 Microsoft SQL Server
 UNIX 92
 Windows NT 76
 MicrosoftExcel 84
 ODBC drivers 433
 Oracle
 OS/2 96
 UNIX 90
 Windows NT 70
 prerequisite products 412
 Sybase
 OS/2 94
 UNIX 89
 Windows NT 67
 TCP/IP communications 428
 TCP/IP on OS/390 416
 TCP/IP on Windows NT 418
 VSAM 87
configuring Data Warehouse Center
 changing 324
 installation 339
 to work with OLAP Server 281
connecting
 to source
 DB2 family 49
 DB2 for VM 51
 DB2 for VSE 51
 DB2 Universal Database for AS/400 51

connecting (*continued*)
 to source (*continued*)
 DB2 Universal Database for OS/390 51
 to warehouse
 DB2 Common Server 105
 DB2 EEE 111
 DB2 for AS/400 107
 DB2 for OS/390 109
connection handler 407
connectivity
 establishing
 AS/400 agent 52
 OS/390 agent 54
connectivity requirements
 between remote databases 52
 between the warehouse server and the warehouse agent 16
connectors
 i2 441
control database
 export considerations 325
 initializing 337
 installing a new one 337
controlling configuration
 parameters 408
Copy File using FTP warehouse program 189
Copying data
 between z/OS and OS/390 tables 188
correlation coefficient 251
Correlation transformer 251
covariance 252
creating target tables with
 DataJoiner 113
CROSS ACCESS ODBC driver 406
cross memory 414

D

data
 filtering 217
 informational 1
 inserting 167
 operational 1
 selecting 167
 transforming 211
 viewing
 source table 55
 target table 122

- Data cleansing, name and address cleansing 261
- Data export with ODBC to file warehouse program 173
- data mapper
 - description 410
 - workflow 411
- data server 406
- data source handler 409
- Data Warehouse Center 1
 - backing up 323
 - configuration files 438
 - environment variables 437
 - logger, starting 7
 - logging on 12
 - mapping to Information Catalog
 - Manager object types 332
 - OLE DB Support 439
 - properties, changing 339
 - security, defining 19
 - server, starting 7
 - starting a step from outside 162
 - status table
 - and DataJoiner 112
 - creating 123
 - token 312
- Data Warehouse Center program location 315
- DataJoiner
 - accessing source and target data 101, 111
 - creating target tables 113
 - defining
 - source 101, 111
 - warehouse 111
 - moving a target table 114
 - server mapping example 102
 - updating an existing table in a remote database 114
- DB2
 - on host systems, accessing as source 50
 - warehouses 105
- DB2 Common Server
 - accessing as target 105
 - connecting to warehouse 105
 - privileges 105
- DB2 Connect 106
- DB2 Connect gateway site 51
- DB2 EEE
 - warehouse target 111
- DB2 family
 - accessing as source 49
 - connecting to source 49
 - privileges 49
- DB2 for AS/400
 - accessing as targets 106
 - connecting to warehouse 107
 - privileges 106
- DB2 for AS/400 Data Load Insert warehouse program 175
- DB2 for AS/400 Data Load Replace warehouse program 180
- DB2 for EEE
 - connecting to warehouse 111
 - privileges 110
- DB2 for OS/390
 - connecting to warehouse 109
 - Grant to public property 110
 - privileges 109
 - warehouse target 110
- DB2 for OS/390 Load warehouse program 185
- DB2 for VM
 - accessing as source 50
 - connecting to source 51
 - privileges 50
 - setting up the DB2 Connect gateway site 51
- DB2 for VSE
 - accessing as source 50
 - connecting to source 51
 - Grant to public property 110
 - privileges 50
 - setting up the DB2 Connect gateway site 51
- DB2 OLAP Integration Server
 - exporting metadata to Data Warehouse Center 352
 - metadata mapping with Data Warehouse Center 375
 - multidimensional cube
 - loading from the Data Warehouse Center on a schedule 353
 - multidimensional cube, loading with data 350
- DB2 Universal Database export warehouse program 172
- DB2 Universal Database for AS/400
 - accessing as source 50
 - connecting to source 51
 - privileges 50
 - setting up the DB2 Connect gateway site 51
- DB2 Universal Database for OS/390
 - accessing as source 50
 - connecting to source 51
 - privileges 50
- DB2 Universal Database for OS/390 (*continued*)
 - setting up the DB2 Connect gateway site 51
- DB2 Universal Database load warehouse program 174
- default agent site 3
- defining
 - agent site 13
 - information resources 54
 - privileges
 - DB2 Common Server 105
 - DB2 family 49
 - DB2 for AS/400 106
 - DB2 for EEE 110
 - DB2 for OS/390 109
 - DB2 for VM 50
 - DB2 for VSE 50
 - DB2 Universal Database for AS/400 50
 - DB2 Universal Database for OS/390 50
 - process 126
 - program group 309
 - security 19
 - source
 - for use with DataJoiner 101, 111
- SQL
 - SQL 167
- step 128
 - Base Aggregate 197
 - Change Aggregate 200
 - Clean Data 220
 - Copy File using FTP 189
 - Data export with ODBC to file 173
 - DB2 for AS/400 Data Load Insert 175
 - DB2 for AS/400 Data Load Replace 180
 - DB2 for OS/390 Load 185
 - DB2 Universal Database export 172
 - DB2 Universal Database load 174
 - Generate Key Table 227
 - Generate Period Table 229
 - Invert Data 233
 - Pivot Data 235
 - Point-in-Time 197
 - Run FTP Command File 190
 - Staging Table 203
 - Submit OS/390 JCL jobstream 192

- defining (*continued*)
 - step (*continued*)
 - User Copy 197
 - user-defined program 310
 - warehouse
 - for use with DataJoiner 111
 - warehouse target
 - DB2 for EEE 111
 - DB2 for OS/390 110
 - in Data Warehouse Center 115
- defining source
 - replication 196
- defining step
 - User Copy 197
- developing steps 153
- development mode 153
- diagnosis 357
- dimension tables 216
- DJXSAMP 408
- DRDA 50
- DSNUTILS 128
- DWC
 - OLE DB Support 439
- E**
 - enable trace 435
 - enterprise server
 - definition 408
 - implementation 409
 - environment variables 437
 - updating 437
 - error handling 155
 - error messages 155
 - errors, logging 360
 - establishing connectivity 52
 - event viewer 358
 - example
 - external trigger program 165
 - examples
 - ANOVA transformer 241
 - DataJoiner server mapping 102
 - left outer join 214
 - mapping source data to target table 150
 - SQL DATE function 218
 - star join 217
 - star schema 216
 - summary 219
 - using warehouse program for data cleansing 211
 - WHERE clause 217
 - exponentially smoothed moving average 254
 - external trigger program 162
 - external trigger program (*continued*)
 - example 165
 - return code 165
 - status 165
 - externally populated step 161
- F**
 - fact tables 216
 - feedback file 318
 - fetch buffer size 433
 - files, configuration 438
 - filtering data 217
 - Fisher-F distributions 241
 - forecasting 254
 - foreign key 217
 - FormatDate transformer 238
 - formulas and expressions for data
 - cleansing 211
 - full-model regression 259
 - full outer join 215
- G**
 - Generate Key Table transformer 227
 - Generate Period Table transformer 229
 - generic ODBC driver 27
 - goodness-of-fit test 248
- H**
 - hostname 416
 - Hyperion Essbase warehouse program 281
- I**
 - i2
 - steps 128
 - import
 - from one Data Warehouse Center system to another 327
 - initializing control database 327
 - object definitions 327
 - import, importing, tag language file, DWC 274
 - import, tag language file, ICM, Information Catalog Manager 275
 - IMS BMP/DBB
 - initialization service 407
 - interface 408
 - IMS connectivity requirements 31, 37, 42, 46
 - IMS DRA
 - initialization service 407
 - interface 408
 - IMS logical table 403
 - incremental commit 170
- information catalog 5
- Information Catalog Manager
 - publishing metadata 331
- Information Catalog Manager object types
 - mapping to Data Warehouse Center object types 332
- informational data 1
- Informix 7.2 connectivity
 - requirements 34, 40, 44, 48
- initialization
 - definition 337
 - parameters 337
 - types 337
- initialization services 407
- initializing second control database 337
- inserting data 167
- installing prerequisite products 412
- Invert Data transformer 233
- IP address 416
- IWH2LOG.LOG 361
- IWH2LOGC.LOG 361
- IWH2SERV.LOG 361
- J**
 - joining source columns 211
 - joins
 - full outer join 215
 - left outer join 215
 - outer join 214
 - right outer join 215
 - star join 216
- K**
 - kernel host name 339
 - key column 227
 - key values 227
 - keys
 - foreign 217
 - primary 216
- L**
 - LAN connectivity requirements 35, 40, 45, 48
 - left outer join 214
 - load balancing 410
 - load rules 281
 - log 357
 - Log files
 - MQSeries 307
 - log level 360
 - log table 156, 360
 - logger service 408
 - logging errors 360

- logging onto the Data Warehouse Center desktop 12
- logical databases 404
- logical tables 404
- Logon window 12
- LU 6.2 configuration
 - example 422
 - worksheet 423

M

- mapping nonrelational data 404, 410
- mapping steps to data sources 150
- message pool size 434
- message type 156
- messages 155
- messaging
 - MQSeries 304
- metadata 5
 - exporting and importing, overview 324
 - exporting to a tag language file 325
 - mapping Data Warehouse Center with DB2 OLAP Integration Server 375
 - mapping Data Warehouse Center with Information Catalog Manager 363
 - mapping with Information Catalog Manager and OLAP server 373
 - propagating deleted objects to the information catalog 336
 - publishing to an information catalog 331
 - setting up environment to publish to information catalog 331
 - synchronizing with the Information Catalog Manager 335
 - updating in information catalog
 - log files created 336
 - updating in the information catalog 335
- metadata, importing for i2 sources 441
- metadata grammar 410
- Microsoft Access
 - connectivity requirements 35
- Microsoft Excel
 - connectivity requirements 36
- Microsoft SQL Server connectivity requirements 35, 40, 44

- migrating from Visual Warehouse
 - Host Adapters 435
- moving a target table from DataJoiner to remote database 114
- Moving Average transformer 254
- moving data
 - replicating 194
- MQSeries
 - creating views 304
 - error log file 307
 - importing 305
 - installation requirements 305
 - user-defined program 306
 - using with the Data Warehouse Center 304
 - XML metadata and 305
- MTO interface 407
- multidimensional cube
 - loading with data 350

N

- Name and address cleansing, Trillium 261
- non-warehouse sources
 - supported versions and release levels 27
- nonparametric tests 248
- nonrelational data 403
- nonrelational data mapper 410
- notebook
 - Program
 - Agent sites page 311
 - Parameters page 312

O

- Object REXX for Windows 315
- ODBC
 - generic driver 27
- ODBC driver 406
- OLAP server
 - metadata mapping with Information Catalog Manager 373
- OLAP Server: Calc with calc rules (ESSCALC2) warehouse program 283
- OLAP Server: Default calc (ESSCALC1) warehouse program 282
- OLAP Server: Free text data load warehouse program 284
- OLAP Server: Load data from a file without using load rules (ESSDATA4) warehouse program 288

- OLAP Server: Load data from file with load rules (ESSDATA2) warehouse program 285
- OLAP Server: Load data from SQL table with load rules (ESSDATA3) warehouse program 287
- OLAP Server: Update outline from file (ESSOTL1) warehouse program 289
- OLAP Server: Update outline from SQL table (ESSOTL2) warehouse program 290
- OLAP Server warehouse program 281
- OLE DB Support 439
- opening process 127
- operational data 1
- Oracle 7 connectivity requirements 34, 39, 44, 48
- OS/390
 - copying table data 188
- OS/390 client application 408
- OS/390 file connectivity requirements 32, 38, 43, 47
- outer join 214
- outline 281
- output log table 156
- overview of warehousing 1
- overwrite existing log 435

P

- p-value 241
- P-value 252
- parameter substitution 312
- password for the control database 339
- Pearson product-moment correlation coefficient 251
- period table 229
- Pivot Data transformer 235
- Point-in-Time step 197
- port number 417
- primary key 216
- printing
 - steps 128
- privileges
 - DB2 Common Server warehouses 105
 - DB2 family source 49
 - DB2 for AS/400 warehouse 106
 - DB2 for EEE warehouse 110

privileges (*continued*)
 DB2 for OS/390
 warehouse 109
 DB2 for VM 50
 DB2 for VSE 50
 DB2 Universal Database for
 AS/400
 source 50
 DB2 Universal Database for
 OS/390 50
 for DB2 warehouses 105
 process
 adding
 sources 127
 targets 127
 defining 126
 opening 127
 process (in warehousing) 3
 program group
 defining 309
 Program notebook
 Agent sites page 311
 Parameters page 312
 program step 4
 project planning
 choosing warehouses 105
 selecting sources 27
 promoting step
 production mode 161
 test mode 154
 pseudo-relational data 404

Q
 query processor 408

R
 recovery
 backing up Data Warehouse
 Center 323
 using log files to recover 357
 region controller 407
 regression, backward 259
 regression, full-model 259
 Regression transformer 259
 relational queries 403
 replicating tables 194
 response time out 434
 right outer join 215
 rolling sum 254
 Run FTP Command File warehouse
 program 190
 running subtotal 245

S
 sample data
 viewing
 source table 55
 target table 122
 SAP
 steps 128
 scheduling step 157
 security 19
 SELECT statement 312
 selecting data 167
 selecting sources 27
 server schedule mode 339
 setting
 log directory 360
 setting the database catalog
 options 429, 432
 setting up a warehouse 7
 simple moving average 254
 SNA protocol 415
 source 14
 DB2
 defining 54
 file
 defining 60
 IMS 87
 Informix
 OS/2 93
 UNIX 87
 Windows NT 63
 linking to steps 145
 Microsoft Access 80
 Microsoft Excel 84
 Microsoft SQL Server
 UNIX 92
 Windows NT 76
 non-DB2
 defining 97
 Oracle
 OS/2 96
 UNIX 90
 Windows NT 70
 replicating 194
 replication 196
 Sybase
 OS/2 94
 UNIX 89
 Windows NT 67
 types 27
 VSAM 87
 source columns
 mapping to target columns 150
 sources
 adding 127
 DB2 family products 49

sources (continued)
 DB2 for VM 50
 DB2 for VSE 50
 DB2 Universal Database for
 AS/400 50
 DB2 Universal Database for
 OS/390 50
 IMS 31, 37, 42, 46
 Informix 7.2 34, 40, 44, 48
 LAN file 35, 40, 45, 48
 Microsoft Access
 connectivity requirements 35
 Microsoft Excel
 connectivity requirements 36
 Microsoft SQL server 35, 40, 44
 non-DB2, connectivity 63
 Oracle 7 34, 39, 44, 48
 OS/390 32, 38, 43, 47
 setting up
 access 27
 DB2 Connect gateway site 51
 Sybase 33, 39, 43, 47
 VM file 33, 38, 43, 47
 VSAM 32, 38, 42, 47

SQL
 aggregation functions 219
 CASE statement 214
 GROUP BY clause 219
 JOIN clause 214
 OUTER JOIN keyword 214
 SELECT statement 312
 SUM function 219
 WHERE clause 217
 WHERE clause and data
 cleansing 211
 SQL codes 155
 SQL step 4
 Staging Table step 203
 standard configuration files 416
 star join 216
 star schema 5, 216
 creating 341
 description of 341
 starting
 warehouse agent daemon
 AS/400 9
 OS/2 8
 OS/390 9
 Windows NT 8
 starting Data Warehouse Center
 administrative interface 12
 logger 7
 server 7
 statistical transformers
 ANOVA transformer 241

- statistical transformers *(continued)*
 - Calculate Statistics transformer 243
 - Calculate Subtotals transformer 245
 - Chi-square transformer 248
 - Correlation transformer 251
 - Moving Average transformer 254
 - Regression transformer 259
- statistics, calculating 243
- step
 - adding 128
 - ANOVA transformer 241
 - Base Aggregate 197
 - Calculate Statistics transformer 243
 - Calculate Subtotals transformer 245
 - Change Aggregate 200
 - Chi-square transformer 248
 - Clean Data 220
 - code transformation 214
 - Copy File using FTP 189
 - Correlation transformer 251
 - Data export with ODBC to file 173
 - DB2 for AS/400 Data Load
 - Insert 175
 - DB2 for AS/400 Data Load
 - Replace 180
 - DB2 for OS/390 Load 185
 - DB2 Universal Database
 - export 172
 - DB2 Universal Database
 - load 174
 - developing 153
 - externally populated 161
 - Generate Key Table 227
 - Generate Period Table 229
 - Invert Data 233
 - linking to sources and targets 145
 - mapping source columns to target columns 150
 - mode
 - definition 153
 - development 153
 - test 153
 - Moving Average transformer 254
 - OLAP Server: Calc with calc rules (ESSCALC2) warehouse program 283

- step *(continued)*
 - OLAP Server: Default calc (ESSCALC1) warehouse program 282
 - OLAP Server: Free text data load warehouse program 284
 - OLAP Server: Load data from a file without using load rules (ESSDATA4) warehouse program 288
 - OLAP Server: Load data from file with load rules (ESSDATA2) warehouse program 285
 - OLAP Server: Load data from SQL table with load rules (ESSDATA3) warehouse program 287
 - OLAP Server: Update outline from file (ESSOTL1) warehouse program 289
 - OLAP Server: Update outline from SQL table (ESSOTL2) warehouse program 290
 - Pivot Data 235
 - Point-in-Time 197
 - promoting
 - production mode 161
 - test mode 154
 - Regression transformer 259
 - Run FTP Command File 190
 - running 153
 - scheduling 157
 - SQL
 - selecting and inserting data 167
 - Staging Table 203
 - starting from outside the Data Warehouse Center 162
 - status
 - and user-defined program feedback 320
 - Submit OS/390 JCL jobstream 192
 - subtypes 131
 - summarization 219
 - testing 153, 154
 - use of user-defined program 313
 - used to filter source data 217
 - User Copy 197
- step, creating
 - i2 442
- step (in warehousing) 3
- steps
 - i2 128

- steps *(continued)*
 - printing 128
 - SAP 128
 - Web traffic 128
- stored procedure 316
- subject area 2
- Submit OS/390 JCL jobstream warehouse program 192
- subsystem interfaces 408
- subtotal, calculating 245
- summarization 219
- Sybase connectivity
 - requirements 33, 39, 43, 47
- system process 156

T

- T-value 252
- table space name 360
- tag language file 325
- tag language file, DWC, ICM, ERwin file, ER1 file 272
- target 14
 - linking to steps 145
- target columns
 - mapping to source columns 150
- target database, export
 - considerations 325
- target table
 - creating with DataJoiner 113
 - moving 114
 - updating existing table in a remote database 114
- targets
 - adding 127
- Targets
 - supported versions and releases 115
- tasks
 - warehousing 5
- TCP/IP
 - configuration example 419
 - configuration information 438
 - configuration worksheet 419
 - protocol 415
- test mode 153
- testing
 - step 153
 - user-defined program 315
- testing step 154
- time-series analysis 254
- token 312
- trace
 - directory 339
 - file name 435
 - level 434

- trace (*continued*)
 - levels 339
- transformer step 4
- transformers, setting up 129
- transforming
 - code 214
 - data 211
- U**
- UDP
 - MQSeries 306
- updating an existing table in a remote database 114
- updating environment variables 437
- User Copy step 197
- user-defined function 238
- user-defined program
 - and SELECT statement 312
 - and step status 320
 - changing agent to user process 156
 - defining 310
 - definition 309
 - feedback 317
 - Object REXX for Windows 315
 - parameters 316
 - return code 317
 - testing 315
 - writing 315
- User-defined program
 - MQSeries 306
- user-defined program step 4
- user process 156
- V**
- variables 437
 - environment 437
- verifying communication 10
- VM file connectivity requirements 33, 38, 43, 47
- VSAM
 - interface 408
 - logical table 403
- VSAM connectivity requirements 32, 38, 42, 47
- W**
- warehouse 1
 - choosing 105
 - DB2 Common Server products 105
 - DB2 EEE privileges 110
- warehouse (*continued*)
 - DB2 for AS/400
 - DB2 Connect gateway site 107
 - privileges 106
 - DB2 for OS/390
 - privileges 109
 - mapping to source data 150
 - program
 - definition 126
 - for data cleansing 211
 - setting up 7
 - setting up access 105
 - supported databases 115
 - warehouse agent 2
 - warehouse agent, local 2
 - warehouse agent, remote 2
 - warehouse agent daemon
 - AS/400
 - starting 9
 - verifying activity 9
 - OS/2
 - starting 8
 - OS/390
 - starting 9
 - stopping 11
 - Windows NT
 - starting 8
 - Warehouse Manager
 - Connector for i2 TradeMatrix BPI
 - creating i2 steps 442
 - importing metadata for i2 sources 441
- warehouse process 3
- warehouse program
 - Copy File using FTP 189
 - Data export with ODBC to file 173
 - DB2 for AS/400 Data Load
 - Insert 175
 - Replace 180
 - DB2 for OS/390 Load 185
 - DB2 Universal Database
 - export 172
 - load 174
 - OLAP Server 281
 - OLAP Server: Calc with calc rules (ESSCALC2) 283
 - OLAP Server: Default calc (ESSCALC1) 282
 - OLAP Server: Free text data load 284
- warehouse program (*continued*)
 - OLAP Server: Load data from a file without using load rules (ESSDATA4) 288
 - OLAP Server: Load data from file with load rules (ESSDATA2) 285
 - OLAP Server: Load data from SQL table with load rules (ESSDATA3) 287
 - OLAP Server: Update outline from file (ESSOTL1) 289
 - OLAP Server: Update outline from SQL table (ESSOTL2) 290
 - parameters 312
 - Run FTP Command File 190
 - Submit OS/390 JCL
 - jobstream 192
 - use in steps 313
- warehouse schema
 - adding tables and views 344
 - autojoining tables 345
 - defining 343
 - exporting to the DB2 OLAP Integration Server 346
 - joining tables 345
 - publishing metadata about 355
 - working with in DB2 OLAP Integration Server 348
- warehouse source 2
 - defining 54
- warehouse sources
 - supported versions and release levels 27
- warehouse step 3
 - program 4
 - SQL 4
 - transformer 4
 - user-defined program 4
- warehouse target 2
 - defining 115
- warehouse transformers
 - Clean Data 220
 - Generate Key Table 227
 - Generate Period Table 229
 - Invert Data 233
 - Pivot Data 235
- warehousing
 - objects 2
 - overview 1
 - tasks 5
- warning messages 155
- Web traffic
 - steps 128
- WHERE clause 217

WHERE clause, use in data
cleansing 211
window
Logon 12
Windows NT
event viewer 358
WLM initialization service 407
writing
user-defined program 315

X

XTClient
syntax 163

Z

z/OS
copying table data 188

Contacting IBM

Before you contact DB2 customer support, check the *Troubleshooting Guide* for help with your specific technical problem. If you still need help, follow the procedures for gathering necessary information before you contact DB2 customer support.

For information or to order any of the DB2 Universal Database products, contact an IBM representative at a local branch office or contact any authorized IBM software remarketer.

If you live in the U.S.A., then you can call one of the following numbers:

- 1-800-237-5511 for customer support
- 1-888-426-4343 to learn about available service options

Product information

DB2 Universal Database product information is available by telephone or by the World Wide Web at <http://www.ibm.com/software/data/db2/udb/winos2unix/support>

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 <http://www.ibm.com/planetwide>

In some countries, IBM-authorized dealers should contact their dealer support structure for information.



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC26-9993-01



Spine information:



IBM® DB2® Universal
Database

Data Warehouse Center Admin Guide

Version 7