IBM Visual Warehouse for Windows NT

# Managing Visual Warehouse

*Version 5 Release 2*

GC26-8822-02

IBM Visual Warehouse for Windows NT

# Managing Visual Warehouse

*Version 5 Release 2*

GC26-8822-02

**Note**

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 171.

**Third Edition (November 1998)**

This edition replaces and makes obsolete the previous edition, GC26-8822-01. The technical changes for this edition are indicated by a vertical bar to the left of a change.

This edition applies to Version 5 Release 2 of Visual Warehouse (5639-VW5), and to any subsequent releases until otherwise indicated in new editions or technical newsletters. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality.

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.

# Contents

# About this book

This book describes the steps that are required to use IBM® Visual Warehouse™ to build and maintain a datamart. A *datamart* is a database that contains informational data that is replicated from your operational data sources.

This book also provides information about using the Visual Warehouse transformers. It contains information about the features and functions of the stored procedures and user-defined functions that are available as transformers in Visual Warehouse.

## Who should read this book

This book is intended for the administrators and datamart designers who use Visual Warehouse to build a datamart. 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 *Planning and Installing Visual Warehouse and DataGuide* to install Visual Warehouse and DataGuide®.

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

For more information about recommended books, see “Bibliography” on page 173.

## How to send your comments

Your feedback helps IBM to provide quality information. Please send any comments that you have about this book or any other Visual Warehouse documentation. Send your comments from the Web. Visit the Web site at http://www.software.ibm.com/data/vw.

The Web site has a feedback page that you can use to enter and send comments.

# Chapter 1. Introducing Visual Warehouse

IBM’s Visual Warehouse provides a complete, scalable datamart solution. It provides access to a wide variety of heterogeneous data and lets you model your business rules and processes through its business views.

## Business views

A *business view* is a logical entity in Visual Warehouse that defines:

- The structure of the output table or file.
- The mechanism (Structured Query Language (SQL) or a program) for populating the output table or file.
- The schedule by which the output table or file is populated.

In general, a business view populates an output (target) table in a warehouse database by extracting data from one or more input (source) tables or files. However, you can also define a business view for launching programs that does not specify source or target tables or files.

Business views move data and transform data by using SQL or by calling programs. When you run a business view, the transfer of data between the source and the target, and any transformation of that data, takes place.

You can run a business view on demand, or you can schedule a business view to run at a set time. You can schedule a business view to run one time only, or you can schedule it to run repeatedly, such as every Friday.

Suppose that you want Visual Warehouse to perform the following process:

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 use separate business views to extract the data from the databases and convert it into the format of the target table. You would then use another business view to extract the data from each of these business views and write the data to a new target table.

You could schedule one of the business views that extracts data from the source databases to run at a set time. You could schedule another business view that extracts source data to run after the first business view. You schedule all the business views in sequence until they extract all the required data from the database. You could schedule the business view that writes to the data warehouse to run after the last business view that extracts data has run.

## Information resources

An *information resource* is the Visual Warehouse definition of a resource from which a business view extracts data, or to which a business view writes data. In general, information resources correspond to operational data stores, which provide source data for the warehouse. However, you can define an information resource for a temporary data file:

- To which a business view writes data
- From which a business view reads data

Visual Warehouse lists information resources on the Sources page of the Visual Warehouse desktop.

## Warehouses

A *warehouse* is the Visual Warehouse definition of the database that is used as the data warehouse. The last business view in a series of business views writes data to a *target table*, which is a table in the warehouse.

When a business view runs, it can save data in the following ways:

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

A *business view edition* is a snapshot of the data in the information resource at a particular time. You can save multiple editions for a business view to provide a record of how the data changed over time.

Visual Warehouse lists warehouses on the Warehouses page of the Visual Warehouse desktop.

## Visual Warehouse programs

A *Visual Warehouse program* is a logical entity within Visual Warehouse that represents an application that you want Visual Warehouse to start. The Visual Warehouse agent can start a Visual Warehouse program:

- During the population of a target table.
- After the population of a target table.

- By itself.

For example, you can write a Visual Warehouse program to perform the following process:

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

## Agent sites

A *Visual Warehouse agent* is a program that transfers data between an information resource and the warehouse. When Visual Warehouse processes a business view, the agent uses ODBC drivers to populate the business view. The agent chooses the appropriate driver to connect to the source. Visual Warehouse can generate additional instances of an agent if multiple business views that require the same agent are processing simultaneously.

*Local agents* are agents that are installed on the same machine as the Visual Warehouse server.

An *agent site* is a logical name for a workstation where agent software is installed. You use this logical name to refer to the agent site throughout Visual Warehouse. The agent site name is not the same as the Transmission Control Protocol/Internet Protocol (TCP/IP) host name. A single physical machine can have only one TCP/IP host name. However, you can define multiple agent sites on a single machine. A logical name identifies each agent site.

The *default agent site*, named the Default VW AgentSite, is a local agent that Visual Warehouse defines during initialization of the Visual Warehouse control database.

## Subjects

A *subject* is a logical collection of business views that does not need to correspond to a physical database. For example, you have a series of business views that work together, using Visual Warehouse programs, to unload, cleanse, and load data. You can use a subject to identify these programs as a set of programs.

## The Visual Warehouse architecture

Visual Warehouse includes several components:

- Visual Warehouse server
- Visual Warehouse administrative client
- Visual Warehouse agents
- Visual Warehouse control database
- Visual Warehouse information catalog (DataGuide)
- Visual Warehouse for the Web
- DataGuide for the Web

Figure 1 shows how the components in Visual Warehouse relate to each other.

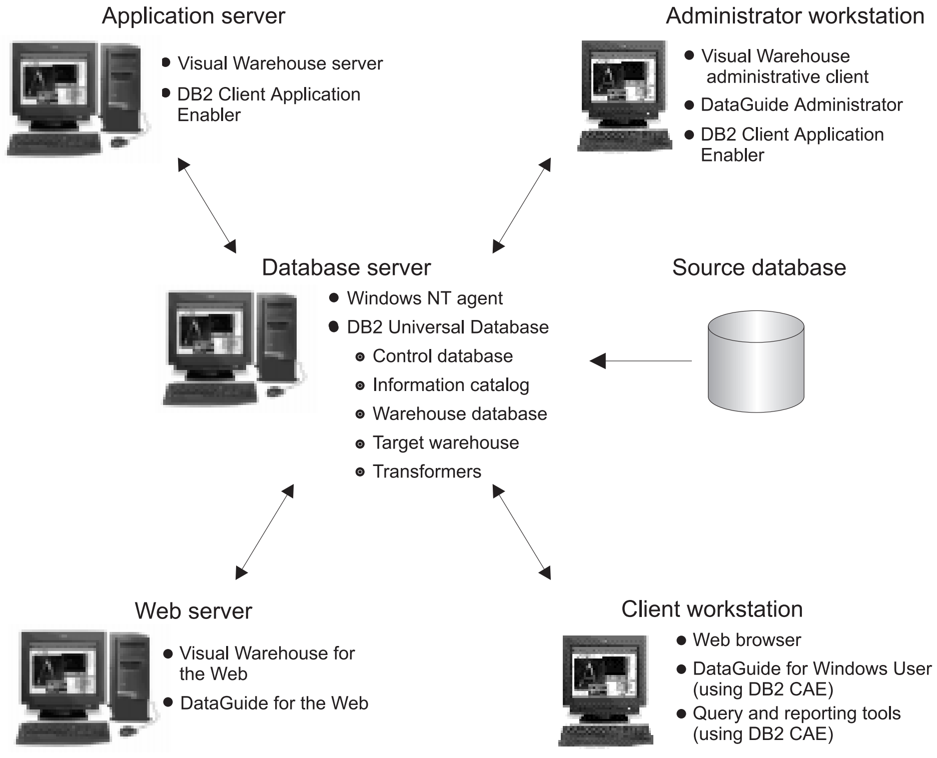

Application server
● Visual Warehouse server
● DB2 Client Application Enabler
Administrator workstation
● Visual Warehouse administrative client
● DataGuide Administrator
● DB2 Client Application Enabler
Database server
● Windows NT agent
● DB2 Universal Database
◎ Control database
◎ Information catalog
◎ Warehouse database
◎ Target warehouse
◎ Transformers
Source database
Web server
● Visual Warehouse for the Web
● DataGuide for the Web
Client workstation
● Web browser
● DataGuide for Windows User (using DB2 CAE)
● Query and reporting tools (using DB2 CAE)

*Figure 1. The Visual Warehouse architecture*

For a description of each component, see *Planning and Installing Visual Warehouse and DataGuide* .

# Chapter 2. Setting up access to data sources and targets

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

- Determine which data sources you will use in your data warehouse or your datamart.
- Select target warehouses that you will use for your datamart.

This chapter describes the types of data sources and target warehouses that work with Visual Warehouse and tells you how to set up access to them.

## Selecting Visual Warehouse data sources

Visual Warehouse supports a wide variety of relational and nonrelational data sources. You can fill your Visual Warehouse datamart 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 Visual Warehouse, see "Using DataJoiner as a Visual Warehouse source or target" on page 31.
- Multiple Virtual Storage (MVS®), Virtual Machine (VM), and local area network (LAN) files
- IMS® and Virtual Storage Access Method (VSAM) (with the Visual Warehouse Adapter option or Data Joiner Classic Connect)

For information about the versions and releases of these products, see *Planning and Installing Visual Warehouse and DataGuide*.

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

Visual Warehouse also includes several Intersolv ODBC drivers that you can use to access non-IBM data. To find out which database data types are supported by each driver, see the IVDRV*nn*.HLP file in the vwslib/iwh/odbc32 directory. (The letters nn in the file name represent a two-digit number that identifies the most recent driver set.) This file also provides ODBC connect strings.

Certain agents support certain data sources. Table 1 summarizes which agents support which data sources.

*Table 1. Agent support for data sources*

| Data Source | Windows NT agent | AIX agent | Sun Solaris agent | IBM Operating System/2 (OS/2) agent |
|---|---|---|---|---|
| DB2 family | ✓ | ✓ | ✓ | ✓ |
| DataJoiner | ✓ | ✓ | | |
| Oracle | ✓ | ✓ | ✓ | ✓ |
| Sybase | ✓ | ✓ | ✓ | ✓ |
| Microsoft SQL Server | ✓ | | | ✓ |
| Informix | ✓ | ✓ | ✓ | |
| Generic ODBC driver | ✓ | ✓ | ✓ | ✓ |
| Flat file LAN | ✓ | ✓ | ✓ | ✓ |
| IMS | ✓ | ✓ | ✓[1] | ✓[1] |
| VSAM | ✓ | ✓ | ✓[1] | ✓[1] |

[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.

For a list of the data sources that are supported by the AS/400® agent, see *Installing and Using the Visual Warehouse AS/400 Agent*.

## Setting up access to DB2 UDB source databases

You can use any DB2 Universal Database® (UDB) database as a source database for your datamart. For more information about using DB2 UDB, see *Installing and Configuring DB2 Clients.*

To set up access to DB2 UDB source databases:

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

### Defining privileges to DB2 UDB 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 UDB 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 with ODBC by using the ODBC Administrator. See “Appendix C. Registering an ODBC database” on page 159.
5. Bind database utilities and ODBC(CLI) to the database. Each type of client requires only one bind.

## Setting up access to DB2 DRDA source databases

To set up access to DB2 Distributed Relational Database Architecture™ (DRDA®) source databases:

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

## Setting up access to data sources and targets

### 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 for OS/400®
- DB2 for MVS
- DB2 for VM
- DB2 for VSE

To configure the server for DRDA, install 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 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 for MVS, 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 for OS/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, 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 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. See “Appendix C. Registering an ODBC database” on page 159.
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.

## Accessing non-DB2 data sources

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

## Setting up access to data sources and targets

There might be more than one way to access some data sources. For example, if you want to use an agent on Windows NT® to access IMS data, you can use the Visual Warehouse Adapter to access the IMS data. Or, if you are using agents on an AIX® system, you can use DataJoiner Classic Connect. If you are using the Sun Solaris or OS/2® agent, you can access the DataJoiner database on AIX remotely from the agent site.

The following sections provide information about connecting to data sources from Windows NT, AIX, Sun Solaris, and OS/2 agent sites. For information about connecting to data sources from AS/400 agent sites, see *Installing and Using the Visual Warehouse AS/400 Agent*.

### Windows NT

Table 2 lists the supported data sources on Windows NT and explains what you need to do to connect to those sources.

*Table 2. Connectivity requirements for supported data sources on Windows NT*

| **Source database or file:** | **Prerequisite products for Windows NT:** | **How to connect:** |
|---|---|---|
| IMS (MVS) | **Database access program:**<br>Visual Warehouse Adapter or DataJoiner and DataJoiner Classic Connect<br>**Source / agent connection:**<br>TCP/IP or APPC<br>**Client enabler program:**<br>None | 1. Establish a link from the agent site to the host.<br>2. Install and configure the adapter on the host.<br>3. Install and configure the Host Adapter Client.<br>4. Identify the user ID and password with access to the source database. |
| VSAM (MVS) | **Database access program:**<br>Visual Warehouse Adapter or DataJoiner and DataJoiner Classic Connect<br>**Source / agent connection:**<br>TCP/IP or APPC<br>**Client enabler program:**<br>None | 1. Establish a link from the agent site to the host.<br>2. Install and configure the adapter on the host.<br>3. Install and configure the Host Adapter Client.<br>4. Identify the user ID and password with access to the source database. |

*Table 2. Connectivity requirements for supported data sources on Windows NT (continued)*

| Source database or file: | Prerequisite products for Windows NT: | How to connect: |
|---|---|---|
| MVS flat file | **Database access program:**<br>FTP or NFS<br>**Source / agent connection:**<br>TCP/IP (FTP or NFS)<br>**Client enabler program:**<br>None | Establish a link from the agent to the host.<br>For more information about accessing flat files using NFS, see "Accessing files with NFS" on page 22.<br>For more information about accessing flat files using FTP, see "Accessing files with FTP" on page 23. |
| VM flat file | **Database access program:**<br>FTP or NFS<br>**Source / agent connection:**<br>TCP/IP (FTP or NFS)<br>**Client enabler program:**<br>None | Establish a link from the agent to the host.<br>For more information about accessing flat files using NFS, see "Accessing files with NFS" on page 22.<br>For more information about accessing flat files using FTP, see "Accessing files with FTP" on page 23. |
| Sybase | **Database access program:**<br>None<br>**Source / agent connection:**<br>TCP/IP<br>**Client enabler program:**<br>For Intel® systems, Sybase Open Client Library Version 10.03 (or later) and the appropriate Sybase Net-Library<br>For Alpha systems, Sybase Open Client Library Version 11.01 (or later) and the appropriate Sybase Net-Library | 1. Install the Open Client on the agent site.<br>2. Catalog the remote database according to the client enabler instructions.<br>3. Identify the user ID and password with access to the source database. |
| Oracle | **Database access program:**<br>None<br>**Source / agent connection:**<br>TCP/IP<br>**Client enabler program:**<br>Oracle SQL*Net V2 | 1. Install SQL*NET on the agent site.<br>2. Catalog the remote database according to client enabler instructions.<br>3. Identify the user ID and password with access to the source database. |

*Table 2. Connectivity requirements for supported data sources on Windows NT (continued)*

| Source database or file: | Prerequisite products for Windows NT: | How to connect: |
|---|---|---|
| Informix | **Database access program:**<br>None<br>**Source / agent connection:**<br>TCP/IP<br>**Client enabler program:**<br>Informix-Connect 7.2 for Windows NT | 1. Install Informix Connect on the agent site.<br>2. Catalog the remote database according to the client enabler instructions.<br>3. Identify the user ID and password with access to the source database. |
| Microsoft SQL Server | **Database access program:**<br>None<br>**Source / agent connection:**<br>TCP/IP<br>**Client enabler program:**<br>For access to a Version 6.0 DBMS, Microsoft SQL Server DB-Library and Net-Library Version 6.0<br>For access to a Version 7.0 DBMS, Microsoft SQL Server DB-Library and Net-Library Version 7.0 | 1. Install the Microsoft SQL Server DB-Library and Net-Library on the agent site.<br>2. Catalog the remote database according to client enabler instructions.<br>3. Identify the user ID and password with access to the source database. |
| LAN flat file | **Database access program:**<br>None<br>**Source / agent connection:**<br>TCP/IP<br>**Client enabler program:**<br>None | 1. Obtain the FTP server or mount string.<br>2. Identify the user ID and password with access to the source files.<br>For more information about accessing flat files using NFS, see "Accessing files with NFS" on page 22.<br>For more information about accessing flat files using FTP, see "Accessing files with FTP" on page 23. |

*Table 2. Connectivity requirements for supported data sources on Windows NT (continued)*

| Source database or file: | Prerequisite products for Windows NT: | How to connect: |
|---|---|---|
| LAN BLOB file | **Database access program:** None<br>**Source / agent connection:** TCP/IP<br>**Client enabler program:** None | 1. Obtain the FTP server or mount string.<br>2. Identify the user ID and password with access to the source files.<br>For more information about accessing flat files using NFS, see “Accessing files with NFS” on page 22.<br>For more information about accessing flat files using FTP, see “Accessing files with FTP” on page 23. |
| Microsoft Access | **Database access program:** None<br>**Source / agent connection:** ODBC<br>**Client enabler program:** None | 1. Use the generic ODBC connect string.<br>2. See the Microsoft Access help topics for a mapping of the ANSI SQL data types supported by Microsoft Access. A business view that uses Access for source data can use only functions that are supported by the Microsoft Access ODBC driver.<br>For detailed instructions about creating a datamart that uses Microsoft Access as a source database, see “Using Microsoft Access” on page 163. |
| Microsoft Excel | **Database access program:** None<br>**Source / agent connection:** ODBC<br>**Client enabler program:** None | 1. Use the generic ODBC connect string.<br>2. See the Microsoft Excel help topics for a mapping of the ANSI SQL data types supported by Microsoft Excel. A business view that uses Excel for source data can use only functions that are supported by the Microsoft Excel ODBC driver.<br>For detailed instructions about creating a datamart that uses Microsoft Excel as a source database, see “Using Microsoft Excel” on page 166. |

## AIX

Table 3 on page 14 lists the supported data sources on AIX and explains what you need to do to connect to those sources.

*Table 3. Connectivity requirements for supported data sources on AIX*

| Source database or file: | Prerequisite products for AIX: | How to connect: |
|---|---|---|
| IMS (MVS) | **Database access program:**<br>DataJoiner and DataJoiner Classic Connect<br>**Source / agent connection:**<br>TCP/IP or APPC<br>**Client enabler program:**<br>None | From the DataJoiner workstation:<br>1. Establish a link from the workstation to the host.<br>2. Install and configure the adapter on the host.<br>3. Identify the user ID and password with access to the source database.<br>From the agent site:<br>1. Catalog the node on which DataJoiner resides.<br>2. Catalog the DataJoiner database. |
| VSAM (MVS) | **Database access program:**<br>DataJoiner and DataJoiner Classic Connect<br>**Source / agent connection:**<br>TCP/IP or APPC<br>**Client enabler program:**<br>None | From the DataJoiner workstation:<br>1. Establish a link from the workstation to the host.<br>2. Install and configure the adapter on the host.<br>3. Identify the user ID and password with access to the source database.<br>From the agent site:<br>1. Catalog the node on which DataJoiner resides.<br>2. Catalog the DataJoiner database. |
| MVS flat file | **Database access program:**<br>FTP or NFS<br>**Source / agent connection:**<br>TCP/IP (FTP or NFS)<br>**Client enabler program:**<br>None | Establish a link from the agent site to the host.<br>For more information about accessing flat files using NFS, see “Accessing files with NFS” on page 22.<br>For more information about accessing flat files using FTP, see “Accessing files with FTP” on page 23. |

*Table 3. Connectivity requirements for supported data sources on AIX (continued)*

| Source database or file: | Prerequisite products for AIX: | How to connect: |
|---|---|---|
| VM flat file | **Database access program:**<br>FTP or NFS<br>**Source / agent connection:**<br>TCP/IP (FTP or NFS)<br>**Client enabler program:**<br>None | Establish a link from the agent site to the host.<br>For more information about accessing flat files using NFS, see "Accessing files with NFS" on page 22.<br>For more information about accessing flat files using FTP, see "Accessing files with FTP" on page 23. |
| Sybase | **Database access program:**<br>None<br>**Source / agent connection:**<br>ODBC<br>**Client enabler program:**<br>Intersolv Version 3.10 Driver Manager and Sybase driver | 1. Catalog the remote database according to the client enabler instructions.<br>2. Identify the user ID and password with access to the source database. |
| Oracle | **Database access program:**<br>None<br>**Source / agent connection:**<br>ODBC<br>**Client enabler program:**<br>Intersolv Version 3.1.0 Driver Manager and Oracle driver | 1. Catalog the remote database according to the client enabler instructions.<br>2. Identify the user ID and password with access to the source database. |
| Informix | **Database access program:**<br>None<br>**Source / agent connection:**<br>ODBC<br>**Client enabler program:**<br>Intersolv Version 3.0.1 Driver Manager and Informix driver | 1. Catalog the remote database according to the client enabler instructions.<br>2. Identify the user ID and password with access to the source database. |
| Microsoft SQL Server | Not supported | Not supported |

Table 3. Connectivity requirements for supported data sources on AIX (continued)

| Source database or file: | Prerequisite products for AIX: | How to connect: |
|---|---|---|
| LAN flat file | **Database access program:**<br>None<br>**Source / agent connection:**<br>TCP/IP<br>**Client enabler program:**<br>None | 1. Obtain the FTP server or mount string.<br>2. Identify the user ID and password with access to the source files.<br>For more information about accessing flat files using NFS, see "Accessing files with NFS" on page 22.<br>For more information about accessing flat files using FTP, see "Accessing files with FTP" on page 23. |
| LAN BLOB file | **Database access program:**<br>None<br>**Source / agent connection:**<br>TCP/IP<br>**Client enabler program:**<br>None | 1. Obtain the FTP server or mount string.<br>2. Identify the user ID and password with access to the source files.<br>For more information about accessing flat files using NFS, see "Accessing files with NFS" on page 22.<br>For more information about accessing flat files using FTP, see "Accessing files with FTP" on page 23. |
| Microsoft Access | Not supported | Not supported |

## Sun Solaris

Table 4 on page 17 lists the supported data sources on Sun Solaris and explains what you need to do to connect to those sources.

*Table 4. Connectivity requirements for supported data sources on Sun Solaris*

| Source database or file: | Prerequisite products for Sun Solaris: | How to connect: |
|---|---|---|
| IMS (MVS) | **Database access program:**<br>DataJoiner and DataJoiner Classic Connect<br>**Source / agent connection:**<br>ODBC (to DataJoiner database)<br>**Client enabler program:**<br>None | From the DataJoiner workstation:<br>1. Establish a link from the workstation to the host.<br>2. Install and configure the adapter on the host.<br>3. Identify the user ID and password with access to the source database.<br>From the agent site:<br>1. Catalog the node on which DataJoiner resides.<br>2. Catalog the DataJoiner database. |
| VSAM (MVS) | **Database access program:**<br>DataJoiner and DataJoiner Classic Connect<br>**Source / agent connection:**<br>ODBC (to DataJoiner database)<br>**Client enabler program:**<br>None | From the agent site:<br>1. Catalog the node on which DataJoiner resides.<br>2. Catalog the DataJoiner database. |
| MVS flat file | **Database access program:**<br>FTP or NFS<br>**Source / agent connection:**<br>TCP/IP (FTP or NFS)<br>**Client enabler program:**<br>None | Establish a link from the agent site to the host.<br>For more information about accessing flat files using NFS, see “Accessing files with NFS” on page 22.<br>For more information about accessing flat files using FTP, see “Accessing files with FTP” on page 23. |
| VM flat file | **Database access program:**<br>FTP or NFS<br>**Source / agent connection:**<br>TCP/IP (FTP or NFS)<br>**Client enabler program:**<br>None | Establish a link from the agent site to the host.<br>For more information about accessing flat files using NFS, see “Accessing files with NFS” on page 22.<br>For more information about accessing flat files using FTP, see “Accessing files with FTP” on page 23. |

*Table 4. Connectivity requirements for supported data sources on Sun Solaris (continued)*

| Source database or file: | Prerequisite products for Sun Solaris: | How to connect: |
|---|---|---|
| Sybase | **Database access program:** None<br>**Source / agent connection:** ODBC<br>**Client enabler program:** For Sybase 11, Intersolv Version 3.0.1 Driver Manager and Sybase driver | 1. Catalog the remote database according to the client enabler instructions.<br>2. Identify the user ID and password with access to the source database. |
| Oracle | **Database access program:** None<br>**Source / agent connection:** ODBC<br>**Client enabler program:** For Oracle Version 7.3.2, Intersolv Version 3.0.1 Driver Manager and Oracle driver | 1. Catalog the remote database according to the client enabler instructions.<br>2. Identify the user ID and password with access to the source database. |
| Informix | **Database access program:** None<br>**Source / agent connection:** ODBC<br>**Client enabler program:** Intersolv Version 3.0.1 Driver Manager and Informix driver | 1. Catalog the remote database according to the client enabler instructions.<br>2. Identify the user ID and password with access to the source database. |
| Microsoft SQL Server | Not supported | Not supported |
| LAN flat file | **Database access program:** None<br>**Source / agent connection:** TCP/IP<br>**Client enabler program:** None | 1. Obtain the FTP server or mount string.<br>2. Identify the user ID and password with access to the source files.<br>For more information about accessing flat files using NFS, see “Accessing files with NFS” on page 22.<br>For more information about accessing flat files using FTP, see “Accessing files with FTP” on page 23. |

*Table 4. Connectivity requirements for supported data sources on Sun Solaris (continued)*

| Source database or file: | Prerequisite products for Sun Solaris: | How to connect: |
|---|---|---|
| LAN BLOB file | **Database access program:** None<br>**Source / agent connection:** TCP/IP<br>**Client enabler program:** None | 1. Obtain the FTP server or mount string.<br>2. Identify the user ID and password with access to the source files.<br>For more information about accessing flat files using NFS, see "Accessing files with NFS" on page 22.<br>For more information about accessing flat files using FTP, see "Accessing files with FTP" on page 23. |
| Microsoft Access | Not supported | Not supported |

## OS/2

Table 5 lists the supported data sources on OS/2 and explains what you need to do to connect to those sources.

*Table 5. Connectivity requirements for supported data sources on OS/2*

| Source database or file: | Prerequisite products for OS/2: | How to connect: |
|---|---|---|
| IMS (MVS) | **Database access program:** DataJoiner and DataJoiner Classic Connect<br>**Source / agent connection:** ODBC (to DataJoiner database)<br>**Client enabler program:** None | From the agent site:<br>1. Catalog the node on which DataJoiner resides.<br>2. Catalog the DataJoiner database. |
| VSAM (MVS) | **Database access program:** DataJoiner and DataJoiner Classic Connect<br>**Source / agent connection:** ODBC (to DataJoiner database)<br>**Client enabler program:** None | From the agent site:<br>1. Catalog the node on which DataJoiner resides.<br>2. Catalog the DataJoiner database. |

*Table 5. Connectivity requirements for supported data sources on OS/2 (continued)*

| Source database or file: | Prerequisite products for OS/2: | How to connect: |
|---|---|---|
| MVS flat file | **Database access program:** FTP or NFS<br>**Source / agent connection:** TCP/IP (FTP or NFS)<br>**Client enabler program:** None | Establish a link from the agent site to the host.<br>For more information about accessing flat files using NFS, see "Accessing files with NFS" on page 22.<br>For more information about accessing flat files using FTP, see "Accessing files with FTP" on page 23. |
| VM flat file | **Database access program:** FTP or NFS<br>**Source / agent connection:** TCP/IP (FTP or NFS)<br>**Client enabler program:** None | Establish a link from the agent site to the host.<br>For more information about accessing flat files using NFS, see "Accessing files with NFS" on page 22.<br>For more information about accessing flat files using FTP, see "Accessing files with FTP" on page 23. |
| Sybase | **Database access program:** None<br>**Source / agent connection:** TCP/IP<br>**Client enabler program:** Sybase Open Client Library Version 10 for OS/2 | 1. Install the Open Client on the agent site.<br>2. Catalog the remote database according to the client enabler instructions.<br>3. Identify the user ID and password with access to the source database. |
| Oracle | **Database access program:** None<br>**Source / agent connection:** TCP/IP<br>**Client enabler program:** Oracle SQL*Net V2.1.4 for OS/2 | 1. Install SQL*NET on the agent site.<br>2. Catalog the remote database according to the client enabler instructions.<br>3. Identify the user ID and password with access to the source database. |
| Informix | Not supported | Not supported |

*Table 5. Connectivity requirements for supported data sources on OS/2 (continued)*

| Source database or file: | Prerequisite products for OS/2: | How to connect: |
|---|---|---|
| Microsoft SQL Server | **Database access program:** None<br>**Source / agent connection:** TCP/IP<br>**Client enabler program:** For access to a Version 6.0 DBMS, Sybase Open Client Version 10 for OS/2 | 1. Install the Open Client on the agent site.<br>2. Catalog the remote database according to the client enabler instructions.<br>3. Identify the user ID and password with access to the source database. |
| LAN flat file | **Database access program:** None<br>**Source / agent connection:** TCP/IP<br>**Client enabler program:** None | 1. Obtain the FTP server or mount string.<br>2. Identify the user ID and password with access to the source files.<br>For more information about accessing flat files using NFS, see "Accessing files with NFS" on page 22.<br>For more information about accessing flat files using FTP, see "Accessing files with FTP" on page 23. |
| LAN BLOB file | **Database access program:** None<br>**Source / agent connection:** TCP/IP<br>**Client enabler program:** None | 1. Obtain the FTP server or mount string.<br>2. Identify the user ID and password with access to the source files.<br>For more information about accessing flat files using NFS, see "Accessing files with NFS" on page 22.<br>For more information about accessing flat files using FTP, see "Accessing files with FTP" on page 23. |
| Microsoft Access | Not supported | Not supported |

## Accessing remote files

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

### Accessing flat files through a Windows NT file server

You can use data files as a source file for a business view. If the file is not on the agent site, but is accessed through a Windows NT 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 information resource for the file, specify the .bat file in the **Pre-Access Command** field on the Connection notebook page.

You can also define a similar .bat file to delete the link to the remote drive after Visual Warehouse processes the files. If you do this, specify the .bat file in the **Post-Access Command** field on the Connection page of the Information Resource notebook.

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 3.10 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 on the Connection page of the Information Resource notebook.

If the agent site does not have NFS installed, use the NET USE command to access NFS, as described in “Accessing flat files through a Windows NT file server” on page 22.

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 3.10 32–bit Textfile (*.*).

### Accessing files with FTP

You can use FTP to access data files on a remote workstation. Use FTP if the file is 20 megabytes or less. When you promote a business view that uses remote files to test status, the files will be transferred just as they would be if the business view was promoted to production status. 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 VWPRCPY instead of FTP (see “Accessing data files with VWPRCPY”).

To use FTP to access a data file:

1. Store the file on an FTP site.
2. Declare the file as a **Flat File Lan Resource** on the Connection page of the Information Resource notebook.
3. Select **FTP Copy** as your access method, and complete the required information.

When you promote the business view that uses this source to test status, Visual Warehouse transfers the file to a temporary file on the agent site.

### Accessing data files with VWPRCPY

You can use VWPRCPY to access data files on a remote workstation. Use VWPRCPY if the file is larger then 20 megabytes. Visual Warehouse does not run Visual Warehouse programs when a business view is promoted to test status, so the file will not be transferred. You can also specify the location of the target file for VWPRCPY.

To use VWPRCPY to access a file:

1. Declare the file as a Flat File Lan Resource on the Connection page of the Information Resource notebook.
2. Select **Local File** as your access method, and complete the required information.

3. Define two business views to access a file of this size:
   a. Define the first business view to use the VWPRCPY Visual Warehouse program.

      Use this business view to copy the file to the agent site.
   b. Define the second business view to use the information resource that you create for the file.

      The business view will access the file as a local file.

      This file is the output file of the first business view.

## Setting up access to a target warehouse

Each set of users will access a target warehouse, which is the database that contains the results of the business views after they run. The operating system you choose for the target warehouse depends on the operating system that the users work on, and on your network constraints. The DB2 family of databases can handle a wide variety of capacity requirements, so select the database that is appropriate for your warehouse.

Datamart users must have access to the warehouse database. They can use any end-user application that can access DB2 data, such as Lotus Approach.

### Setting up a DB2 UDB target warehouse

You can use any DB2 UDB database as a target warehouse for your datamart. For more information about connecting to DB2 UDB databases through an UDB server or through CAE, see *Installing and Configuring DB2 Clients*.

#### Defining privileges to DB2 UDB target warehouses

Ask the system administrator of the target system to do the following tasks:

1. Create a database to use as the target warehouse.
2. 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 UDB target warehouses

After you have defined 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 if the database is remote.
3. Catalog the database.
4. Register the database as a system DSN by using the ODBC Administrator. See “Appendix C. Registering an ODBC database” on page 159.
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 OS/400 target warehouse

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

### Using DB2 Connect

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

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

**Prerequisite products:** If you want to access data from a DB2 for OS/400 target 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 OS/400 target 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 OS/400 collections.

**Setting up the DB2 Connect gateway site:** After you have defined 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 OS/400 target 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. See “Appendix C. Registering an ODBC database” on page 159.
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 OS/400 database:

1. Install the following TCP/IP PTFs on an AS/400 machine that runs OS/400 V3.2:
   - SF36159
   - MF14406
   - MF14679
2. Ensure that the AS/400 QUSER password is set (not expiring). You can set the password to no password, if required.

3. Enter the TCP/IP host name for the AS/400 machine 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
   ```

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

   ```
   cwbping hostname ip
   ```

   You should see 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 machine to start the servers:

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

6. 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.
7. Set the DB2CODEPAGE environment variable to 850 in the Windows NT system variables (not user variables).
8. 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.as400.ibm.com/clientaccess/service.htm
   ```

9. 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)
   ```

10. 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

11. 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 target warehouse

By default, Visual Warehouse creates warehouse target tables in a format that is supported by the DB2 UDB products. You also can tailor business views to work with target tables in DB2 for OS/390®.

### Defining privileges to DB2 for OS/390 target warehouses

Before you begin, ask the administrator of the DB2 for OS/390 system to perform the following tasks:

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

### Establishing connectivity to the DB2 for OS/390 target warehouse database

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

- Set up access to DB2 for MVS.
- Connect to the target warehouse database.

## Creating the Visual Warehouse status table

Create the Visual Warehouse status table by using the following command:

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

where:

- *database-name* is the name of the target warehouse database
- *tablespace-name* is the name of the table space that contains the database

## Defining the warehouse to Visual Warehouse

After the system is set up, perform the following tasks in Visual Warehouse to use a DB2 for MVS target warehouse:

1. Create a warehouse within the datamart.
2. Open the warehouse by double-clicking the warehouse name in the Warehouses Desktop.
3. Create a business view.
4. After the Business View notebook opens, click the **Information** tab.
5. Click **Create DDL**.

   The BV Create Statement window opens.
6. Add the following clause to the end of the CREATE statement in the **Create Statement** field:

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

   where:

   - *database-name* is the name of the target warehouse database
   - *tablespace-name* is the name of the table space that contains the database
7. Click **OK**.
8. After you finish creating the business view, click **OK** to save changes and close the Business View notebook.

   When you promote the business view to test status, Visual Warehouse creates the target table in the DB2 for OS/390 database.
9. Repeat steps 3 through 8 for each business view in the warehouse.

## Setting up a DB2 EEE target warehouse

By default, Visual Warehouse creates warehouse target tables in a format that is supported by the DB2 UDB products. You also can tailor business views to work with target tables in DB2 Extended Enterprise Edition (DB2 EEE).

### Defining privileges to the DB2 EEE database

Before you begin, ask the administrator of the DB2 EEE system to perform the following tasks:

- Create a database, node group, and table space to use as the target 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 UDB target warehouses" on page 25, starting with the explicit SELECT privilege.

### Defining the DB2 EEE database to Visual Warehouse

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

1. Create a warehouse within the datamart.
2. Open the warehouse by double-clicking the warehouse name in the Warehouses desktop.
3. Create a business view.
4. After the Business View notebook opens, click the **Information** tab.
5. Click **Create DDL**.

   The BV Create Statement window opens.
6. Add the following clause to the end of the CREATE statement in the **Create Statement** field:

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

   where:

   - *tablespace-name* is the name of the table space that contains the database
   - *column-name* is the name of the column to use for partitioning
7. Click **OK**.
8. After you finish creating the business view, click **OK** to save changes and close the Business View notebook.

When you promote the business view to test status, Visual Warehouse creates the target table in the DB2 EEE database.

9. Repeat steps 3 on page 30 through 8 on page 30 for each business view in the warehouse.

## Using DataJoiner as a Visual Warehouse source or target

DataJoiner provides several advantages for accessing data for business view processes. 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 Visual Warehouse business views that take advantage of DataJoiner's benefits. First, you define warehouses that use DataJoiner databases. Then you define business views in those warehouses.

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

Before you read this appendix, read “Chapter 3. Setting up your datamart” on page 37 and “Chapter 4. Writing data to a warehouse database” on page 47 to learn about the basic components of a Visual Warehouse datamart. 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*.

### Defining information resources for use with DataJoiner

Define a separate information resource definition in Visual Warehouse for each DataJoiner source database. Before you define the information resources, you must map each source database to a DataJoiner database through 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 Visual Warehouse 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 Visual Warehouse.

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 DataJoiner *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.

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

## Defining warehouses for use with DataJoiner

Create a separate logical warehouse definition in Visual Warehouse 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 Visual Warehouse generates the SQL to extract data from the source database and write data to the target database, Visual Warehouse 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 business views with sources from more than one database by taking advantage of the DataJoiner heterogeneous join optimization.

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

The BVBESTATUS table contains timestamps for the business view 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, 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 Visual Warehouse requires that the name of the table be BVBESTATUS. You cannot have one DataJoiner nickname that represents 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,
              BVEDITION 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*.

## Creating target tables with DataJoiner

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

To create the target table:

1. Create a business view with a target table in a DataJoiner database.
2. On the Information page of the Business View notebook, type the server mapping name in the **Tablespace Name** field.

   In the example in “Defining information resources for use with DataJoiner” on page 31, the server mapping name is Oracle 1.
3. Determine what you need to do because the default table name qualifier for Visual Warehouse is IWH:
   - 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 the next step.

     In the example in “Defining information resources for use with DataJoiner” on page 31, the Oracle user ID must have the privilege to create a table with a qualifier that is different from the user ID.
   - Otherwise, do one of the following options:
     - Request a user ID called IWH.
     - Change the table name qualifier to match your user ID.

       In the example in “Defining information resources for use with DataJoiner” on page 31, the table name qualifier should change to `"iwhserve"`.
     - 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.
4. Promote the business view to test status.
5. Run the business view to verify that the correct data is written to the target table.
6. Promote the business view to production status.

If you have a version of DataJoiner prior 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.

## Moving the target table

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

1. Create a business view with a target table in a DataJoiner database.
2. Promote the business view to test status.
3. Run the business view 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 could 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 business view in Visual Warehouse.
6. Run the business view again to test that the data moves through DataJoiner to the target correctly.
7. Promote the business view to production status.

## Creating or updating a table in a remote database

You can use Visual Warehouse 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 Visual Warehouse warehouse by using the **Import Table** button in the Warehouse notebook.
4. Create a business view in that warehouse, selecting the imported table as the target table for the business view. In the Business View notebook, the Column Mapping Grid displays the imported table definition. On the Information page, the **Visual Warehouse Created Table** check box will not be selected. Verify that it is still not selected at the time you promote the business view.
5. Promote the business views to test status.
6. Run the business views to test access to the target tables.
7. Promote the business views to production status.

# Chapter 3. Setting up your datamart

Your next step is to specify the source and target databases to use for your datamart. You also need to specify the agent sites that Visual Warehouse will use when it accesses the source and target databases. Finally, you need to specify which users can perform administrative tasks for the datamart.

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

- “Starting Visual Warehouse” on page 37.
- “Defining security” on page 40.
- “Defining agent sites” on page 42.
- “Defining information resources” on page 42.
- “Defining a warehouse” on page 44

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

## Starting Visual Warehouse

The Visual Warehouse components start automatically when you start Windows NT. This section gives you information about working with the components that run as Windows NT Services and shows you how to open the Visual Warehouse desktop.

### Starting the server and the logger

The Visual Warehouse server and the Visual Warehouse logger run as Windows NT Services. To start them, you must restart the system after you initialize the Visual Warehouse control database. Then Visual Warehouse will automatically start every time you start Windows NT, unless you change Visual Warehouse to a manual service.

To start the Visual 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 **Visual Warehouse Server**. Click **Start** and click **OK.**
- Enter the following command from a DOS command prompt:

  ```
  NET START VWKERNEL
  ```

The Visual Warehouse logger and server start.

## Starting an agent

The server starts local agents, which are agents that are installed on the same machine as the Visual Warehouse server, automatically.

If you installed an AIX or Sun Solaris agent, the agent daemon starts automatically.

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

For information about starting the AS/400 agent, see *Installing and Using the Visual Warehouse AS/400 Agent*.

For more information about installing agents, see *Planning and Installing Visual Warehouse and DataGuide*.

### Manually starting the Windows NT agent

To manually start the Windows NT agent, start the agent daemon just as you would start any Windows NT service:

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

### Manually starting the OS/2 agent

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

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

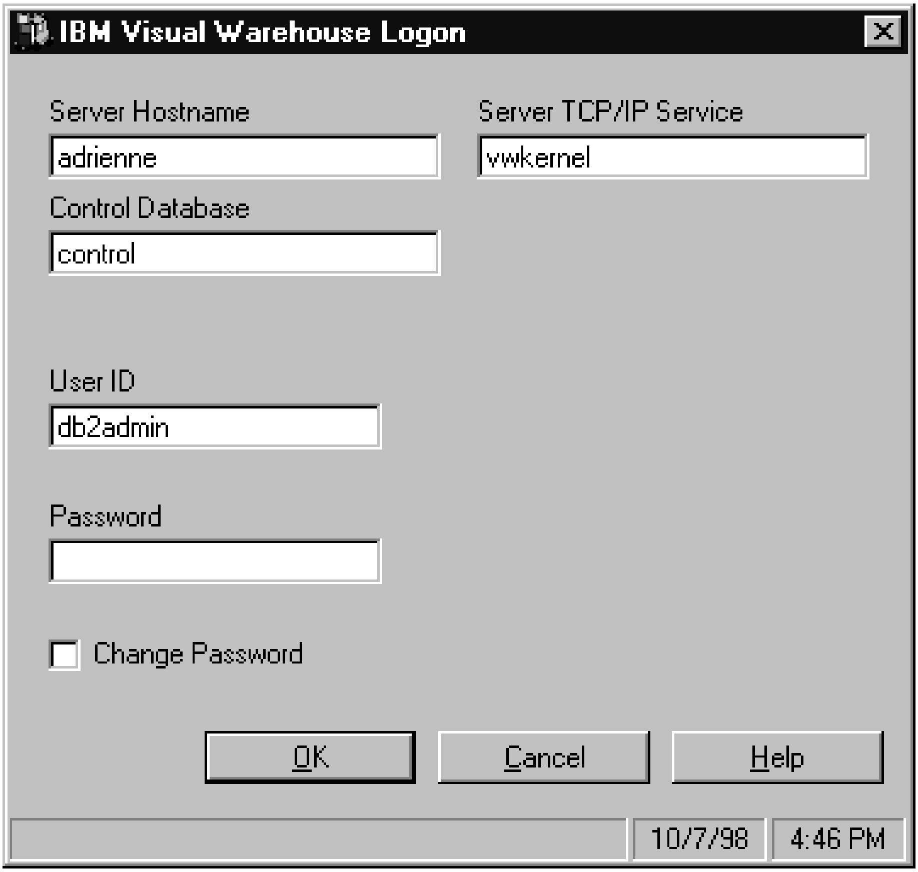
where `d:\directory` is the drive and directory in which you installed the agent and daemon.

## Starting the Visual Warehouse desktop

To start Visual Warehouse:

1. Click **Start**.
2. Click **Programs**.
3. Click **Visual Warehouse**.
4. Click **Visual Warehouse**.
5. Click **Visual Warehouse Desktop**.

A logon window opens.



Server Hostname: adrienne

Server TCP/IP Service: vwkernel

Control Database: control

User ID: db2admin

Password:

□ Change Password

10/7/98 4:46 PM

*Figure 2. Logon window for the Visual Warehouse desktop*

Type the necessary information in the window. Visual Warehouse user IDs and passwords are case sensitive. Click **Help** for assistance.

### Setting up your datamart

When you log on to the Visual Warehouse desktop for the first time, you must use the same user ID that you used to install Visual Warehouse. After you are logged on to Visual Warehouse, you can create security groups and add new users to those groups.

When you click **OK,** Visual Warehouse opens. You can begin defining information resources and business views.

## Defining security

All Visual Warehouse users must have a unique user ID and password to access the Visual Warehouse administrative client. In addition, Visual Warehouse lets you define security groups. A *security group* is a group of users with specified privileges for Visual Warehouse objects.

For each security group, you can assign one or all of the following privileges:

**Operations**
Groups with operations privileges can perform any function in the **Operations** menu. For example, they can view the log or the Operations Work in Progress window. They can also change and run any business view in the system.

**Administration**
Groups with administration privileges can add and delete users from the system and assign users to security groups.

**Resource definition**
Groups with resource definition privileges can create information resources and target warehouses, and make them available to others. Any user who needs to import information resource definitions must have resource definition privileges.

**Visual Warehouse program definition**
Groups with Visual Warehouse program definition privileges can define Visual Warehouse programs to Visual Warehouse.

**Business view definition**
Groups with business view definition privileges can create business views.

**Business view maintenance**
Groups with business view maintenance privileges can change business views in production status that have other business views dependent on them.

In all cases, privileges are system-wide. For example, if you have operations privileges, you can run any business view that is in the appropriate business view status (production or test). If you have resource definition privileges, you can open any information resource in the system.

You can use security groups to isolate access to sensitive information. For example, your warehouse might contain financial or human resource information. You can limit access to this information by including its information resources in a security group from which only users of this warehouse are authorized access. The available access authority levels are:

**Update**

To restrict access to sensitive information, Visual Warehouse restricts the authority to update a business view definition to only one security group. Visual Warehouse selects a group that includes:

- The user who created the business view
- The source for the business view
- The target warehouse for the business view

If more than one group is authorized to the user, information resource, and target warehouse, Visual Warehouse selects the first group in alphabetical order.

For information about transferring update authority from one group to another, see the online help.

**Read** Users other than the creator of a business view might need to view the business view definition. For example, users can create a business view by copying and changing another user's business view. You can grant those users read authority, which enables them to open business view definitions and look at them, but not to update them. Visual Warehouse assigns read authority to security groups that are authorized to the warehouse that contains the business view.

To work with security groups, log on to the Visual Warehouse desktop and click **Security –> Groups**.

For examples of how to use security groups and authorizations, see the online help.

## Defining agent sites

After you establish connectivity to your sources and targets, you define the sites on which you installed agents. Agent site definitions specify which agent sites Visual Warehouse uses for which sources and targets.

To define an agent site:

1. Log on to the Visual Warehouse desktop. Type your Visual Warehouse user ID and password, which are case sensitive, in the appropriate fields.
2. Click **Definitions —> Agent Definition**.

   The Agent Sites window opens.
3. Click **File —> New**.

   The Agent Sites notebook opens.
4. Type the following information in the appropriate fields:
   - The TCP/IP host name of the site
   - The type of operating system
   - A user ID and password
   - The information resources and warehouses that use that agent site

If you are installing a Visual Warehouse program on the agent site, you must define the program to Visual Warehouse before you can use it. For more information, see “Defining a Visual Warehouse program” on page 71.

## Defining information resources

After you define agent sites, you can begin using Visual Warehouse to define the objects that you want to create. Define all the data sources that you think that you might use to populate your datamart as information resources. You can always remove an information resource later, but it is best at this point to define everything you might possibly use.

Visual Warehouse supports information resource 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, Visual Warehouse stores the lowercase letter as an uppercase letter.

Visual Warehouse does not support information resource tables that use delimited identifiers. A delimited identifier:

- Is enclosed within double quotes
- 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 data sources into Visual Warehouse. For more information, see the online help.

To create an information resource, you must use a Visual Warehouse user ID with resource definition privilege. For information about defining users and security groups, see the online help.

To define a data source as an information resource:

1. Log on to the Visual Warehouse desktop. Type your Visual Warehouse user ID and password, which are case sensitive, in the appropriate fields.
2. Click the **Sources** tab and select **New**.

   The Resource Type window opens.
3. Select the type of database that you want to access.

   The Information Resource notebook opens.
4. Type the following information in the appropriate fields:
   - The source database name
   - The user ID and password used to access the database
   - The agent site name

For guided help on completing the Information Resource notebook, see “Add an information resource to Visual Warehouse” in the online help.

To help you determine which tables in the data source that you want to use, you can view the data in the source tables. You view the data from one table at a time. Visual Warehouse displays all the column names of the table, regardless of whether data is present in the column. It displays up to a maximum of 200 rows of data.

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

- To view source data before you import it into Visual Warehouse:
  1. Select a table from the **Import Table** list in the Information Resource notebook.

     When you view the data from the **Import Table** list, Visual Warehouse displays the data in the format of the source database, regardless of whether you imported a subset of the table.
  2. Click **Sample Data**.

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

  You cannot view data in Flat File information resources before you define the file to Visual Warehouse.
- To view source data after you import it into Visual Warehouse:
  1. Click the **Tables** or **Files** tab of the Information Resource notebook.
  2. Select a table or file.
  3. Click **Sample Data**.

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

  If you selected only a subset of the available columns when you imported the table, Visual Warehouse displays only the subset that you imported. For example, if the source table contains five columns, and you imported only three columns, Visual Warehouse displays only the three columns.

## Defining a warehouse

After you define the sources for your warehouse as information resources, you define the warehouse that will contain the data. Visual Warehouse supports the DB2 UDB and DB2 for OS/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 target warehouse” on page 30 and “Setting up a DB2 for OS/390 target warehouse” on page 28.

To define a warehouse, you must use a Visual Warehouse user ID with resource definition privilege. For information about defining users and security groups, see the online help. To define a warehouse:

1. Log on to the Visual Warehouse desktop. Type your Visual Warehouse user ID and password, which are case sensitive, in the appropriate fields.
2. Click the **Warehouses** tab and select **New**.

   The Warehouse notebook opens.
3. Type the following information in the appropriate fields:
   - The target database name

- The user ID and password used to access the database
- The agent site name

For guided help on completing the Warehouse notebook, see “Define a new warehouse” in the online help.

You can also group business views logically, by subject area. For more information, see “Using a Visual Warehouse program in a business view” on page 75.

# Chapter 4. Writing data to a warehouse database

After you define a warehouse, you need to populate the warehouse with useful information. To do this, you need to understand what the end users need, what source data is available, and how Visual Warehouse can transform the source data into information.

You define these transformations by defining business views. Each business view defines a transformation of data from a source format to a target format, by including the following specifications:

- One or more source tables from which Visual Warehouse is to extract data.

  You must define these source tables as part of an information resource before you can use the source tables in a business view.
- A target table to which Visual Warehouse is to write data.

  The target table is a table that Visual Warehouse will create in a warehouse database, according to your specifications in the business view.
- An SQL statement that specifies the data to extract and how to transform the data to its target format.

  For example, the SQL statement could select data from multiple source tables, join the tables, and write the joined data to a target table.

After you define the business view, you can run it to perform the transformation you described.

Most of the time, you can use Visual Warehouse's SQL processing to extract and transform data for a business view. But in some situations, you might want another program to manipulate your data. For example, you might want to use DB2's bulk load and unload utilities to transfer data to your warehouse. Or you might want to use a data cleansing tool to clean your data.

You can also define an external program to Visual Warehouse as a Visual Warehouse program, and use Visual Warehouse to schedule and monitor the program's activities.

For more information about Visual Warehouse's support for Visual Warehouse programs, see "Chapter 5. Extending Visual Warehouse with Visual Warehouse programs" on page 71.

The following sections describe Visual Warehouse's SQL support.

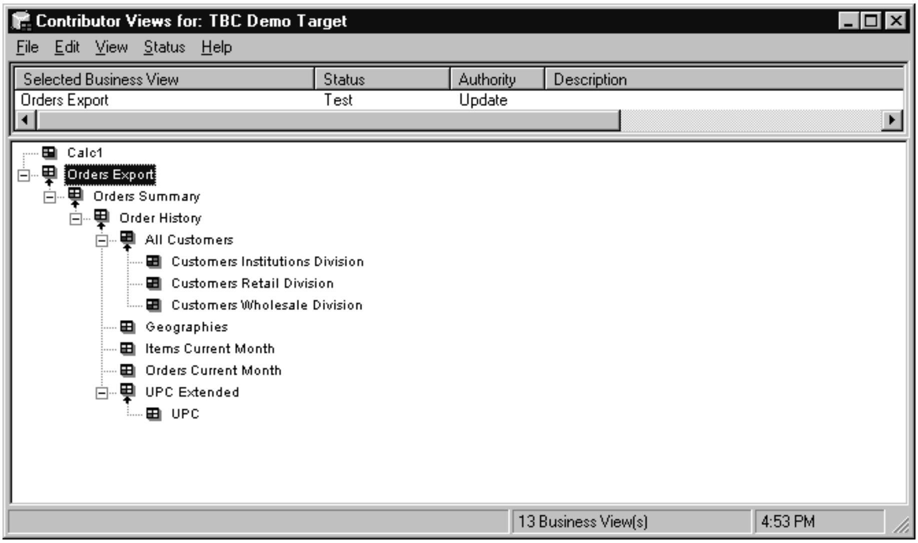
## TBC Demo example

Some of the following sections refer to a sample warehouse, called TBC Demo, that is included with Visual Warehouse. The TBC Demo uses sales data from a fictional beverage company. The operational data for the company includes the following data:

- A list of products
- A list of geographies
- A list of customers
- The monthly sales for the company

In the warehouse, the sales data is summarized by geography, by product, and by package.

Conceptually, the TBC Demo warehouse looks like Figure 3.



| Selected Business View | Status | Authority | Description |
|---|---|---|---|
| Orders Export | Test | Update | |

*Figure 3. Conceptual view of the TBC Demo sample data*

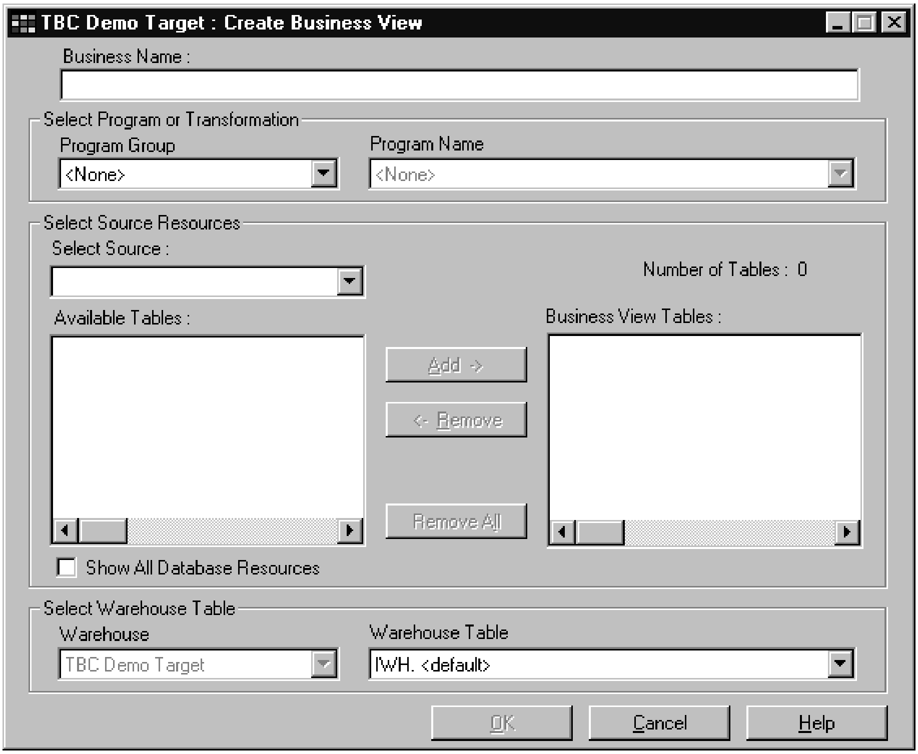
## Selecting source tables

You select source tables or files for the business view when you initially define the business view. You can select tables or files from any information resource that you have defined for your datamart.

To define a new business view:

1. Open the Visual Warehouse desktop.

2. Click the **Warehouses** tab.
3. Double-click on the warehouse you want to use. The Business View window opens.
4. Click **File —>New**. The Create Business View window opens, as shown in Figure 4.



TBC Demo Target : Create Business View

Business Name :

Select Program or Transformation

Program Group: <None>

Program Name: <None>

Select Source Resources

Select Source :

Number of Tables : 0

Available Tables :

Business View Tables :

☐ Show All Database Resources

Select Warehouse Table

Warehouse: TBC Demo Target

Warehouse Table: IWH. <default>

*Figure 4. The Create Business View window*

5. Type a name for the business view that describes its contents from a business perspective.

To select source tables or files:

1. Select an information resource from the **Select Source** list.
2. Select a table or file from the **Available Tables** list.
3. Click **Add**.

   The name of the table or file moves to the **Business View Tables** list.
4. Continue to select tables until you have selected all the tables you need. (You can select only one file.)

You can also select source tables on the Sources page of the Business View notebook. For more information, see the online help.

## Defining a target table

You also define the target table for the business view when you initially define the business view. You can select an existing table or create a table in any warehouse you have defined for your datamart.

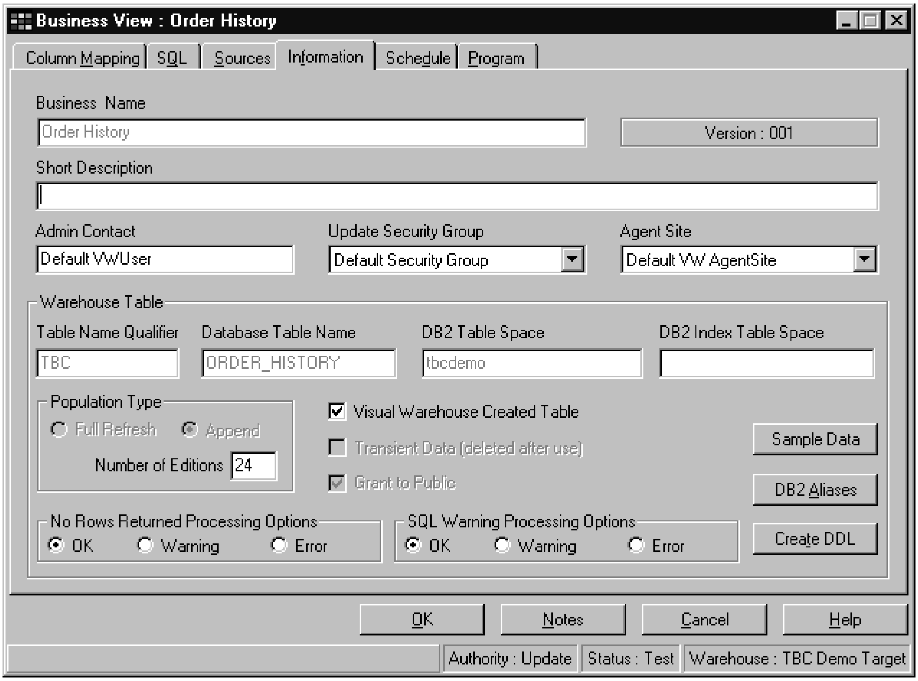
To define a target table:

1. Select a warehouse from the **Warehouse** list.
2. If you want Visual Warehouse to create a target table for the business view, leave *schema***<default>** selected in the **Warehouse Table** list, where *schema* is the schema or qualifier for the table. The default schema is IWH.

   If you want to use an existing table in the warehouse database, select the table from the list.
3. Click **OK**.

   The Business View notebook opens.

You can view information about the target table for the business view on the Information page of the Business View notebook, shown in Figure 5 on page 51.



Business Name: Order History

Version : 001

Short Description:

Admin Contact: Default VWUser

Update Security Group: Default Security Group

Agent Site: Default VW AgentSite

Warehouse Table

| Table Name Qualifier | Database Table Name | DB2 Table Space | DB2 Index Table Space |
|---|---|---|---|
| TBC | ORDER_HISTORY | tbcdemo | |

Population Type: ○ Full Refresh ● Append

Number of Editions: 24

☑ Visual Warehouse Created Table

☐ Transient Data (deleted after use)

☑ Grant to Public

No Rows Returned Processing Options: ● OK ○ Warning ○ Error

SQL Warning Processing Options: ● OK ○ Warning ○ Error

Authority : Update | Status : Test | Warehouse : TBC Demo Target

*Figure 5. A target table on the Information page of the Business View notebook.* In this example, the target table is TBC.ORDER_HISTORY. Visual Warehouse will define the table in the tbcdemo table space.

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

Visual Warehouse 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, Visual Warehouse stores the lowercase letter as an uppercase letter.

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

- Is enclosed within double quotes
- Can include uppercase and lowercase letters, numbers, underscores, and spaces

- Can contain a double quotation mark, represented by two consecutive quotation marks.

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

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

## Specifying how to save data

In Visual Warehouse you can define *persistent* or *transient* business views. Visual Warehouse permanently stores the contents of a persistent business view in a warehouse. The contents of a transient business view exist only for the duration of a specific event, such as the population of another business view. Visual Warehouse empties the contents of a transient business view when the job that requires it completes.

You can replace the output data each time a persistent business view runs, or you can append to the existing data.

If one of the objectives of your datamart is history tracking, you will find business view editions useful. Each time a business view runs, you can save the results as a new edition. Visual Warehouse lets you have unlimited editions of a given business view.

When you create a business view, you can specify how many editions of that business view you want to keep. To specify the number of editions, click the **Information** tab in the Business View notebook. If you specify one or more editions, Visual Warehouse adds a column to the business view's target table that identifies each edition of the business view. The default name of the column is VWEDITION. For more information on using the Information page, click **Help** on the Information page.

The example shown in Figure 5 on page 51 specifies that Visual Warehouse save (append) 24 editions of the data. On the 25th time the business view runs, Visual Warehouse will replace the data in the first edition with newly extracted data.

## Creating the Visual Warehouse status table in the warehouse database

End 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 the end 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. Copy the BVSTATUS.DDL file from the Vwsample\Vwp directory of the Visual Warehouse CD-ROM.
2. Change the following CREATE TABLE statement in the BVSTATUS.DDL file:

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

   - If your target warehouse is in DB2 for Windows NT, DB2 for OS/2, DB2 for OS/400, DB2 for AIX, or DB2 on a UNIX-based platform, you need not modify the statement.
   - If your target warehouse is in DB2 for MVS, 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
     ```

     where:

     – *database-name* is the name of the database in which you want to create the BVBESTATUS table

     – *tablespace-name* is 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 (VWEDITION)
     USING HASHING
     ```

     where:

     – *tablespace-name* is the name of the table space in which you want to create the BVBESTATUS table

     – *partitioning key* is the name of the column used to determine in which partition a particular row of data is stored
3. Connect to the target warehouse database.
4. From the directory that contains the BVSTATUS.DDL file, issue the following command:

   ```
   DB2 -TVF BVSTATUS.DDL
   ```

## Mapping source columns to target columns

When you use Visual Warehouse, 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 business view.

For example, you want to create some business views 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 business view for each of the four sites. Figure 6 shows the initial mapping between a source table and a warehouse table.

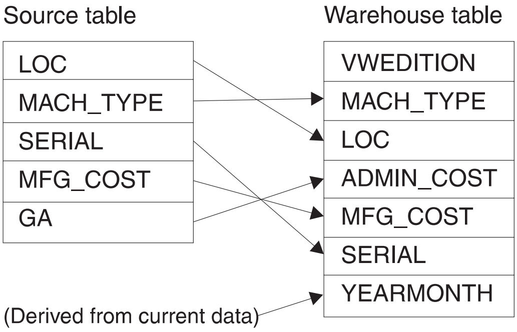
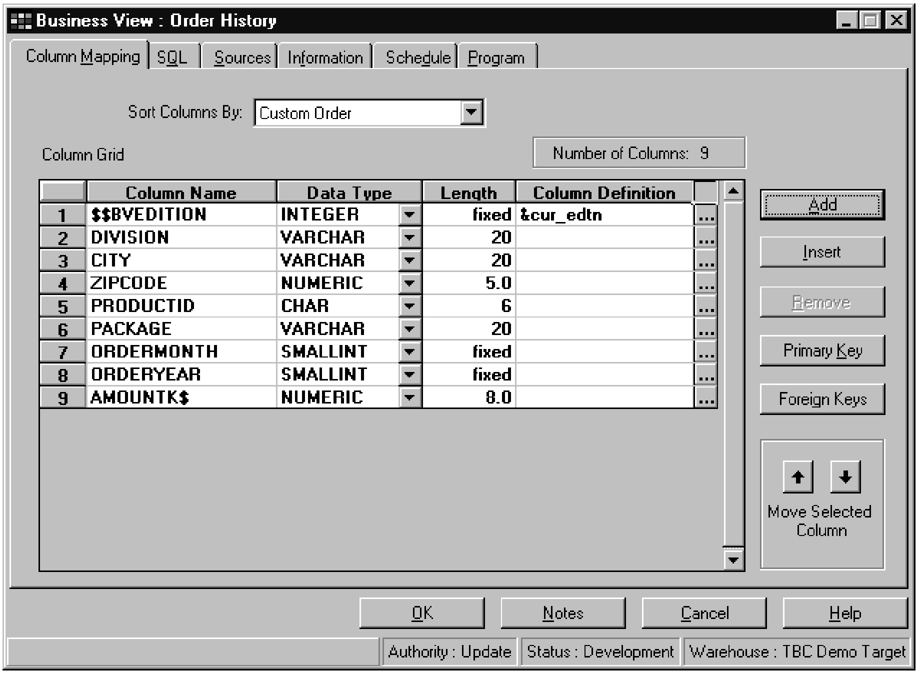


*Figure 6. A mapping of source data to a business view*

The first page of the Business View notebook is the Column Mapping page. You use this page to specify the names and data types of the columns in your business view. From this page, you also define how Visual Warehouse computes the columns of the target table using columns from the source data, constants, and tokens.

Visual Warehouse uses your selections to generate an SQL statement that selects data from the source tables and inserts the data into the target tables. You can use special functions such as the **Where** and **Join** push buttons to change the SQL statement.

For example, you want to create a business view that extracts data from three tables about customers, their orders, and the items they ordered. You select common columns to include in the target table for the business view. (You will join the tables later.) The Column Grid displays the definitions of the columns you select. Figure 7 on page 55 shows an example of column definitions for a business view.



Sort Columns By: Custom Order

Number of Columns: 9

| | Column Name | Data Type | Length | Column Definition |
|---|---|---|---|---|
| 1 | $$BVEDITION | INTEGER | fixed | &cur_edtn |
| 2 | DIVISION | VARCHAR | 20 | |
| 3 | CITY | VARCHAR | 20 | |
| 4 | ZIPCODE | NUMERIC | 5.0 | |
| 5 | PRODUCTID | CHAR | 6 | |
| 6 | PACKAGE | VARCHAR | 20 | |
| 7 | ORDERMONTH | SMALLINT | fixed | |
| 8 | ORDERYEAR | SMALLINT | fixed | |
| 9 | AMOUNTK$ | NUMERIC | 8.0 | |

Authority : Update | Status : Development | Warehouse : TBC Demo Target

*Figure 7. Columns in the Column Mapping page*

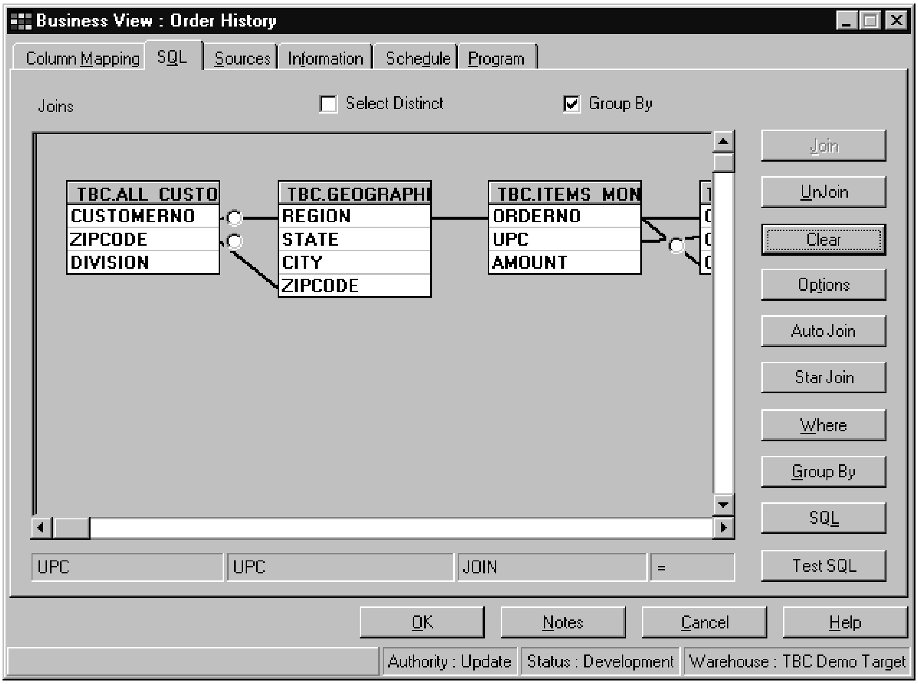
You can also create a primary key and foreign keys for the target table. A *primary key* is one or more columns that you select to identify a row in the table. You cannot include columns that allow nulls in the primary key. A *foreign key* is a column in a table whose allowable values must exist as the primary key for another table.

Visual Warehouse creates the primary key in the target table when it creates the target table.

You generally specify primary and foreign keys for use in a star join, which is a specialized type of join. For information about star joins, see “Generating star joins” on page 58.

## Joining source columns

If you select multiple source tables, you also need to specify how the rows of these columns will be associated in the join of the tables. The SQL page of the Business View notebook, shown in Figure 8 on page 56, contains graphical controls that you can use to define the join.



☐ Select Distinct

☑ Group By

UPC | UPC | JOIN | =

Authority : Update | Status : Development | Warehouse : TBC Demo Target

Figure 8. Graphical join on SQL page

For example, you want to join columns of the five source tables for the Order History business view. You use the controls to join the columns. Figure 9 shows the SQL statement that Visual Warehouse generates.

```
SELECT &cur_edtn, TBC.ALL_CUSTOMERS.DIVISION, TBC.GEOGRAPHIES.City,
TBC.GEOGRAPHIES.Zipcode, TBC.TPC.EXTENDED.PRODUCTID,
TBC.UPC_EXTENDED.PACKAGE,
MONTH(DATE(TBC.ORDERS_MONTH.OrderDate)),
YEAR(DATE(TBC.ORDERS_MONTH.OrderDate)),
SUM(TBC.ITEMS_MONTH.Amount/1000
FROM TBC.UPC_EXTENDED,TBC.ALL_CUSTOMERS,TBC.ORDERS_MONTH,
TBC.ITEMS_MONTH,TBC.GEOGRAPHIES
WHERE TBC.ITEMS_MONTH.OrderNo=TBC.ORDERS_MONTH.OrderNo
     AND TBC.ALL_CUSTOMERS.Zipcode=TBC.GEOGRAPHIES.Zipcode
     AND TBC.ORDERS_MONTH.CustomerId=TBC.ALL_CUSTOMERS.CustomerNo
     AND TBC.UPC_EXTENDED.UPC=TBC.ITEMS_MONTH.UPC
GROUP BY TBC.ALL_CUSTOMERS.DIVISION, TBC.GEOGRAPHIES.City,
     TBC.GEOGRAPHIES.Zipcode,
     TBC.UPC_EXTENDED.PRODUCTID, TBC.UPC_EXTENDED.PACKAGE
```

Figure 9. The SQL statement generated by a graphical join

If your database supports the INNER JOIN keyword, you can use Visual Warehouse to generate a join that uses the keyword and a standard join.

You can test the results of an SQL statement before you save it to your business view. To view a sample of the data that will result from your SQL statement, click **Test SQL**.

The Sample Data window opens and displays a sample of your data. Visual Warehouse displays all the columns of the table, and up to a maximum of 200 rows.

For more information about the SQL page, see the online help.

## 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 datamart 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 an information resource. Then select those tables as source tables for a business view. You then click the **Join** push button on the SQL page of the Business View notebook 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 specify 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

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 or not, 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 a detailed example of the types of joins, see “About joins” in the online help.

## 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, Visual Warehouse 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 a detailed comparison of the types of joins, see “About joins” in the online help. For information about specifying the different types of joins, see the help for the SQL page of the Business View notebook.

## Filtering data

In most cases, when you create a business view, you want only a subset of the source data. You might want to extract only the rows that meet certain criteria. You can use Visual Warehouse to build an SQL WHERE clause to limit the rows that you extract from the source table.

For example, in the TBC Demo warehouse, the Order History business view selects rows from the most recent edition of the source table:

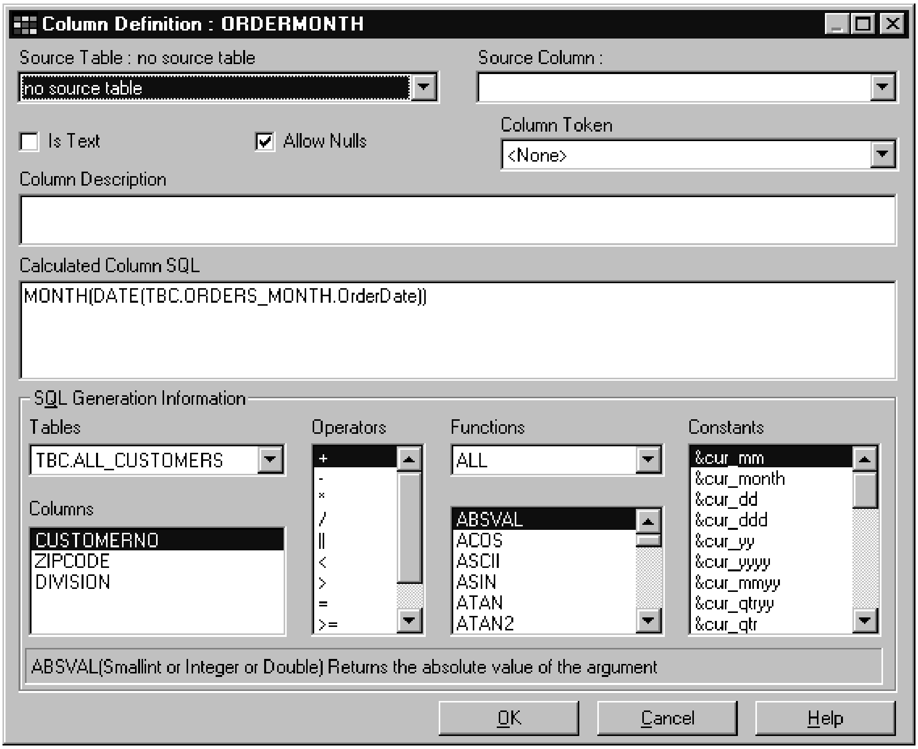
```
WHERE TBC.ORDER_HISTORY.VWEDITION = &cur_edtn.IWHDATA.TBC.ORDER_HISTORY
```

The `VWEDITION` column contains information about the business view edition. The `&cur_edtn` token represents the current business view edition. Therefore, this WHERE clause selects rows in which the business view edition equals the current edition.

To build the WHERE clause, click the **SQL** tab of the Business View notebook, click **Where**, and use the **SQL Generation** controls. You can use these controls to specify conditions on any column of the source tables. The online help guides you through this procedure.

## 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. Figure 10 on page 60 shows the SQL statement for the calculated column.



Column Definition : ORDERMONTH

Source Table : no source table

no source table

Source Column :

□ Is Text ☑ Allow Nulls

Column Token

<None>

Column Description

Calculated Column SQL

```
MONTH(DATE(TBC.ORDERS_MONTH.OrderDate))
```

ABSVAL(Smallint or Integer or Double) Returns the absolute value of the argument

*Figure 10. Calculating the value of a column*

Visual Warehouse does not add the calculated columns to the GROUP BY clause. You click the **SQL** push button and add the two columns to the GROUP BY clause:

```
GROUP BY TBC.ALL_CUSTOMERS.DIVISION, TBC.GEOGRAPHIES.City,
     TBC.GEOGRAPHIES.Zipcode, TBC.UPC_EXTENDED.PRODUCTID,
     TBC.UPC_EXTENDED.PACKAGE

     ,MONTH(DATE(TBC.ORDERS_MONTH.OrderDate))
     ,YEAR(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 datamart. 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.

Visual Warehouse lets you easily and accurately define business views 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 business views that summarize the source data.

Summary business views 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 business views that use summary techniques to summarize other business views. Summarizing reduces the size of the target warehouse that you create.

To create a business view with this type of summary, click the **SUM** function in the **Functions** field of the Column Definition window.

In the TBC Demo example, a business view summarizes all the items that are sold in a month and expresses the amount in thousands of dollars:

```
SUM(TBC.ITEMS_MONTH.Amount)/1000
```

## Cleansing data

When you build business views with data from a variety of sources, you will find that the data is sometimes inconsistent or incompatible between sources. You need to perform some level of data cleansing to ensure that the data is consistent with itself and is as accurate as possible.

Table 6 lists several methods of cleansing data.

*Table 6. Methods of cleansing data*

| Method | Description | For more information, see: |
|---|---|---|
| WHERE clause | You can build an SQL WHERE clause to limit the rows you extract from the source table. | “Filtering data” on page 59 |
| Formulas and expressions | You can use formulas and expressions to eliminate the data that you do not need and to create the data that you do need. Use the Column Definition window to specify formulas, constants, and tokens. | “Adding calculated columns” on page 59 |
| Visual Warehouse programs | Visual Warehouse programs let you perform any functions on the source data that are not available from the previous two methods. | “Chapter 5. Extending Visual Warehouse with Visual Warehouse programs” on page 71 |

For example, you have several business views that extract revenue data from a database that has already converted the local currency into U.S. dollars.

### Writing data to a warehouse database

Your REVENUE_EUROPE business view, however, extracts data from a database that stores revenue data in local currencies. To use the European revenue data, you need to do some data cleansing.

Visual Warehouse performs data cleansing by running a Visual Warehouse program on the data that you want to clean. In this example, the REVENUE_EUROPE business view runs after the REVENUE_SINGAPORE business view completes. REVENUE_EUROPE calls a batch file named udpclns, which uses an exchange rate from a business view called Rate to convert the data in REVENUE_EUR_LC to U.S. dollars. In this way, REVENUE_EUROPE is consistent with all the other business views that contain revenue data.

You associate a Visual Warehouse program with a business view by selecting the program when you create the business view. You can also change a business view to use a Visual Warehouse program by selecting the program in the Program page of the Business View notebook.

## Running business views

Visual Warehouse lets you manage the development of your business view by classifying business views in one of three statuses: development, test, or production. The status determines whether you can make changes to the business view and whether Visual Warehouse will run the business view according to its schedule.

### Developing business views

When you first create a business view, it has development status. You can change any part of the business view definition with this status. Visual Warehouse has not created a table for it in the target warehouse. You cannot run the business view to test it, and Visual Warehouse will not run it according to its automated schedule.

### Testing a business view

You can direct Visual Warehouse to create a target table for a business view. Visual Warehouse creates the table when you promote the business view to test status. On the Information page of the Business View notebook, you specify whether Visual Warehouse creates a target table. For specific instructions on completing this task, see the online help.

You can make changes to a business view in test status that do not modify existing columns from the underlying table. For example, you can add columns to the target table, but you cannot change the data type of an existing column.

To promote or demote the status of a business view, select the business view that you want to promote or demote. Click the **Status** menu and the action that you want to perform.

For more information about status levels, see “business view status” in the online help.

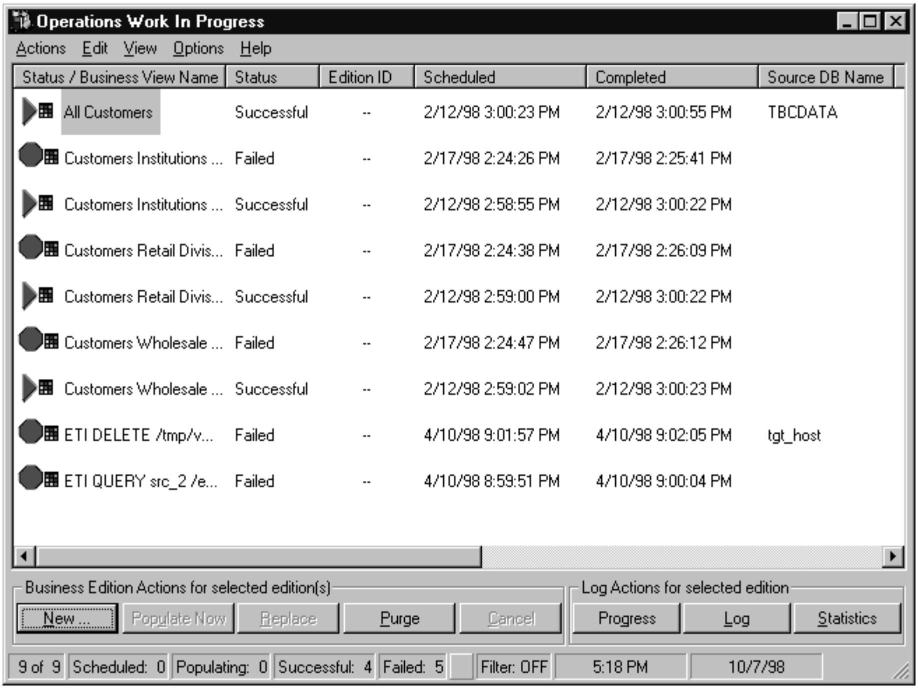
If data already exists in the target table, you can view the data before running the business view:

1. Click the **Information** tab of the Business View notebook.
2. Click **Sample Data**.

   This button is available only if you have specified a qualifier and name for the target table.

The Sample Data window opens and displays a sample of your data. Visual Warehouse displays all the columns of the table, and up to a maximum of 200 rows.

You can run and test a business view in test status by clicking **New** in the Operations Work in Progress window, shown in Figure 11.



| Status / Business View Name | Status | Edition ID | Scheduled | Completed | Source DB Name |
|---|---|---|---|---|---|
| All Customers | Successful | -- | 2/12/98 3:00:23 PM | 2/12/98 3:00:55 PM | TBCDATA |
| Customers Institutions ... | Failed | -- | 2/17/98 2:24:26 PM | 2/17/98 2:25:41 PM | |
| Customers Institutions ... | Successful | -- | 2/12/98 2:58:55 PM | 2/12/98 3:00:22 PM | |
| Customers Retail Divis... | Failed | -- | 2/17/98 2:24:38 PM | 2/17/98 2:26:09 PM | |
| Customers Retail Divis... | Successful | -- | 2/12/98 2:59:00 PM | 2/12/98 3:00:22 PM | |
| Customers Wholesale ... | Failed | -- | 2/17/98 2:24:47 PM | 2/17/98 2:26:12 PM | |
| Customers Wholesale ... | Successful | -- | 2/12/98 2:59:02 PM | 2/12/98 3:00:23 PM | |
| ETI DELETE /tmp/v... | Failed | -- | 4/10/98 9:01:57 PM | 4/10/98 9:02:05 PM | tgt_host |
| ETI QUERY src_2 /e... | Failed | -- | 4/10/98 8:59:51 PM | 4/10/98 9:00:04 PM | |

*Figure 11. Operations Work in Progress window*

## Writing data to a warehouse database

Visual Warehouse issues the SQL statements for the business view, or starts the Visual Warehouse program when you run the business view. You can track the status of the business view on the Operations Work in Progress window. However, you cannot test post-processing programs in test status. Visual Warehouse will not run the business view according to its automated schedule.

## Scheduling a production business view

If data already exists in the target table for the business view, you can view the data before running the business view. See “Testing a business view” on page 62 for more information.

To schedule a data extract, you can use Visual Warehouse’s scheduling capabilities, or you can use Visual Warehouse with another product that has its own scheduling functions.

### Using the Visual Warehouse scheduling functions

Use the Business View notebook to schedule how often a business view will run. The business view will run automatically at the scheduled time. In addition, you can start a business view from the Run Business View window at any time.

When you set up a schedule for a business view, you also specify when the business view will run in relation to other business views. For example, you can specify that another business view will run after this business view completes. If two business views do not relate to each other hierarchically, you can specify that they run at the same time.

If you plan to run several business views concurrently, you might want to tune your database. Check the DB2 database configuration parameter LOCKS TIMEOUT. The default value is -1, which means that DB2 will wait forever. If you are experiencing locking problems, use the DB2 Control Center or the DB2 Command Line Processor to set the wait time to 30 seconds.

To define a schedule for a business view, click the **Schedule** tab. Click **Help** for detailed instructions on defining a schedule.

When you promote the business view to production status, Visual Warehouse runs the object according to the automated schedule that you specified.

### Using Visual Warehouse with other products’ scheduling functions

When you define a business view, you specify how it is to be populated. By default, Visual Warehouse fills a warehouse by running the SQL statements that are specified in a business view. However, you can define a business view as *externally populated*, which means that Visual Warehouse will create the table, but another program will populate it. The other program is scheduled and runs outside of Visual Warehouse.

For example, you define your business view as externally populated. You can then use the scheduling capabilities of DPropR to populate the Visual Warehouse business view.

## Starting a business view from outside Visual Warehouse

You can start a business view independently of the Visual Warehouse user interface by using an external trigger program. An *external trigger program* is a Visual Warehouse program that calls the Visual Warehouse Populate API.

The Visual Warehouse CD-ROM includes a sample external trigger program IWHETRIG.EXE, and its source code, IWHTRIG.CPP.

### Syntax

The syntax for starting IWHETRIG.EXE is as follows:

**IWHETRIG.EXE**

```
►►──IWHETRIG.EXE──vwserver_host──vwserver_port──vwserver_userid──────────────────────►

►─vwserver_password──business_view_name──┬─────────────────┬──────────────────────────►◄
                                         └─fetch_limit─────┘
```

*vwserver_host*
: The TCP/IP host name for the Visual Warehouse server.

  Specify a fully qualified host name.

*vwserver_port*
: The TCP/IP port assigned to the Visual Warehouse server.

  This value is usually 11000.

*vwserver_userid*
: A user ID with Visual Warehouse Operations privileges.

*vwserver_password*
: The password for the user ID.

*business_view_name*
: The name of the business view to start.

  The name is case-sensitive. Enclose the name in double quotation marks (“”) if it includes blanks, such as “Corporate Profit”.

*fetch_limit*
The maximum number of rows to fetch.

This parameter is optional and valid only when the business view is in test status.

### Example

For example, if you want to start the Corporate Profit business view on the vwserver host using a user ID of vwadmin and a password of vwadmin, issue the following command:

```
iwhetrig vwserver 11000 vwadmin vwadmin "Corporate Profit"
```

### Status

When you run IWHETRIG.EXE, it sends a message to the Visual Warehouse server. If the message is sent successfully, IWHETRIG.EXE returns a zero return code.

IWHETRIG.EXE returns a nonzero return code if it could not send the message to the Visual Warehouse server. The return codes match the corresponding codes that are issued by the Visual Warehouse Operations function when there is a communications error or when authentication fails. For information about the Visual Warehouse operations codes, see *Messages and Reason Codes*.

The business view might take hours or even days to run, depending on the size of the data and the complexity of the query. Because of this, IWHETRIG.EXE does not return any information about the success, failure, or status of the business view. You can monitor the status of the business view from the Operations Work in Progress and Log Viewer windows.

## Combining business views

After you map your data sources to business views, you can begin modeling your business using business views. Visual Warehouse gives you flexibility in how you combine business views:

- You can use the target table for a business view as the source for another business view.
- You can specify the order in which business views are run.
- You can specify conditional actions that can trigger business views.

## Using a business view as a source for another business view

When you use Visual Warehouse to build a datamart, you model your business by creating a tree-like structure of business views. You create this structure by defining business views that use the target tables of other business views as source tables.

For example, you can combine manufacturing-cost data from one set of sources with sales data from another set of sources. You can then create a business view that gives you an accurate picture of the profitability of your organization. In this way, you can transform raw data into useful information.

For example, you have two business views, Sum of Cost by Machine and Sum of Revenue by Machine. Both business views contain data that is summarized from several different sources. The Sum of Cost by Machine business view contains cost information; the Sum of Revenue by Machine business view contains revenue information. Both business views are summarized according to the type of machine.

You can also create a Profit business view. To do this, extract the data in the cost and revenue business views. Then, on a row-by-row basis, transform the data into a profitability statement. For each machine that has a row in both tables, this business view calculates the profitability of that machine using a standard formula specified in the Column Definition window.

The result is a business view that shows which machines are profitable and which are not.

Figure 12 on page 68 shows the SQL statement that the business view uses.

```
SELECT &cur_edtn, FINANCE.COST_BY_MACHINE.MACHINE_TYPE,
FINANCE.COST_BY_MACHINE.TOTAL_COST
/ FINANCE.COST_BY_MACHINE.TOTAL_QUANTITY,
FINANCE.REVENUE_BY_MACHINE.REVENUE
/ FINANCE.REVENUE_BY_MACHINE.QUANTITY
(((100.0* FINANCE.REVENUE_BY_MACHINE.REVENUE)
/ FINANCE.REVENUE_BY_MACHINE.QUANTITY)
/ FINANCE.COST_BY_MACHINE.TOTAL_COST
/ FINANCE.COST_BY_MACHINE.TOTAL_QUANTITY) - 100.0),
FINANCE.COST_BY_MACHINE.YEARMONTH
FROM FINANCE.COST_BY_MACHINE,FINANCE.REVENUE_BY_MACHINE
WHERE FINANCE.COST_BY_MACHINE.MACHINE_TYPE
= FINANCE.REVENUE_BY_MACHINE.MACHINE_TYPE
AND FINANCE.COST_BY_MACHINE.YEARMONTH
= FINANCE.REVENUE_BY_MACHINE.YEARMONTH
AND FINANCE.COST_BY_MACHINE.YEARMONTH = &cur_yyyymm-1
```

*Figure 12. SQL statement that calculates the profitability of a machine*

## Cascading business views

In addition to an hierarchical structure, you might also have business views that are not hierarchically connected. These business views can relate to each other through some business process or rule. When you create a set of business views that are related in this way, you are creating *cascading* business views.

For example, you might create multiple cascading business views that extract revenue data. The Revenue-Atlanta business view runs first. When it completes processing, Visual Warehouse starts the Revenue-Seattle business view. When that business view completes processing, Visual Warehouse starts the Revenue-Mexico business view, and so on. When processing finishes, Visual Warehouse has a completed business view for each of the seven revenue centers. Hierarchically, all the business views that are related to revenue are on the same level. But by cascading one after the other, the warehouse ensures that all the revenue data is in place before proceeding with the next set of calculations.

You can specify that a business view is to start when another business view is successful or when another business view fails.

To specify the order in which business views are run, click the **Schedule** tab of the Business View notebook. For step-by-step instructions on how to complete the page, click **Help**.

## Conditional cascades of business views

You can specify that a business view will start another business view to run only under certain conditions. This type of relationship between business views is called a *conditional cascade*. You use a Visual Warehouse program to determine whether the conditions for the cascade exist.

For example, you might write a conditional cascade program that keeps track of whether the company should issue awards for exceptional sales. Every time the Income business view completes, the program looks to see if the net income for any region exceeds an overachievement level. If a region exceeds its target sales goals, then another business view, Corporate Awards, runs to calculate the award amounts for each receiving region.

Both the business view and its associated conditional cascade program must run successfully for the cascaded business view to start.

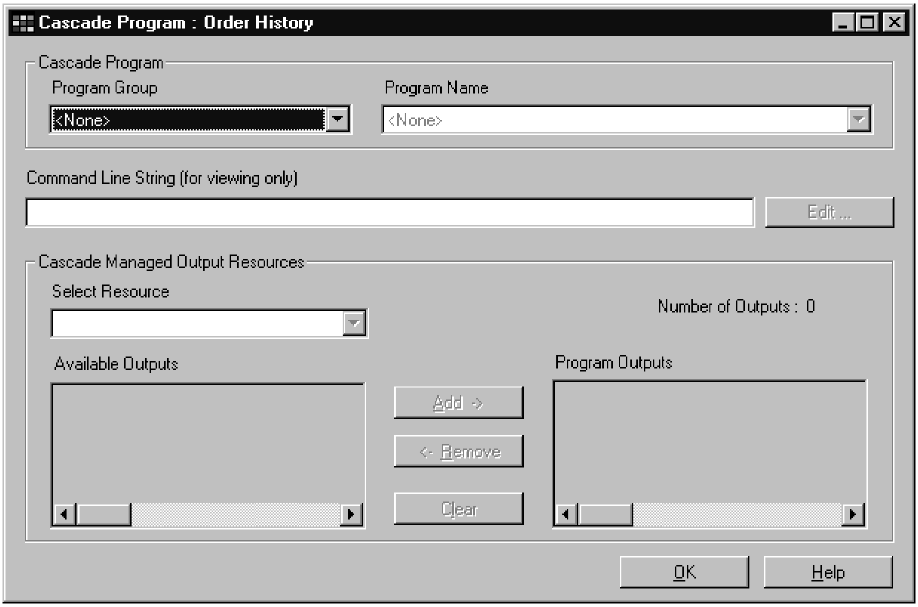
You can also use Visual Warehouse programs for post-processing. For example, if you create a business view using a Visual Warehouse program that loads data, you can schedule another Visual Warehouse program, such as the DB2 UDB Runstats program, to run the RUNSTATS utility on the table after it is loaded.

Before you can use a program for a conditional cascade or for post-processing, you must define it as a Visual Warehouse program. For more information, see "Defining a Visual Warehouse program" on page 71.

To select a Visual Warehouse program:

1. Click the **Schedule** tab of the Business View notebook.
2. Click **Program**.

   The Cascade Program window, shown in Figure 13 on page 70, opens.



Cascade Program : Order History

Program Group <None>

Program Name <None>

Number of Outputs : 0

Figure 13. Cascade Program window

For detailed instructions on specifying a conditional cascade, click **Help**.

## Accessing the warehouse from a client workstation

After you have all the components installed, and a warehouse database is set up and populated, you can use the data in the warehouse from a client workstation. Client workstations need connectivity to the warehouse database. They do not need connectivity to the Visual Warehouse server.

# Chapter 5. Extending Visual Warehouse with Visual Warehouse programs

You can use Visual Warehouse programs to include best-of-breed data warehouse software to your datamart, while providing a single point-of-control to administer the datamart.

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 Visual Warehouse program, and run it as a post-processing program for a business view. (See "Conditional cascades of business views" on page 69.)

The following sections explain how to use Visual Warehouse programs:

- "Defining a Visual Warehouse program"
- "Using a Visual Warehouse program in a business view" on page 75
- "Writing your own program for use with Visual Warehouse" on page 83

## Defining a Visual Warehouse program

You define a program to Visual Warehouse so that Visual Warehouse can start it at a scheduled time and pass parameters to it. Visual Warehouse starts the program when a business view that uses the program runs.

If you are using one of the supplied Visual Warehouse programs, you need not define the supplied program to Visual Warehouse. Visual Warehouse defines the supplied programs during initialization. For information about using the supplied Visual Warehouse programs, see "Chapter 6. Using the supplied Visual Warehouse programs" on page 91.

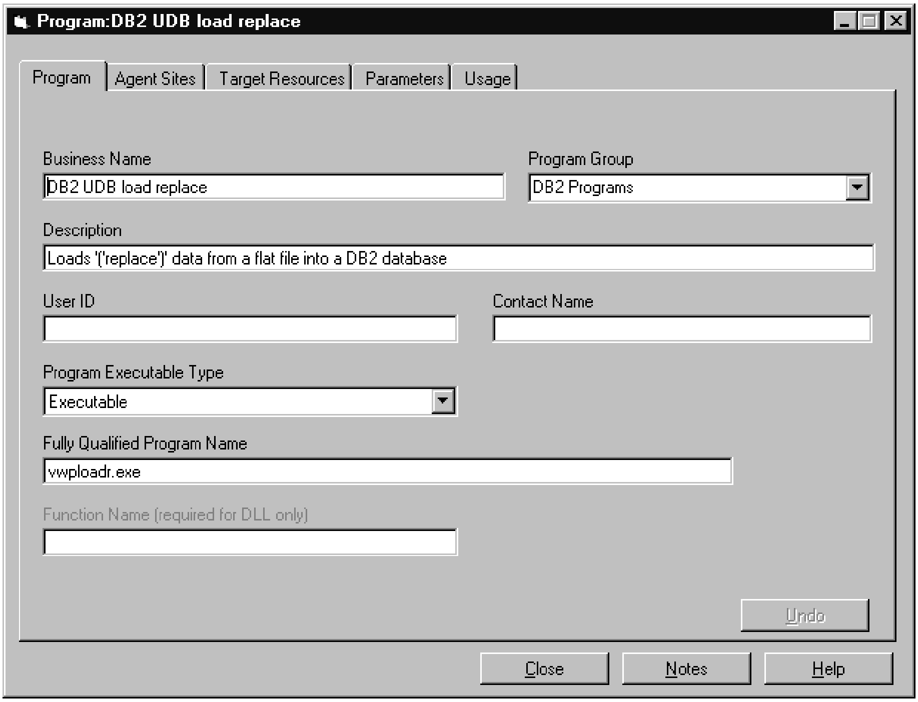
### Specifying the program to use

To define a new Visual Warehouse program:

1. Click **Definitions—>Visual Warehouse Programs** from the Visual Warehouse desktop.

   The Programs window opens.
2. Click **New**.

   The Program notebook opens.



Business Name: DB2 UDB load replace

Program Group: DB2 Programs

Description: Loads '('replace')' data from a flat file into a DB2 database

User ID:

Contact Name:

Program Executable Type: Executable

Fully Qualified Program Name: vwploadr.exe

Function Name (required for DLL only):

*Figure 14. Defining a Visual Warehouse program*

On the Program page of the Program notebook, shown in Figure 14, you need to provide the following information to Visual Warehouse:

- The name of the program.
- The type of program: executable, batch command program, or dynamic load library.

  If you select **Dynamic Load Library**, you must also provide the function name.

## 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. Figure 15 on page 73 shows the Agent Sites page.

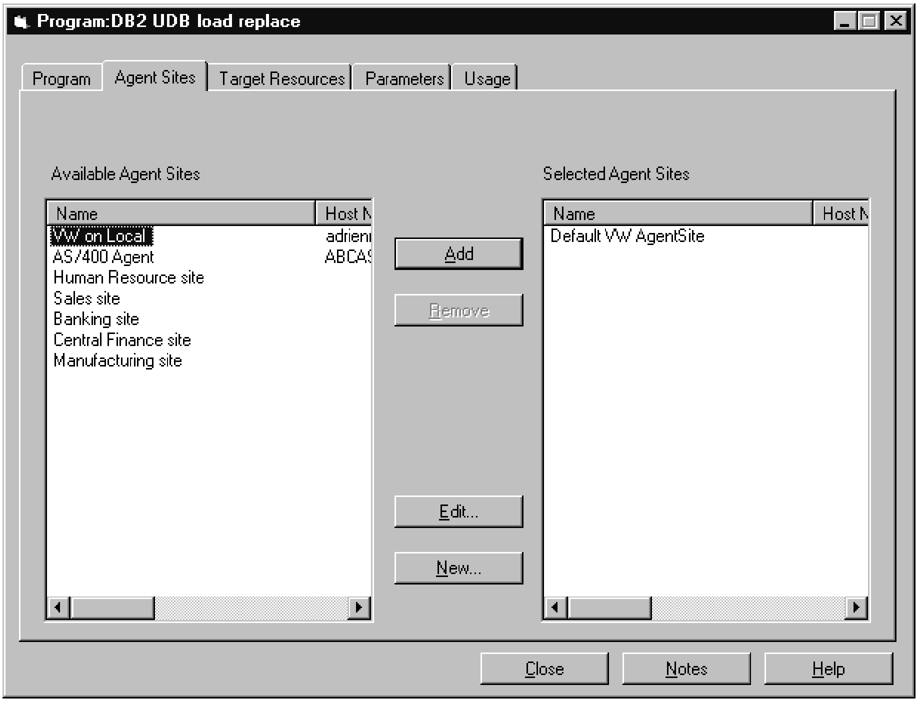


*Figure 15. Specifying the agent sites on which the Visual Warehouse program is installed*

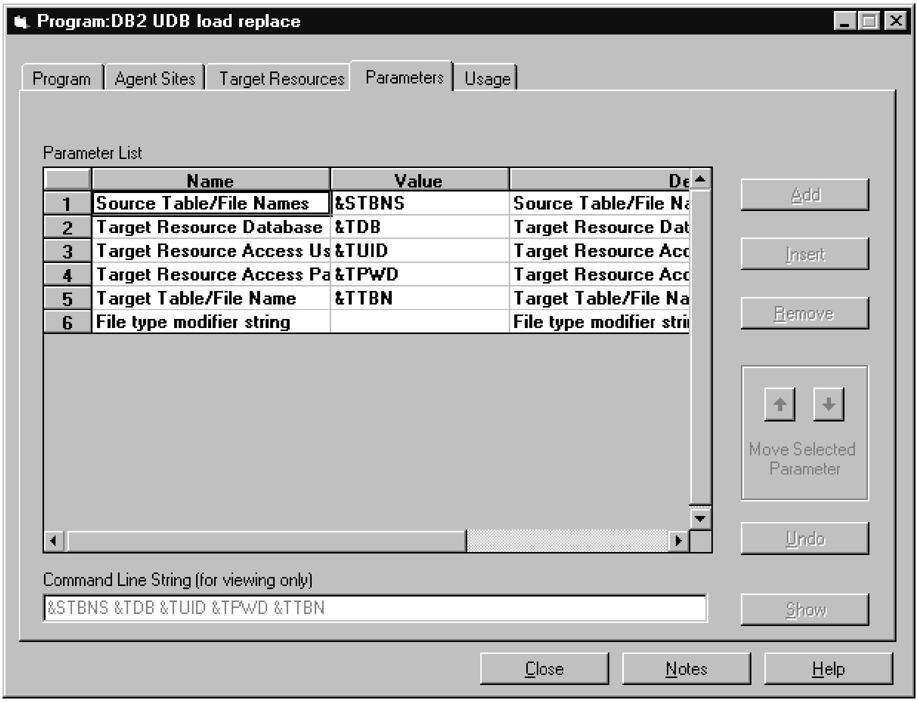
If you specified a user ID and password when you defined the agent site, the program will run as a user process. If you did not specify a user ID and password, the program will run as a system process.

You can run some programs as user processes and other programs as system processes on the same physical workstation. To do this, define two agent sites on the workstation: one that has a user ID and password, and one that does not.

For more information about defining agent sites, see “Defining agent sites” on page 42 and the online help.

## Specifying parameters

You specify the parameters to use on the Parameters page of the Program notebook, shown in Figure 16 on page 74.



| | Name | Value | De |
|---|---|---|---|
| 1 | Source Table/File Names | &STBNS | Source Table/File Na |
| 2 | Target Resource Database | &TDB | Target Resource Dat |
| 3 | Target Resource Access Us | &TUID | Target Resource Acc |
| 4 | Target Resource Access Pa | &TPWD | Target Resource Acc |
| 5 | Target Table/File Name | &TTBN | Target Table/File Na |
| 6 | File type modifier string | | File type modifier stri |

Command Line String (for viewing only)

&STBNS &TDB &TUID &TPWD &TTBN

*Figure 16. Defining parameters for a Visual Warehouse program*

To add parameters with predefined tokens, click **Add**.

You can use predefined tokens for some parameters. Visual Warehouse substitutes values for the tokens at run time. For example, there is a token for the database name of the target resource for a business view, &TDB. If you include this token in your parameter list, Visual Warehouse provides the name of the database defined in the Warehouse notebook of the business view that uses the program. This allows you to change the value that is passed depending on which business view uses the program.

If you want your Visual Warehouse program to extract data from the sources that are defined to the business view that uses the program, you can include the &STMT token in your parameter list. This token directs Visual Warehouse to generate the SELECT statement that is defined for the business view and then passes the statement to the program. Each token of the SELECT statement is passed to the program as a separate command-line parameter. 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 “Add Parameters” in the online help.

To define parameters without predefined tokens, click **Insert**. A new row is displayed in the grid. Type the name and value of the parameter in the new row.

For specific instructions on defining a Visual Warehouse program, see the online help.

## Using a Visual Warehouse program in a business view

After you define your Visual Warehouse program, you define business views that use the program for processing. You can define the business views as part of a warehouse, part of a subject, or both.

### Defining subjects

Business views in a warehouse generally must have a source and target table. Business views in a subject do not require source and target tables.

For example, if you want your business view to use the supplied Visual Warehouse program Synchronize dynamic VW metadata to DataGuide, which synchronizes timestamp metadata between Visual Warehouse and DataGuide, you can define the business view without a specific source or target.

Business views in a subject generally use Visual Warehouse programs. In some cases, these programs process interim versions of the data before writing the data to the warehouse. For example, you want to run a program that cleanses data before writing the data to the warehouse. You define two Visual Warehouse programs, each with associated business views:

- One program extracts data from a source and writes the data to a flat file.
- One program performs data cleansing on the data in the flat file and writes the data to the warehouse.

The business view for the program that writes the data to the flat file must be part of a subject, not a warehouse.

The business view for the program that writes the data to the warehouse can be a part of the same subject, a part of the warehouse to which the program writes data, or both.

To define a new subject:

1. Click the **Subjects** tab on the Visual Warehouse desktop.
2. Click **File —>New**.

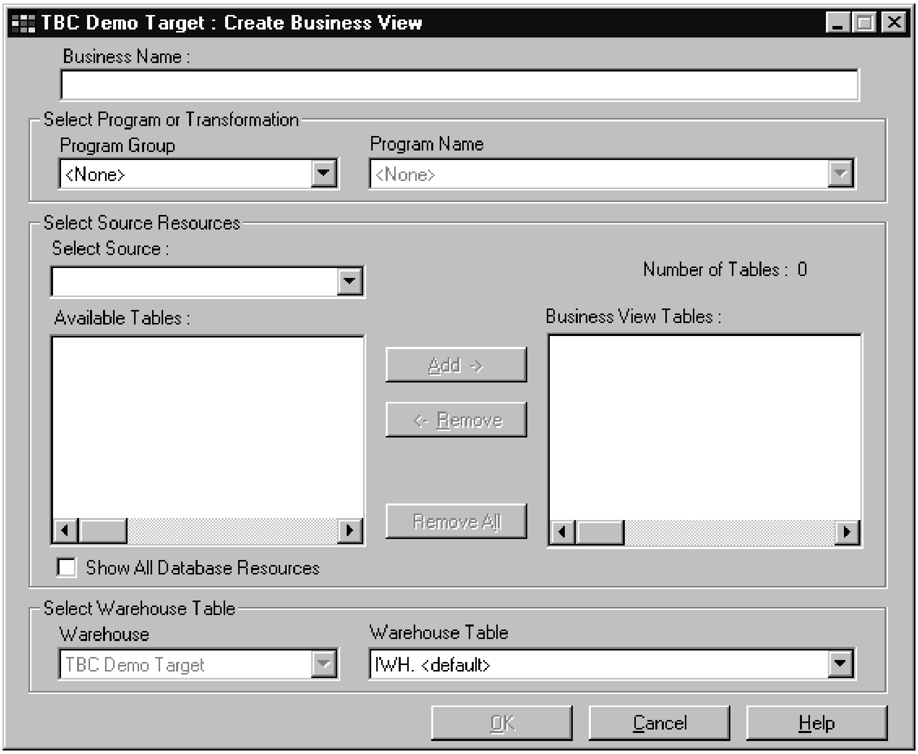
   The Subjects notebook opens.

3. Supply the following information:
   - The name of the subject
   - The security groups that are authorized to the subject

## Defining business views that use Visual Warehouse programs

To define the business view for a Visual Warehouse program:

1. Open the Visual Warehouse desktop.
2. Click the **Warehouses** or **Subjects** tab.
3. Double-click the warehouse or subject that you want to use. The Business View window opens.
4. Click **File —> New**. The Create Business View window, shown Figure 17, opens.
5. Select the source tables and the Visual Warehouse program to use in your business view.



TBC Demo Target : Create Business View

Program Group: <None>

Program Name: <None>

Number of Tables : 0

Warehouse: TBC Demo Target

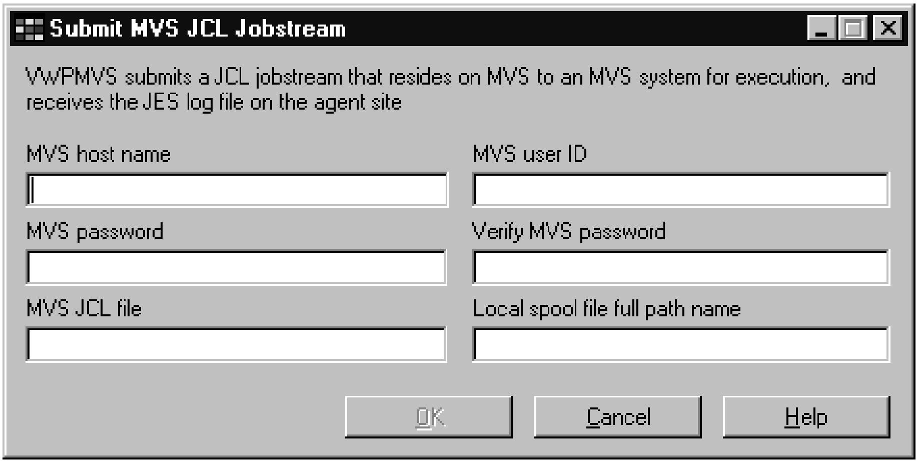
Warehouse Table: IWH. <default>

□ Show All Database Resources

*Figure 17. Create Business View window*

When you click **OK**, one of the following actions occurs:

- If you selected a supplied Visual Warehouse program or transformer, the window for the program or transformer opens. For example,Figure 18 shows the window for the Visual Warehouse program Submit MVS JCL Jobstream.



Submit MVS JCL Jobstream

VWPMVS submits a JCL jobstream that resides on MVS to an MVS system for execution, and receives the JES log file on the agent site

MVS host name

MVS user ID

MVS password

Verify MVS password

MVS JCL file

Local spool file full path name

Figure 18. Submit MVS JCL Jobstream window

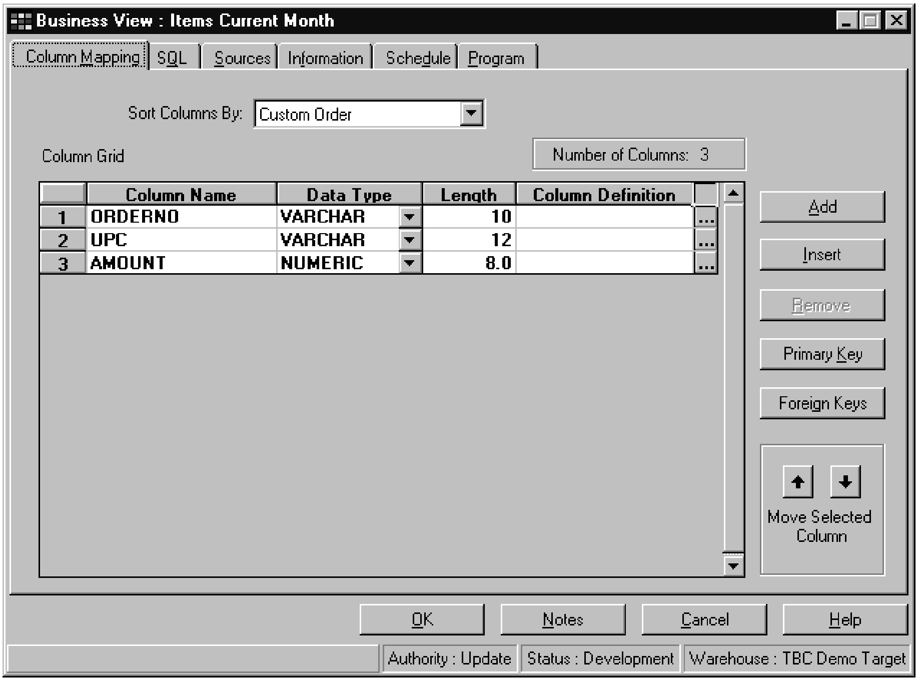
- If you selected a Visual Warehouse program that you defined, the Business View notebook opens.

For more information about defining a business view, see “Defining a business view” in the online help.

## Mapping columns

If you want Visual Warehouse to create a target table in a warehouse for a Visual Warehouse program to update, define column mappings for the associated business view. When you promote the business view to test status, Visual Warehouse creates the target table. Then your program can populate the target table.

For example, you want to define a business view with a Visual Warehouse program that loads data from a flat file into a database. You want Visual Warehouse to create the table into which the program will load data. You map the columns of the source data as shown in Figure 19 on page 78.



Sort Columns By: Custom Order

Number of Columns: 3

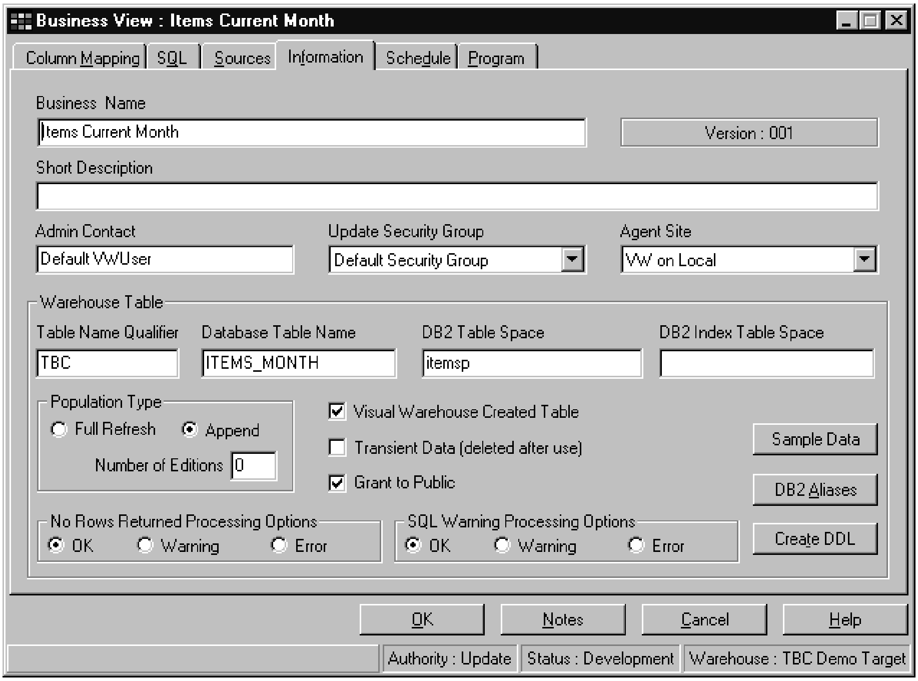
| | Column Name | Data Type | Length | Column Definition |
|---|---|---|---|---|
| 1 | ORDERNO | VARCHAR | 10 | |
| 2 | UPC | VARCHAR | 12 | |
| 3 | AMOUNT | NUMERIC | 8.0 | |

Authority : Update | Status : Development | Warehouse : TBC Demo Target

*Figure 19. Mapping columns for a business view using a Visual Warehouse program*

## Specifying the target table

If you want Visual Warehouse to create the target table in which the program will write the data, specify that Visual Warehouse is to create the table and supply a target table name, as shown in Figure 20 on page 79. (You must select Target Table/File Name as one of the parameters for your program.)



Business Name: Items Current Month

Version : 001

Short Description:

Admin Contact: Default VWUser

Update Security Group: Default Security Group

Agent Site: VW on Local

Warehouse Table

Table Name Qualifier: TBC

Database Table Name: ITEMS_MONTH

DB2 Table Space: itemsp

DB2 Index Table Space:

Population Type: ○ Full Refresh ● Append

Number of Editions: 0

☑ Visual Warehouse Created Table

□ Transient Data (deleted after use)

☑ Grant to Public

No Rows Returned Processing Options: ● OK ○ Warning ○ Error

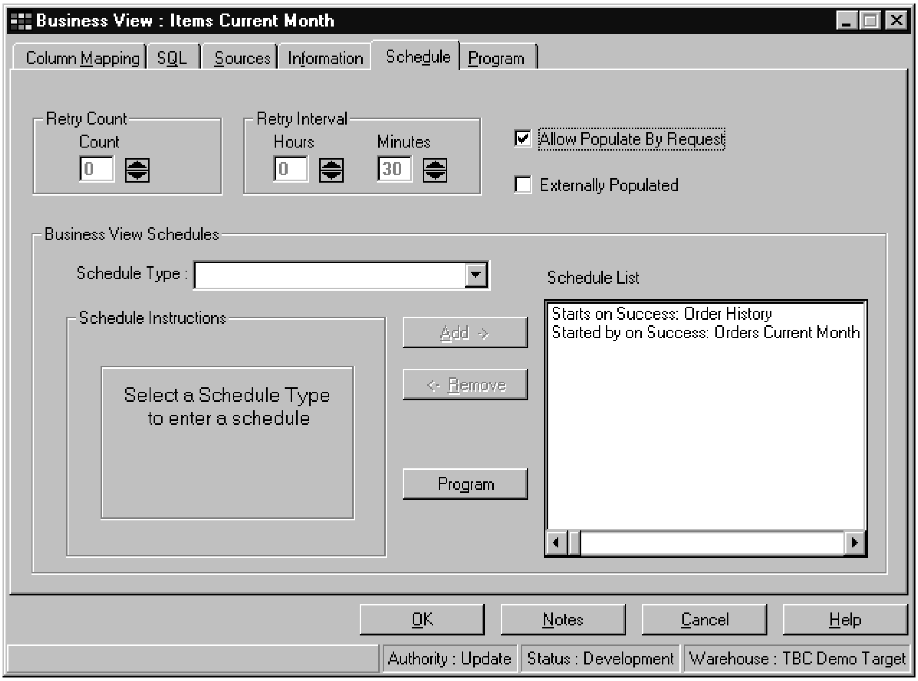
SQL Warning Processing Options: ● OK ○ Warning ○ Error

Authority : Update | Status : Development | Warehouse : TBC Demo Target

*Figure 20. Specifying that Visual Warehouse manage the target table*

## Scheduling the business view

You decide that your business view will start only when another related business view, Orders Current Month, finishes processing. When your business view finishes processing, it will start another business view, Order History, which includes data from the Orders Current Month business view and your business view, Items Current Month. Figure 21 on page 80 shows the relationship between the business views.



Business View : Items Current Month

Retry Count: Count 0

Retry Interval: Hours 0, Minutes 30

☑ Allow Populate By Request

□ Externally Populated

Schedule Type :

Select a Schedule Type to enter a schedule

Schedule List

Starts on Success: Order History
Started by on Success: Orders Current Month

Authority : Update | Status : Development | Warehouse : TBC Demo Target

*Figure 21. Specifying a schedule for a business view with a Visual Warehouse program*

When your business view runs, your Visual Warehouse program will run.

For more information about scheduling business views, see “Scheduling a production business view” on page 64.

For more information about combining business views, see “Combining business views” on page 66.

## Changing parameters

Business views use the parameters that are defined to a Visual Warehouse program as they exist at the moment the program is attached to the business view. Therefore, any changes you make to the parameters in a program will not affect copies of the program that are currently defined to a business view. To change the parameters of a program that is attached to a business view, use the Program page of the Business View notebook.

Any other changes you make to a program definition will be updated in all copies of the program.

## Testing business views that use Visual Warehouse programs

After you define a business view that uses a Visual Warehouse program, promote the business view to test status and run it.

To promote the business view:

1. From the Business View List window, select the business view.
2. Click **Status —> Promote to Test**.

   The business view's status changes to test when the business view is promoted.

To run the business view:

1. Open the Operations Work in Progress window.
2. Click **New**.

   The Run Business View window opens.
3. Select the business view from the list.
4. Click **OK**.

   The business view is displayed on the Operations Work in Progress window with a status of **Populating**. The processing should complete successfully. However, if the processing fails:

   a. Select the business view.

   b. Click **Log**.

      The Log Viewer window opens.

   c. Look for log records with a message type of Run Time Error.

   d. Select a record.

   e. 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 in the Return Codes section of the online help for the program.

      Transformer messages are different than other messages in Visual Warehouse:

      - Transformer error messages start with IWH14.
      - Transformer error messages, warning messages, and returned SQL codes are stored as secondary codes. Therefore, if a message starts with IWH14, a transformer (stored procedure) caused the error. If the secondary code includes an SQLCODE, an SQL statement in the 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.

f. Look in the log files for the program 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 *x*:\vwswin\logging\ on Windows NT and OS/2, and /var/IWH on UNIX.

   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, Visual Warehouse writes the log files to the new directory you specified, but the value of VWS_LOGGING does not change.

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

One common problem is caused by running Visual Warehouse agents as a system process rather than a user process. When the 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 agent being unable to find the Visual Warehouse program (**Error RC2** = 128 or **Error RC2** = 1 in the Log Viewer Details window) or being unable to initalize the program.

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

5. To avoid these problems, perform the following steps:
   a. Change the Visual Warehouse server, logger, and agent daemon processes to run as user processes by performing the following steps:
      1) Double-click the Services icon in the 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 your workstation.

b. If you are using a supplied Essbase 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 Essbase 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.

d. Verify that the Visual Warehouse solution is installed in the system path, not user path.

## Writing your own program for use with Visual Warehouse

You can write Visual Warehouse programs in any language that supports one of the following program types: executable, batch program, dynamic load library, or stored procedure.

If the Visual Warehouse program or a cascade program has a program type of executable, batch program, batch command file, or dynamic load library, it must reside on the agent site. The Visual 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.

To change the Visual Warehouse server, logger, and agent daemon processes to run as user processes:

1. Double-click the Services icon in the Control Panel folder.
2. Stop the Agent service.
3. Select the Agent service and click **Startup**.
4. Click **This Account**.
5. Click the push button after the **This Account** field to select a user ID.

   The user ID must have administrator 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 workstation.

## Extending Visual Warehouse with Visual Warehouse programs

If you write Visual Warehouse programs using Object REXX for Windows, complete the following procedure to enable these programs to run under Windows NT:

1. Define the Visual 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 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 vwadmin USING vwpass
   ```

### Passing parameters

At run time, Visual Warehouse generates a command-line parameter list that it passes as input to your Visual Warehouse program. Whenever possible, test your program from the command line before using it in a business view.

**Example:** The Visual Warehouse program DB2 UDB load replace (VWPLOADR) selects data from a file and loads the data into a database. It uses the following parameters:

- Source file name
- Target database name
- Target database user ID
- Target database password
- Target table name
- Column delimiter

The program gets the parameters as shown in Figure 22 on page 85:

```
char * sourceFile;
     sourceFile = argv[1]:
     char * dbName;
     dbName = argv[2];
     char * dbUser;
     dbUser = argv[3];
     char * dbPassword
     dbPassword = argv[4];
     char * dbTable;
     dbTable = argv[5]
     char * fileMod;
     if(argc>6) fileMod = argv[6];
     else fileMod = NULL;
```

*Figure 22. Reading parameters from the command line*

The program uses the target parameters to connect to the target database: The program then uses the DB2 load utility to load data into the database.

```
rc = SQLConnect (hdbc, (SQLCHAR *)dbName, SQL_NTS,
           (SQLCHAR *)dbUser, SQL_NTS,       /* UID */
           (SQLCHAR *)dbPassword, SQL_NTS);  /* Password */
```

*Figure 23. Connecting to the target database*

You can see the complete listing of the DB2 UDB load replace program in the VWSAMPLE\VWP directory of the Visual Warehouse CD-ROM.

## Returning status information

After your Visual Warehouse program runs, it must return a return code to the business view that uses the program. The return code must be a positive integer. If your program does not return a return code, the business view using the program fails. Visual Warehouse displays the return code in the **Error RC2** field of the Log Details window when the value of **Error RC1** is 8410.

Your Visual Warehouse program can return additional status information to Visual Warehouse:

- Another return code, which can be the same as or different from the code that is returned by the Visual Warehouse program.
- A warning flag that indicates that SQL returned a warning code, or that the Visual Warehouse program found no data in the source table. When this flag is set, the business view that uses this program will have a status of **Warning** in the Operations Work in Progress window.
- A message, which Visual Warehouse will display in the **System Message** field of the Log Viewer Details window

- The number of rows of data processed by the Visual Warehouse program, which Visual Warehouse will display in the Log Viewer Details window for the business view.
- The number of bytes of data processed by the Visual Warehouse program, which Visual Warehouse will display in the Log Viewer Details window for the business view.
- The SQLSTATE return code, which Visual Warehouse will display in the SQL state field of the Log Viewer Details window.

The Visual Warehouse agent transfers the additional status information to the Visual Warehouse server.

### Transferring the information to Visual Warehouse

To transfer the additional status information to the Visual Warehouse agent, your Visual Warehouse program must create a file, called a *feedback file*, containing the additional status information. The path and file name (*processid*.log, where *processid* is the ID of the agent process) for the feedback file must be the value of the VWP_LOG environment variable, which the agent sets before it calls the Visual Warehouse program. After the Visual Warehouse program finishes running, the agent checks whether the feedback file exists. If it exists, the agent processes the file. Otherwise, the agent will do nothing. If the Visual Warehouse program cannot create the file, it should continue to run.

### Format of the feedback file

Your Visual Warehouse program can write the additional status information to the feedback file in any order, but must use the following format to identify information. Enclose each item returned within the begin tag <TAG> and end tag </TAG> in the following list. Each begin tag must be followed by its end tag; you cannot include two begin tags in a row. For example, the following tag format is valid:

<RC>...</RC>...<MSG>...</MSG>

The following embedded tag format is not valid:

<RC>...<MSG>...</RC>...</MSG>

You can specify the following information in the feedback file:

**Return code**
<RC>*return code*</RC>, where *return code* is a positive integer

**Return code warning flag**
<WARNING>1</WARNING> sets the return code warning flag to on.

**Visual Warehouse system message**

<MSG>*message text*\n</MSG>, where

*message text*

Is the text of one or more messages

**\n**

Is 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 24 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 24. Example of the feedback file*

## How the feedback determines the business view status

The return codes and business view status for the Visual Warehouse 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 Visual Warehouse program
- Whether a feedback file exists
- The value of the return code in the feedback file
- Whether the warning flag is set on

Table 7 lists the possible combinations of these values and the results that they produce.

*Table 7. Feedback file conditions and results*

| **Conditions** | | | | **Results** | |
|---|---|---|---|---|---|
| | | | | **Business view status**[1] | **Values of Error RC1 and RC2** |
| Visual Warehouse program return code is 0 | No feedback file exists[2] | | | Successful | **RC1** = 0;<br>**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;<br>**RC2** = 0 |
| | | | The value of <WARNING> in the feedback file is 1 | Warning | **RC1** = 0;<br>**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<br>(the Visual Warehouse program failed);<br>**RC2** = the value of <RC> in the feedback file |
| | | | The value of <WARNING> in the feedback file is 1 | Warning | **RC1** = 0;<br>**RC2** = the value of <RC> in the feedback file |

*Table 7. Feedback file conditions and results (continued)*

| Conditions | | | | Results | |
|---|---|---|---|---|---|
| | | | | **Business view status[1]** | **Values of Error RC1 and RC2** |
| The Visual Warehouse program return code is nonzero | No feedback file exists[2] | | | Failed | **RC1** = 8410 (the Visual Warehouse program failed);<br>**RC2** = the code returned by the Visual Warehouse 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;<br>**RC2** = 0 |
| | | | The value of <WARNING> in the feedback file is 1 | Warning | **RC1** = 0;<br>**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 Visual Warehouse program failed);<br>**RC2** = the code returned by the Visual Warehouse program |
| | | | The value of <WARNING> in the feedback file is 1 | Warning | **RC1** = 0;<br>**RC2** = the value of <RC> in the feedback file |

**Notes:**

1. The business view processing status, as displayed in the Operations Work in Progress window.
2. Visual Warehouse checks for the existence of the feedback file, regardless of whether the return code for the Visual Warehouse program is 0 or non-zero.
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 6. Using the supplied Visual Warehouse programs

Visual Warehouse supplies a set of Visual Warehouse programs and stored procedures that move and transform data.

You can use the Visual Warehouse programs for bulk data operations that are managed by Visual Warehouse.

Visual Warehouse also provides a set of transformers you can use to transform data in a business view. You can use the transformers to refine the data that you move with the Visual Warehouse programs.

The following tables list the supplied Visual Warehouse programs by program group. Table 8 lists Visual Warehouse file programs.

*Table 8. Visual Warehouse file programs*

| Name | Description |
|---|---|
| Copy file using FTP (VWPRCPY) | Copies files on the agent site to and from a remote host. |
| Run FTP command file (VWPFTP) | Runs any FTP command file, the name of which is passed as a parameter. |
| Submit MVS JCL Jobstream (VWPMVS) | Submits a JCL jobstream to an MVS system for processing. |

Table 9 lists Visual Warehouse DB2 programs.

*Table 9. Visual Warehouse DB2 programs*

| Name | Description |
|---|---|
| DB2 UDB load replace (VWPLOADR) | Loads data from a flat file into a DB2 UDB database, replacing existing data. |
| DB2 UDB load insert (VWPLOADI) | Loads data from a flat file into a DB2 table, appending it to existing data. |
| Load flat file into DB2 UDB EEE (AIX only) (VWPLDPR) | Loads data from a flat file into a DB2 EEE database, replacing existing data. |
| DB2 UDB data export (VWPEXPT1) | Exports data from a local DB2 UDB database to a flat file. |
| Data export with ODBC to file (VWPEXPT2) | Selects data in a table, contained in a database registered in ODBC, and writes the data to a delimited flat file. |
| DB2 UDB runstats (VWPSTATS) | Runs the DB2 RUNSTATS utility on the specified table. |

*Table 9. Visual Warehouse DB2 programs (continued)*

| Name | Description |
|---|---|
| DB2 UDB reorg (VWPREORG) | Runs the DB2 REORG and RUNSTATS utilities on the specified table. |

Table 10 lists DB2 OLAP Server™ programs.

*Table 10. DB2 OLAP Server programs*

| Name | Description |
|---|---|
| DB2 OLAP: 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. |
| DB2 OLAP: 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. |
| DB2 OLAP: Load data from SQL table with load rules (ESSDATA3) | Loads data from an SQL source into a multidimensional DB2 OLAP Server database using load rules. |
| DB2 OLAP: Update outline from file (ESSOTL1) | Updates a DB2 OLAP Server outline from a source flat file using load rules. |
| DB2 OLAP: Update outline from SQL table (ESSOTL2) | Updates a DB2 OLAP Server outline from an SQL source using load rules. |
| DB2 OLAP: Default calc (ESSCALC1) | Calls the default DB2 OLAP Server calc script associated with the target database. |
| DB2 OLAP: Calc with calc rules (ESSCALC2) | Applies the specified calc script to a DB2 OLAP Server database. |

Table 11 lists metadata synchronization programs.

*Table 11. Metadata synchronization programs*

| Name | Description |
|---|---|
| Synchronize DB2 OLAP to DataGuide | Synchronizes OLAP metadata between DB2 OLAP Server or Hyperion Essbase and DataGuide. |
| Synchronize dynamic VW metadata to DataGuide | Updates the DataGuide information catalog with metadata about the business view processing. |
| Synchronize static VW metadata to DataGuide | Updates the DataGuide information catalog with changes to Visual Warehouse definitional metadata. |

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

Visual Warehouse also provides a set of transformers you can use to transform data in a business view. For information about the transformers, see “Using transformers” on page 96.

Visual Warehouse provides source code, a definition, and an executable file for each supplied Visual Warehouse program:

- For AIX, Sun Solaris, OS/2, AS/400, and Windows NT, the executable files for a particular operating system are packaged with the agent for that operating system. You install the Visual Warehouse programs when you install the agent code.
- The definition of each program is added during initialization. To see a list of the definitions, click **Definitions –> Visual Warehouse Programs** in the Visual Warehouse desktop.
- The source ($C^{++}$) files are in the VWSAMPLE\VWP directory of the Visual Warehouse CD-ROM. You can use the source files as a starting point for writing your own Visual Warehouse programs.

Visual Warehouse provides a Visual Warehouse program definition and an installable stored procedure or user-defined function for each of the transformers. Visual Warehouse does not provide the source code for the transformers.

Some supplied Visual Warehouse programs and transformers require that the business view be part of a warehouse, and some require that the business view be part of a subject. For a summary of the settings you need to specify for a business view that uses a supplied program or transformer, see the online help.

To use the supplied Visual Warehouse programs and transformers, you must define business views that use the programs or transformers. When you create the business view, Visual Warehouse guides you through selecting source and target tables and entering parameters for the program. For more information, see “Using a Visual Warehouse program in a business view” on page 75.

You also can use the supplied Visual Warehouse programs to familiarize yourself with using Visual Warehouse programs. Then you can write your own Visual Warehouse programs. For information on writing your own Visual Warehouse program, see “Writing your own program for use with Visual Warehouse” on page 83.

## Using the bulk load and export programs

You can use the supplied export programs, such as VWPEXPT1, to extract data from a DB2 database and write it to a flat file. You can use the supplied load programs, such as VWPLOADR, to extract data from a flat file and write it to another DB2 database.

**Recommendation:** Use these programs instead of Visual Warehouse's SQL processing when there is a substantial amount of data to load.

These 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 specified by the VWS_LOGGING environment variable. The default value of VWS_LOGGING is *x*:\vwswin\logging\ on Windows NT and OS/2, and /var/IWH on UNIX.

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, Visual Warehouse 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, open the online help and click **Contents —> Using the Supplied Visual Warehouse programs**.

## Using the DB2 OLAP Server programs

You can use Visual Warehouse programs to automatically refresh a DB2 OLAP Server database. Visual Warehouse can help you schedule three types of operations with the DB2 OLAP Server:

- Loading data
- Updating dimensions
- Running calculations

To use Visual Warehouse to load data into the DB2 OLAP Server database:

1. Using the Essbase Application Manager, create the DB2 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 Visual Warehouse program.

2. Using the Essbase Application Manager, define the outline for the database.
3. Define the data that you want to extract from the operational sources for DB2 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 business view that extracts data from the operational data sources and build the data as defined in the previous step. During this step, you can also define business views that the Hyperion Essbase SQL Drill-Through function will use.
5. Promote the business view to test status and run it at least once.
6. Using the Essbase Application Manager, write and test the load rules that will load the data sources into the Essbase database. Save the load rules into the database or as files on the Visual Warehouse agent site.

   During this step, you can also define calculation scripts to run after the data is loaded. Save the calculation scripts in files on the Visual Warehouse agent site.

   For information on defining load rules and calculation scripts, see the *DB2 OLAP Server Database Administrator’s Guide*.
7. Define a business view that uses one of the Visual Warehouse programs for Hyperion Essbase, such as DB2 OLAP: Load data from flat file with load rules (ESSDATA2). On the Schedule notebook page, specify that the business view that extracts data is to start this business view.
8. Promote the business view to test status and run it at least once.
9. Define a schedule for the business view that extracts data, and promote the business view to production status.

Figure 25 on page 96 shows the data flow between Visual Warehouse and DB2 OLAP Server.

### Using the supplied Visual Warehouse programs

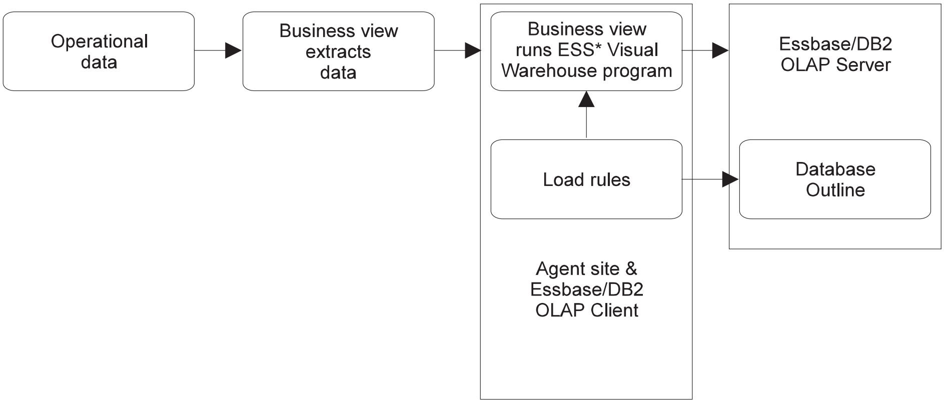

Operational data
Business view extracts data
Business view runs ESS* Visual Warehouse program
Essbase/DB2 OLAP Server
Load rules
Database Outline
Agent site & Essbase/DB2 OLAP Client

*Figure 25. Data flow between Visual Warehouse and DB2 OLAP Server*

For more information on completing the steps in this process, see the Visual Warehouse online help and *DB2 OLAP Server: Using DB2 OLAP Server*.

## Using transformers

Visual Warehouse transformers are stored procedures and user-defined functions that you can use to transform data in a business view. You can use transformers to clean, invert, and pivot data; generate keys and period tables; and calculate various statistics.

The transformers are grouped into the following program groups:

- Warehouse transformers
  - Clean Data
  - Generate Key Table
  - Generate Period Table
  - Invert Data
  - Pivot Data
- Statistical transformers
  - ANOVA
  - Calculate Statistics
  - Calculate Subtotals
  - Chi-square
  - Correlation
  - Moving Average
  - Regression

- User-defined functions: Format Data and Time transformer

This section provides an overview of each transformer. Each overview also provides an example of how the transformer changes data.

## Setting up the transformers

Before you use the transformers:

- Install the transformers in the target database.
- 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.

To use the transformers, you must create a business view:

1. Create a business view in a warehouse where the transformers are installed.
2. In the **Program Group** field of the Business View notebook, select either **Statistical transformers** or **Warehouse transformers**.
3. In the **Program Name** field, select the transformer that you want to use.
4. Select a source table.

   **Exception:** The Generate Key Table, Moving Average, and Calculate Subtotals transformers do not use source tables.
5. Click **OK**.

   The transformer window opens.
6. Enter the necessary values in the transformer window. After you enter values, click **OK**.
7. Click **OK** in the Business View notebook to close and save the business view.

**Restrictions:** When you use the transformers, the Column Mapping page of the Business View notebook has the following restrictions:

- You cannot change the order of the output columns for the Correlation, Chi-square, ANOVA, and Regression transformers.
- You can change the column name and description, but you cannot change other characteristics.
- To add or remove columns, use the **Edit** button on the **Program** page of the Business View notebook to open the transformer window. You cannot add or remove columns from the Column Mapping page of the Business View notebook.

For more information about the transformers, including task help, see the online help.

## Warehouse transformers

Use warehouse transformers to perform the following basic data transformations:

- Cleaning data
- Generating key columns
- Generating period tables
- Inverting data
- Pivoting data

### Clean Data transformer

The Clean Data transformer provides a rules-based find-and-replace function. This transformer locates values or ranges of values that you specify in the data columns of the source table that your business view accesses. Then the transformer updates corresponding columns with replacement values that you specify in the table to which your business view writes.

Use the Clean Data transformer to clean and standardize data values after load or import, as part of the more general datamart building application. Do not use this transformer as a general-purpose data column editor.

You can use the Clean Data transformer only if your source and target tables are in the same database. The target is the business view table that is created by default.

You can use the Clean Data transformer to perform the following tasks:

- Replace values in selected data columns that are missing, nonvalid, 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 choose to ignore case and white space when locating strings, and you can specify a tolerance value for numeric data.

**Rules tables:** Before you use the Clean Data transformer, you must create a rules table. A rules table designates the values that the Clean Data transformer uses for the find-and-replace function. The rules table must be in the same database as the source and target tables.

At a minimum, a rules table must contain at least two columns. One column contains the values to be found, and the other column contains the replacement values. The rows in each column correspond to each other.

If a value in the input table is not found in the rules table, the Clean Data transformer copies the value from the input table to the output table.

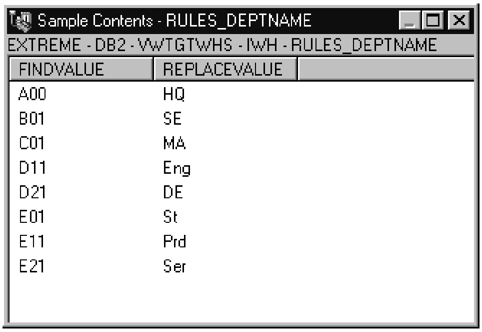
Examples of rules tables are shown in “Types”.

**Types:** When you use the Clean Data transformer, you specify one of several types. For each type (except Carry-over), you specify a rules table. The types are:

**Find and Replace**

Use this type to perform basic find-and-replace functions.

A sample rules table for the Find and Replace type is shown in Figure 26:



EXTREME - DB2 - VWTGTWHS - IWH - RULES_DEPTNAME

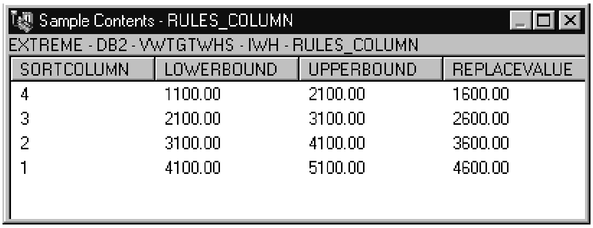
| FINDVALUE | REPLACEVALUE |
|---|---|
| A00 | HQ |
| B01 | SE |
| C01 | MA |
| D11 | Eng |
| D21 | DE |
| E01 | St |
| E11 | Prd |
| E21 | Ser |

*Figure 26. Sample rules table for the Find and Replace type*

**Discretize**

Use this type to perform find-and-replace functions within a range of values.

A sample rules table for the Discretize type is shown in Figure 27 :

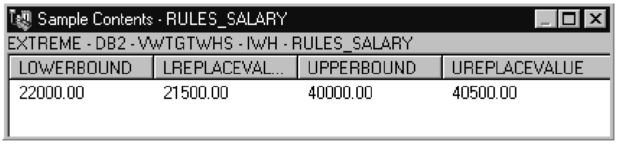

Sample Contents - RULES_COLUMN
EXTREME - DB2 - VWTGTWHS - IWH - RULES_COLUMN

| SORTCOLUMN | LOWERBOUND | UPPERBOUND | REPLACEVALUE |
|---|---|---|---|
| 4 | 1100.00 | 2100.00 | 1600.00 |
| 3 | 2100.00 | 3100.00 | 2600.00 |
| 2 | 3100.00 | 4100.00 | 3600.00 |
| 1 | 4100.00 | 5100.00 | 4600.00 |

*Figure 27. Sample rules table for the Discretize type*

**Clip** Use this type to perform find-and-replace functions on a range of values that fall outside of a specified value range.

A sample rules table for the Clip type is shown in Figure 28:


Sample Contents - RULES_SALARY
EXTREME - DB2 - VWTGTWHS - IWH - RULES_SALARY

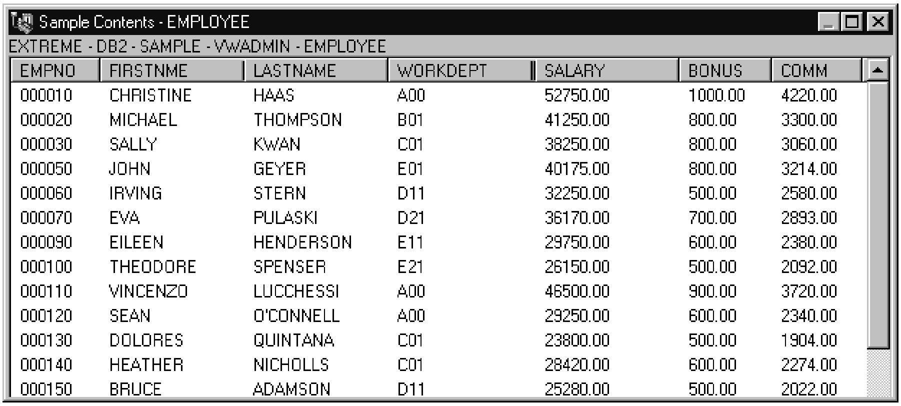
| LOWERBOUND | LREPLACEVAL... | UPPERBOUND | UREPLACEVALUE |
|---|---|---|---|
| 22000.00 | 21500.00 | 40000.00 | 40500.00 |

*Figure 28. Sample rules table for the Clip type*

**Carry-over**
Use this type to copy one or more columns in the input table to the output table. The Clean Data transformer does not change the values in these columns.

The Carry-over type does not use a rules table.

**Example:** For example, suppose that you have this input data:



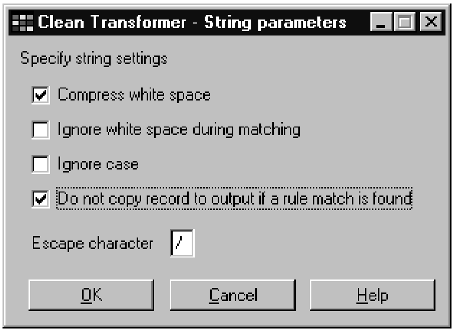
Sample Contents - EMPLOYEE

EXTREME - DB2 - SAMPLE - VWADMIN - EMPLOYEE

| EMPNO | FIRSTNME | LASTNAME | WORKDEPT | SALARY | BONUS | COMM |
|---|---|---|---|---|---|---|
| 000010 | CHRISTINE | HAAS | A00 | 52750.00 | 1000.00 | 4220.00 |
| 000020 | MICHAEL | THOMPSON | B01 | 41250.00 | 800.00 | 3300.00 |
| 000030 | SALLY | KWAN | C01 | 38250.00 | 800.00 | 3060.00 |
| 000050 | JOHN | GEYER | E01 | 40175.00 | 800.00 | 3214.00 |
| 000060 | IRVING | STERN | D11 | 32250.00 | 500.00 | 2580.00 |
| 000070 | EVA | PULASKI | D21 | 36170.00 | 700.00 | 2893.00 |
| 000090 | EILEEN | HENDERSON | E11 | 29750.00 | 600.00 | 2380.00 |
| 000100 | THEODORE | SPENSER | E21 | 26150.00 | 500.00 | 2092.00 |
| 000110 | VINCENZO | LUCCHESSI | A00 | 46500.00 | 900.00 | 3720.00 |
| 000120 | SEAN | O'CONNELL | A00 | 29250.00 | 600.00 | 2340.00 |
| 000130 | DOLORES | QUINTANA | C01 | 23800.00 | 500.00 | 1904.00 |
| 000140 | HEATHER | NICHOLLS | C01 | 28420.00 | 600.00 | 2274.00 |
| 000150 | BRUCE | ADAMSON | D11 | 25280.00 | 500.00 | 2022.00 |

*Figure 29. Input data for the Clean Data transformer*

Specify these settings for the Find and Replace type for the LASTNAME column:



Clean Transformer - String parameters

Specify string settings

☑ Compress white space

☐ Ignore white space during matching

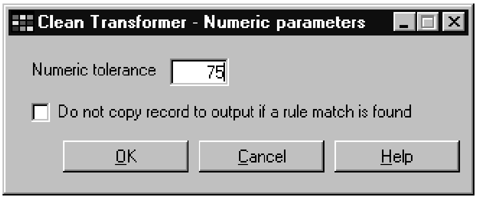
☐ Ignore case

☑ Do not copy record to output if a rule match is found

Escape character: /

*Figure 30. Settings for the Find and Replace type of the Clean Data transformer*

Specify these settings for the Discretize type for the COMM column:



Clean Transformer - Numeric parameters

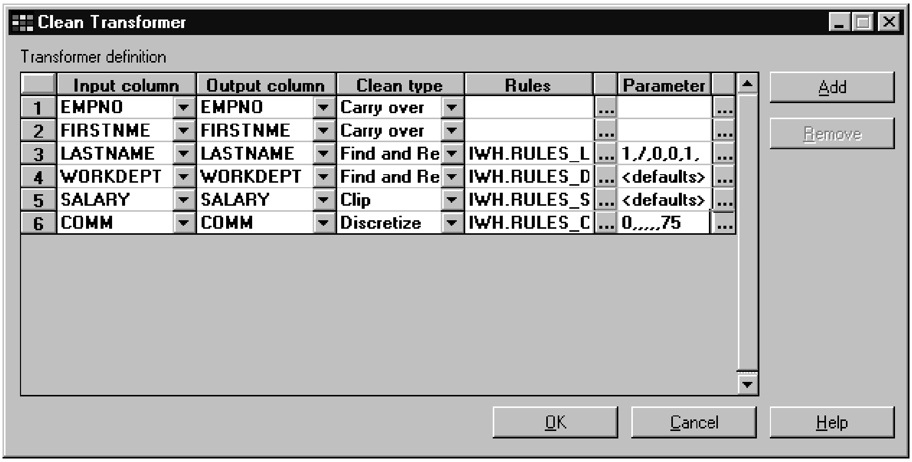
Numeric tolerance: 75

☐ Do not copy record to output if a rule match is found

*Figure 31. Settings for the Discretize type of the Clean Data transformer*

Specify the rules tables shown in “Types” on page 99.

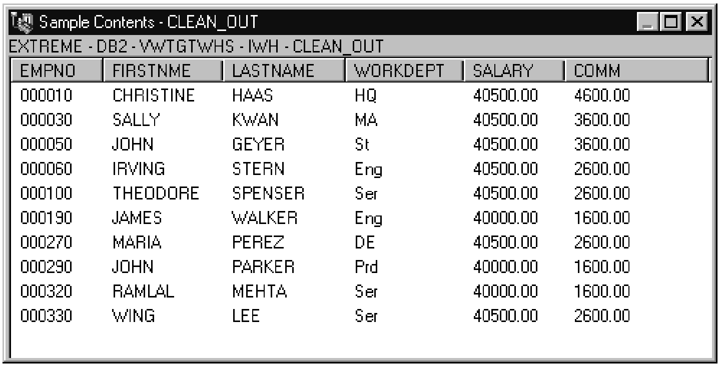
The Clean Data transformer window looks like this:



| | Input column | Output column | Clean type | Rules | | Parameter | |
|---|---|---|---|---|---|---|---|
| 1 | EMPNO | EMPNO | Carry over | | ... | | ... |
| 2 | FIRSTNME | FIRSTNME | Carry over | | ... | | ... |
| 3 | LASTNAME | LASTNAME | Find and Re | IWH.RULES_L | ... | 1,/,0,0,1, | ... |
| 4 | WORKDEPT | WORKDEPT | Find and Re | IWH.RULES_D | ... | <defaults> | ... |
| 5 | SALARY | SALARY | Clip | IWH.RULES_S | ... | <defaults> | ... |
| 6 | COMM | COMM | Discretize | IWH.RULES_C | ... | 0,,,,,75 | ... |

*Figure 32. Settings in the Clean Data transformer*

The Clean Data transformer produces the following output data:



EXTREME - DB2 - VWTGTWHS - IWH - CLEAN_OUT

| EMPNO | FIRSTNME | LASTNAME | WORKDEPT | SALARY | COMM |
|---|---|---|---|---|---|
| 000010 | CHRISTINE | HAAS | HQ | 40500.00 | 4600.00 |
| 000030 | SALLY | KWAN | MA | 40500.00 | 3600.00 |
| 000050 | JOHN | GEYER | St | 40500.00 | 3600.00 |
| 000060 | IRVING | STERN | Eng | 40500.00 | 2600.00 |
| 000100 | THEODORE | SPENSER | Ser | 40500.00 | 2600.00 |
| 000190 | JAMES | WALKER | Eng | 40000.00 | 1600.00 |
| 000270 | MARIA | PEREZ | DE | 40500.00 | 2600.00 |
| 000290 | JOHN | PARKER | Prd | 40000.00 | 1600.00 |
| 000320 | RAMLAL | MEHTA | Ser | 40000.00 | 1600.00 |
| 000330 | WING | LEE | Ser | 40500.00 | 2600.00 |

*Figure 33. Output data for the Clean Data transformer*

### Generate Key Table transformer

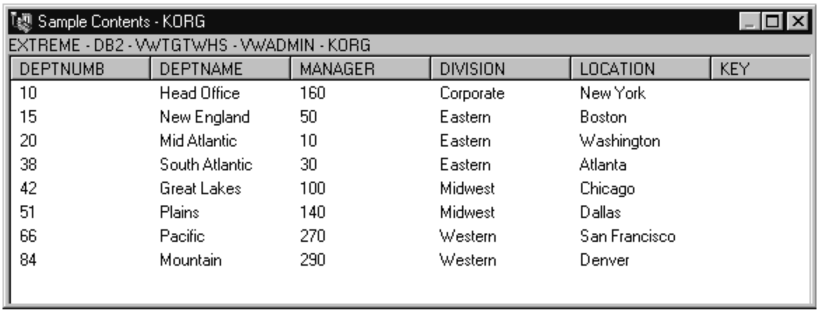
The Generate Key Table transformer creates or changes a sequence of key values in an existing table. If key values already exist in the key column, the Generate Key Table transformer overwrites them. The key column must have a data type of INTEGER.

For this transformer, the source and target table are the same. The transformer adds data to the source table to create the target table.

Use this transformer to perform the following tasks:

- Update the null values in an existing key column.
- Update the null values in an existing key column, based on the key values in another column.
- Assign key values to a specific column.

For example, suppose you have this input data (note the lack of data in the Key column):

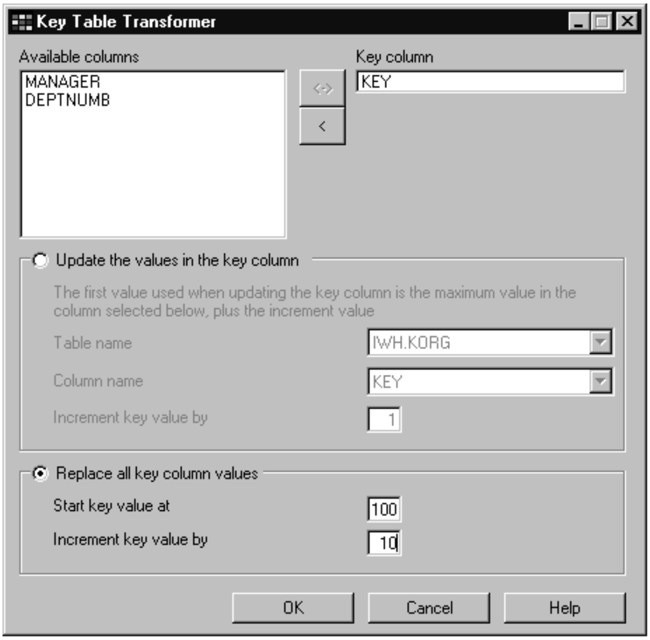


EXTREME - DB2 - VWTGTWHS - VWADMIN - KORG

| DEPTNUMB | DEPTNAME | MANAGER | DIVISION | LOCATION | KEY |
|---|---|---|---|---|---|
| 10 | Head Office | 160 | Corporate | New York | |
| 15 | New England | 50 | Eastern | Boston | |
| 20 | Mid Atlantic | 10 | Eastern | Washington | |
| 38 | South Atlantic | 30 | Eastern | Atlanta | |
| 42 | Great Lakes | 100 | Midwest | Chicago | |
| 51 | Plains | 140 | Midwest | Dallas | |
| 66 | Pacific | 270 | Western | San Francisco | |
| 84 | Mountain | 290 | Western | Denver | |

*Figure 34. Input data for the Generate Key Table transformer*

Specify these settings:



Key Table Transformer

Available columns
MANAGER
DEPTNUMB

Key column: KEY

○ Update the values in the key column

The first value used when updating the key column is the maximum value in the column selected below, plus the increment value

Table name: IWH.KORG

Column name: KEY

Increment key value by: 1

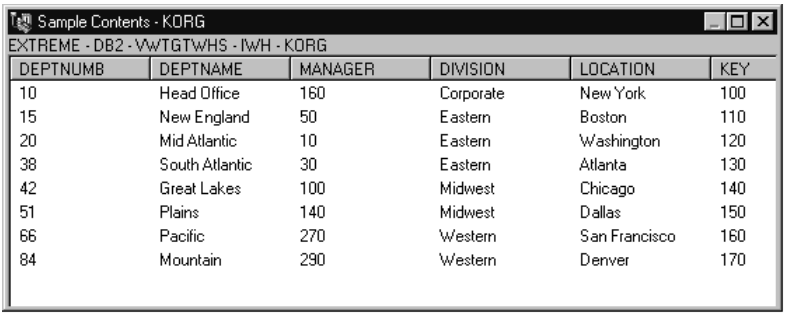
● Replace all key column values

Start key value at: 100

Increment key value by: 10

Figure 35. Settings for the Generate Key Table transformer

The Generate Key Table transformer produces the following output data:



Sample Contents - KORG

EXTREME - DB2 - VWTGTWHS - IWH - KORG

| DEPTNUMB | DEPTNAME | MANAGER | DIVISION | LOCATION | KEY |
|---|---|---|---|---|---|
| 10 | Head Office | 160 | Corporate | New York | 100 |
| 15 | New England | 50 | Eastern | Boston | 110 |
| 20 | Mid Atlantic | 10 | Eastern | Washington | 120 |
| 38 | South Atlantic | 30 | Eastern | Atlanta | 130 |
| 42 | Great Lakes | 100 | Midwest | Chicago | 140 |
| 51 | Plains | 140 | Midwest | Dallas | 150 |
| 66 | Pacific | 270 | Western | San Francisco | 160 |
| 84 | Mountain | 290 | Western | Denver | 170 |

Figure 36. Output data for the Generate Key Table transformer

Notice that the transformer adds data to the KEY column in the output table.

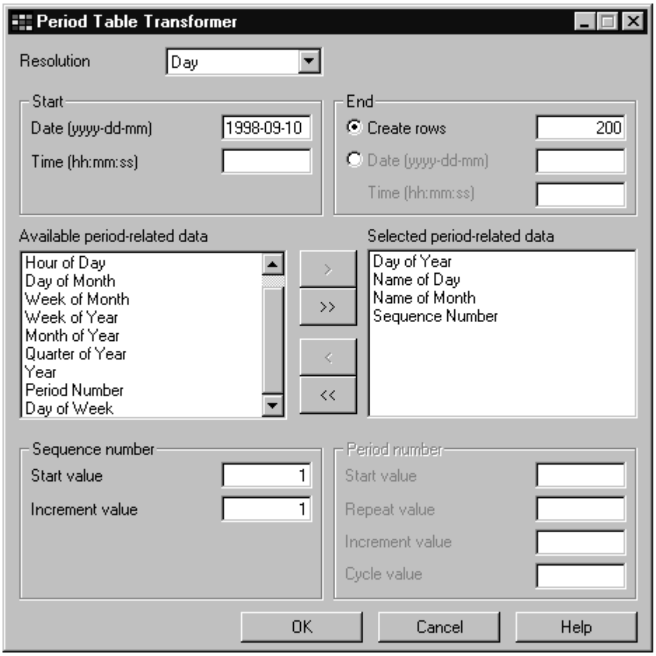
## Generate Period Table transformer

The Generate Period Table transformer creates a period table. This transformer does not require an input table. A period table contains a column that consists of a generated date, time, or timestamp value. The period table can also contain one or more optional output columns that are based on the date/time value for the row. You can also specify parameters for the optional output columns.

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.

For example, specify these settings:



Period Table Transformer

Resolution: Day

Start
Date (yyyy-dd-mm): 1998-09-10
Time (hh:mm:ss):

End
● Create rows: 200
○ Date (yyyy-dd-mm):
Time (hh:mm:ss):

Available period-related data:
Hour of Day
Day of Month
Week of Month
Week of Year
Month of Year
Quarter of Year
Year
Period Number
Day of Week

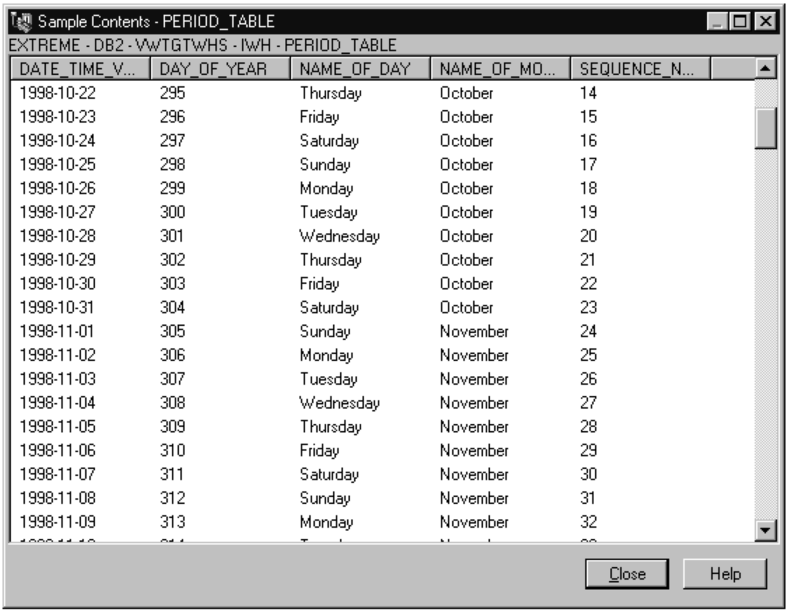
Selected period-related data:
Day of Year
Name of Day
Name of Month
Sequence Number

Sequence number
Start value: 1
Increment value: 1

Period number
Start value:
Repeat value:
Increment value:
Cycle value:

Figure 37. Settings for the Generate Period Table transformer

The Generate Period Table transformer produces the following output data:



EXTREME - DB2 - VWTGTWHS - IWH - PERIOD_TABLE

| DATE_TIME_V... | DAY_OF_YEAR | NAME_OF_DAY | NAME_OF_MO... | SEQUENCE_N... |
|---|---|---|---|---|
| 1998-10-22 | 295 | Thursday | October | 14 |
| 1998-10-23 | 296 | Friday | October | 15 |
| 1998-10-24 | 297 | Saturday | October | 16 |
| 1998-10-25 | 298 | Sunday | October | 17 |
| 1998-10-26 | 299 | Monday | October | 18 |
| 1998-10-27 | 300 | Tuesday | October | 19 |
| 1998-10-28 | 301 | Wednesday | October | 20 |
| 1998-10-29 | 302 | Thursday | October | 21 |
| 1998-10-30 | 303 | Friday | October | 22 |
| 1998-10-31 | 304 | Saturday | October | 23 |
| 1998-11-01 | 305 | Sunday | November | 24 |
| 1998-11-02 | 306 | Monday | November | 25 |
| 1998-11-03 | 307 | Tuesday | November | 26 |
| 1998-11-04 | 308 | Wednesday | November | 27 |
| 1998-11-05 | 309 | Thursday | November | 28 |
| 1998-11-06 | 310 | Friday | November | 29 |
| 1998-11-07 | 311 | Saturday | November | 30 |
| 1998-11-08 | 312 | Sunday | November | 31 |
| 1998-11-09 | 313 | Monday | November | 32 |

*Figure 38. Output table for the Generate Period Table transformer*

### Invert Data transformer

Use the Invert Data transformer to invert the order of the rows and columns in a table. When you use the Invert Data transformer, the rows in the source table become columns in the output table, and the columns in the input table become rows in the output table. The order of data among the columns, from top to bottom, is maintained and placed in rows, from left to right.

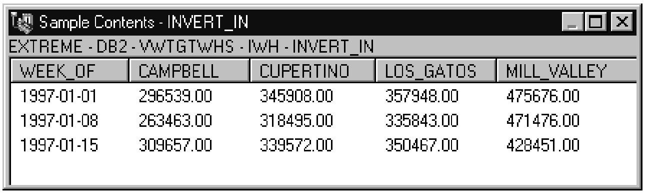
For example, consider the input table as a matrix. This transformer swaps the data in the table around a diagonal line that extends from the upper left of the table to the lower right. Then the transformer writes the transformed data to the target table.

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 have 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*.

For example, suppose that you have this input data:



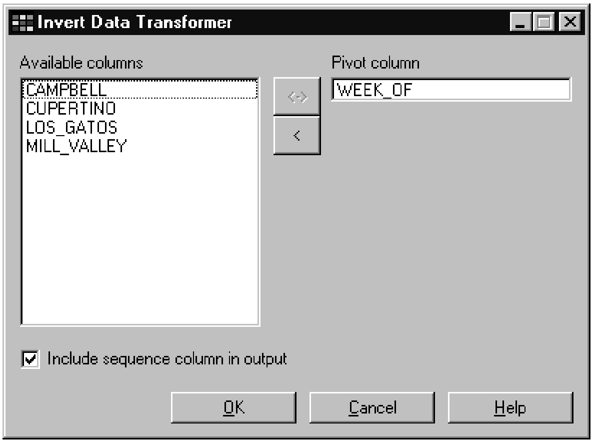
Sample Contents - INVERT_IN

EXTREME - DB2 - VWTGTWHS - IWH - INVERT_IN

| WEEK_OF | CAMPBELL | CUPERTINO | LOS_GATOS | MILL_VALLEY |
|---|---|---|---|---|
| 1997-01-01 | 296539.00 | 345908.00 | 357948.00 | 475676.00 |
| 1997-01-08 | 263463.00 | 318495.00 | 335843.00 | 471476.00 |
| 1997-01-15 | 309657.00 | 339572.00 | 350467.00 | 428451.00 |

*Figure 39. Input data for the Invert Data transformer*

Specify these settings:



Invert Data Transformer

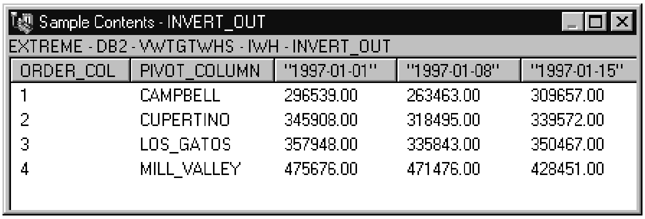
Available columns:
CAMPBELL
CUPERTINO
LOS_GATOS
MILL_VALLEY

Pivot column: WEEK_OF

☑ Include sequence column in output

*Figure 40. Settings for the Invert Data transformer*

The Invert Data transformer produces the following output data:



Sample Contents - INVERT_OUT

EXTREME - DB2 - VWTGTWHS - IWH - INVERT_OUT

| ORDER_COL | PIVOT_COLUMN | "1997-01-01" | "1997-01-08" | "1997-01-15" |
|---|---|---|---|---|
| 1 | CAMPBELL | 296539.00 | 263463.00 | 309657.00 |
| 2 | CUPERTINO | 345908.00 | 318495.00 | 339572.00 |
| 3 | LOS_GATOS | 357948.00 | 335843.00 | 350467.00 |
| 4 | MILL_VALLEY | 475676.00 | 471476.00 | 428451.00 |

*Figure 41. Output data for the Invert Data transformer*

### Pivot Data transformer

Use the Pivot Data transformer to group related data from selected columns in a source table into a single column in a output table. This column in the output table is called a pivot column. You can create more than one pivot column.

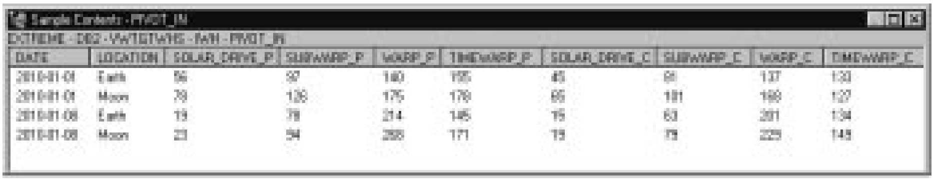
The data in each selected column in the source table is assigned a data group in the output table. You can create only one data group column in the output table. If you use the default choices, the Pivot Data transformer chooses the columns to pivot and creates the data groups. You can change the default choices by clicking **Advanced** in the transformer window to change the settings.

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 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.*

For example, suppose that you have this input data:



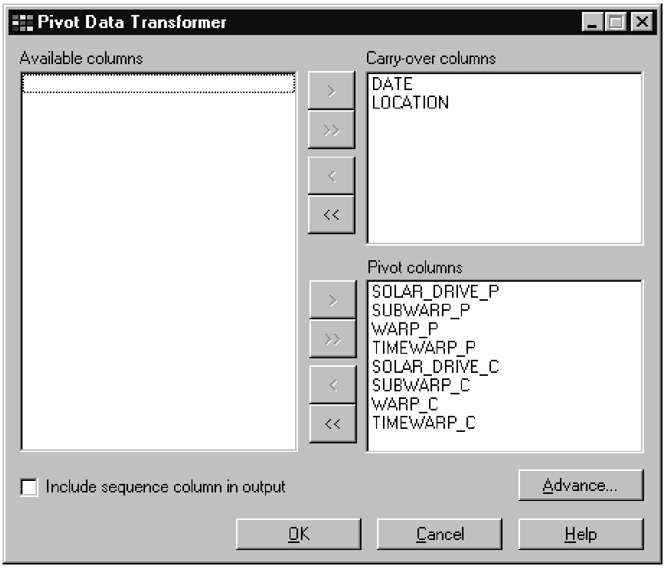
Sample Contents - PIVOT_IN

EXTREME - DB2 - VWTGTWHS - IWH - PIVOT_IN

| DATE | LOCATION | SOLAR_DRIVE_P | SUBWARP_P | WARP_P | TIMEWARP_P | SOLAR_DRIVE_C | SUBWARP_C | WARP_C | TIMEWARP_C |
|---|---|---|---|---|---|---|---|---|---|
| 2010-01-01 | Earth | 56 | 97 | 140 | 155 | 45 | 91 | 137 | 133 |
| 2010-01-01 | Moon | 79 | 126 | 175 | 178 | 65 | 181 | 168 | 127 |
| 2010-01-08 | Earth | 19 | 78 | 214 | 145 | 15 | 63 | 201 | 134 |
| 2010-01-08 | Moon | 23 | 54 | 288 | 171 | 19 | 79 | 225 | 148 |

Figure 42. Input data for the Pivot Data transformer

Specify these settings:



Pivot Data Transformer

Available columns

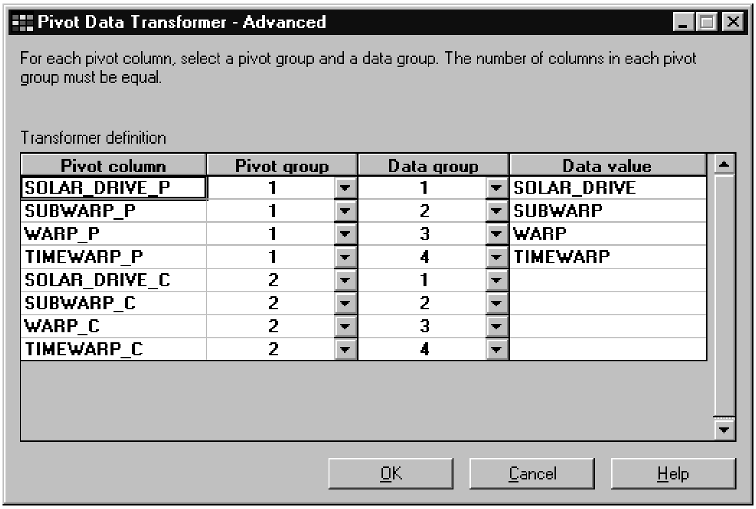
Carry-over columns: DATE, LOCATION

Pivot columns: SOLAR_DRIVE_P, SUBWARP_P, WARP_P, TIMEWARP_P, SOLAR_DRIVE_C, SUBWARP_C, WARP_C, TIMEWARP_C

□ Include sequence column in output

Figure 43. Settings for the Pivot Data transformer

Specify these advanced settings:



Pivot Data Transformer - Advanced

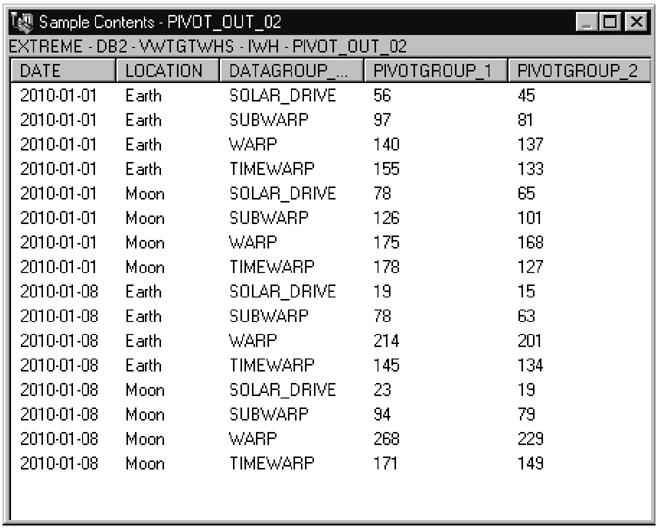
For each pivot column, select a pivot group and a data group. The number of columns in each pivot group must be equal.

Transformer definition

| Pivot column | Pivot group | Data group | Data value |
|---|---|---|---|
| SOLAR_DRIVE_P | 1 | 1 | SOLAR_DRIVE |
| SUBWARP_P | 1 | 2 | SUBWARP |
| WARP_P | 1 | 3 | WARP |
| TIMEWARP_P | 1 | 4 | TIMEWARP |
| SOLAR_DRIVE_C | 2 | 1 | |
| SUBWARP_C | 2 | 2 | |
| WARP_C | 2 | 3 | |
| TIMEWARP_C | 2 | 4 | |

*Figure 44. Advanced settings for the Pivot Data transformer*

The Pivot Data transformer produces the following output data:



Sample Contents - PIVOT_OUT_02

EXTREME - DB2 - VWTGTWHS - IWH - PIVOT_OUT_02

| DATE | LOCATION | DATAGROUP_... | PIVOTGROUP_1 | PIVOTGROUP_2 |
|---|---|---|---|---|
| 2010-01-01 | Earth | SOLAR_DRIVE | 56 | 45 |
| 2010-01-01 | Earth | SUBWARP | 97 | 81 |
| 2010-01-01 | Earth | WARP | 140 | 137 |
| 2010-01-01 | Earth | TIMEWARP | 155 | 133 |
| 2010-01-01 | Moon | SOLAR_DRIVE | 78 | 65 |
| 2010-01-01 | Moon | SUBWARP | 126 | 101 |
| 2010-01-01 | Moon | WARP | 175 | 168 |
| 2010-01-01 | Moon | TIMEWARP | 178 | 127 |
| 2010-01-08 | Earth | SOLAR_DRIVE | 19 | 15 |
| 2010-01-08 | Earth | SUBWARP | 78 | 63 |
| 2010-01-08 | Earth | WARP | 214 | 201 |
| 2010-01-08 | Earth | TIMEWARP | 145 | 134 |
| 2010-01-08 | Moon | SOLAR_DRIVE | 23 | 19 |
| 2010-01-08 | Moon | SUBWARP | 94 | 79 |
| 2010-01-08 | Moon | WARP | 268 | 229 |
| 2010-01-08 | Moon | TIMEWARP | 171 | 149 |

*Figure 45. Output data for the Pivot Data transformer*

## Statistical transformers

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

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.

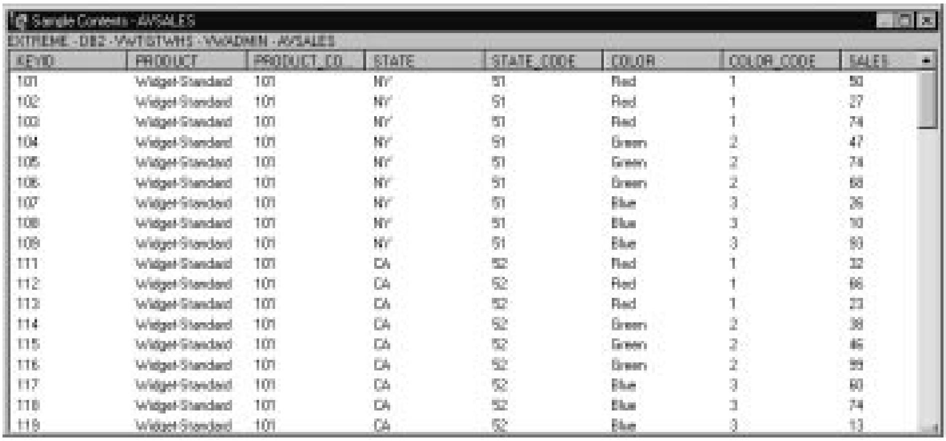
The ANOVA transformer creates an additional output table. You can specify the name of this table on the Program page of the Business View notebook.

You can perform three types of ANOVA calculations with this transformer:

- One-way ANOVA
- Two-way ANOVA
- Three-way ANOVA

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.

For example, suppose that you have this input data:



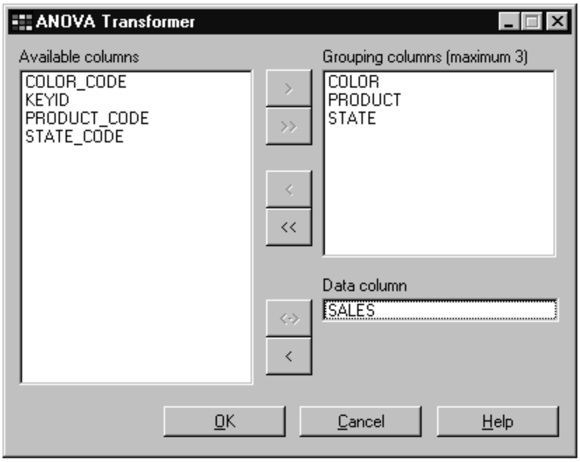
Sample Contents - AVSALES

EXTREME - DB2 - VWTGTWHS - VWADMIN - AVSALES

| KEYID | PRODUCT | PRODUCT_CO... | STATE | STATE_CODE | COLOR | COLOR_CODE | SALES |
|---|---|---|---|---|---|---|---|
| 101 | Widget-Standard | 101 | NY | 51 | Red | 1 | 50 |
| 102 | Widget-Standard | 101 | NY | 51 | Red | 1 | 27 |
| 103 | Widget-Standard | 101 | NY | 51 | Red | 1 | 74 |
| 104 | Widget-Standard | 101 | NY | 51 | Green | 2 | 47 |
| 105 | Widget-Standard | 101 | NY | 51 | Green | 2 | 74 |
| 106 | Widget-Standard | 101 | NY | 51 | Green | 2 | 68 |
| 107 | Widget-Standard | 101 | NY | 51 | Blue | 3 | 26 |
| 108 | Widget-Standard | 101 | NY | 51 | Blue | 3 | 10 |
| 109 | Widget-Standard | 101 | NY | 51 | Blue | 3 | 93 |
| 111 | Widget-Standard | 101 | CA | 52 | Red | 1 | 32 |
| 112 | Widget-Standard | 101 | CA | 52 | Red | 1 | 66 |
| 113 | Widget-Standard | 101 | CA | 52 | Red | 1 | 23 |
| 114 | Widget-Standard | 101 | CA | 52 | Green | 2 | 38 |
| 115 | Widget-Standard | 101 | CA | 52 | Green | 2 | 46 |
| 116 | Widget-Standard | 101 | CA | 52 | Green | 2 | 59 |
| 117 | Widget-Standard | 101 | CA | 52 | Blue | 3 | 60 |
| 118 | Widget-Standard | 101 | CA | 52 | Blue | 3 | 74 |
| 119 | Widget-Standard | 101 | CA | 52 | Blue | 3 | 13 |

Figure 46. Input data for the ANOVA transformer

Specify these settings:



ANOVA Transformer

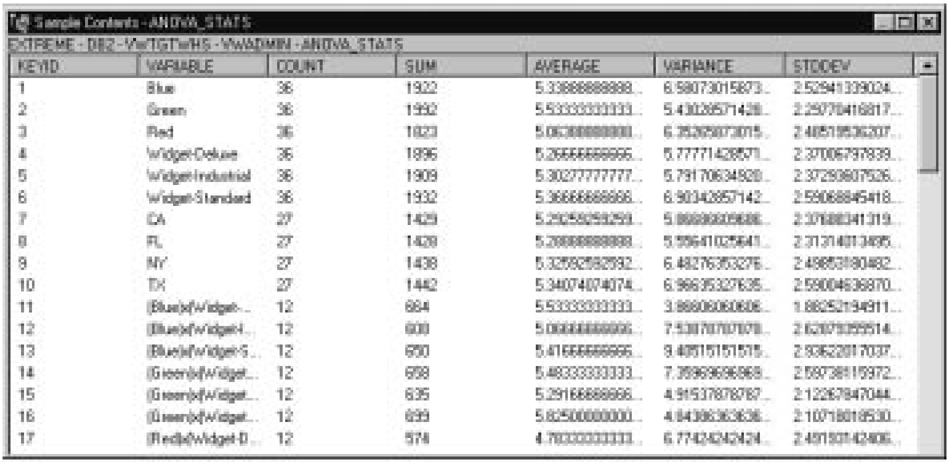
Available columns:
COLOR_CODE
KEYID
PRODUCT_CODE
STATE_CODE

Grouping columns (maximum 3):
COLOR
PRODUCT
STATE

Data column: SALES

Figure 47. Settings for the ANOVA transformer

The ANOVA transformer produces the following statistics table:

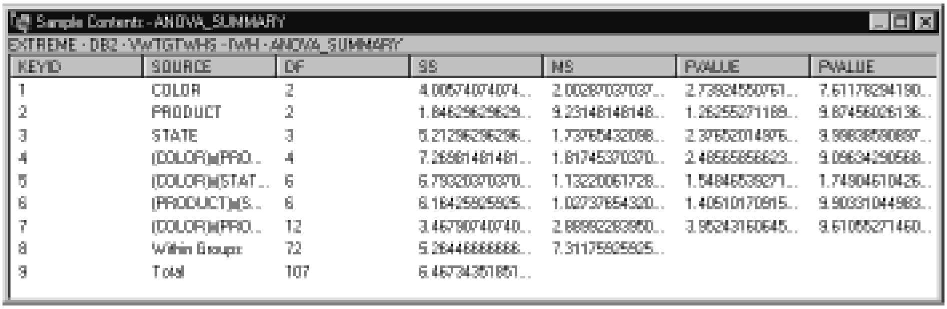

Sample Contents - ANOVA_STATS

EXTREME - DB2 - VWTGTWHS - VWADMIN - ANOVA_STATS

| KEYID | VARIABLE | COUNT | SUM | AVERAGE | VARIANCE | STDDEV |
|---|---|---|---|---|---|---|
| 1 | Blue | 36 | 1922 | 5.33888888888... | 6.58073015873... | 2.52941339024... |
| 2 | Green | 36 | 1992 | 5.53333333333... | 5.43028571428... | 2.25770416817... |
| 3 | Red | 36 | 1823 | 5.06388888888... | 6.35265873015... | 2.48519536207... |
| 4 | Widget-Deluxe | 36 | 1896 | 5.26666666666... | 5.77771428571... | 2.37006797839... |
| 5 | Widget-Industrial | 36 | 1909 | 5.30277777777... | 5.79170634920... | 2.37293807526... |
| 6 | Widget-Standard | 36 | 1932 | 5.36666666666... | 6.90342857142... | 2.59068845418... |
| 7 | CA | 27 | 1429 | 5.29259259259... | 5.86666666666... | 2.37688341319... |
| 8 | FL | 27 | 1428 | 5.28888888888... | 5.55641025641... | 2.31214013495... |
| 9 | NY | 27 | 1438 | 5.32592592592... | 6.48276353276... | 2.49853180482... |
| 10 | TX | 27 | 1442 | 5.34074074074... | 6.96635327635... | 2.59004636870... |
| 11 | (Blue)x(Widget-... | 12 | 664 | 5.53333333333... | 3.86606060606... | 1.88252194911... |
| 12 | (Blue)x(Widget-I... | 12 | 608 | 5.06666666666... | 7.53878787878... | 2.62079355514... |
| 13 | (Blue)x(Widget-S... | 12 | 650 | 5.41666666666... | 9.40515151515... | 2.93622017037... |
| 14 | (Green)x(Widget... | 12 | 658 | 5.48333333333... | 7.35969696969... | 2.59738115972... |
| 15 | (Green)x(Widget... | 12 | 635 | 5.29166666666... | 4.91537878787... | 2.12267847044... |
| 16 | (Green)x(Widget... | 12 | 699 | 5.82500000000... | 4.84386363636... | 2.10718018530... |
| 17 | (Red)x(Widget-D... | 12 | 574 | 4.78333333333... | 6.77424242424... | 2.49150142406... |

*Figure 48. Statistics output table for the ANOVA transformer*

The ANOVA transformer also produces the following summary table:


Sample Contents - ANOVA_SUMMARY

EXTREME - DB2 - VWTGTWHS - IWH - ANOVA_SUMMARY

| KEYID | SOURCE | DF | SS | MS | PVALUE | PVALUE |
|---|---|---|---|---|---|---|
| 1 | COLOR | 2 | 4.00574074074... | 2.00287037037... | 2.73824550761... | 7.61178294190... |
| 2 | PRODUCT | 2 | 1.84629629629... | 9.23148148148... | 1.26255271189... | 9.87456026136... |
| 3 | STATE | 3 | 5.21296296296... | 1.73765432098... | 2.37652014976... | 9.99838580897... |
| 4 | (COLOR)x(PRO... | 4 | 7.26981481481... | 1.81745370370... | 2.48565856623... | 9.09634290568... |
| 5 | (COLOR)x(STAT... | 6 | 6.79320370370... | 1.13220061728... | 1.54846538271... | 1.74904610426... |
| 6 | (PRODUCT)x(S... | 6 | 6.16425925925... | 1.02737654320... | 1.40510170915... | 9.90331044983... |
| 7 | (COLOR)x(PRO... | 12 | 3.46790740740... | 2.88992283950... | 3.95243160645... | 9.61055271460... |
| 8 | Within Groups | 72 | 5.26446666666... | 7.31175925925... | | |
| 9 | Total | 107 | 6.46734351851... | | | |

*Figure 49. Summary output table for the ANOVA transformer*

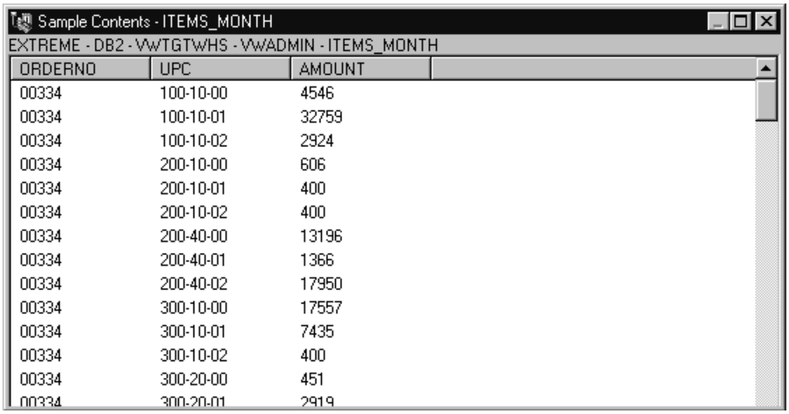
## Calculate Statistics transformer

You can use the Calculate Statistics transformer to calculate the following descriptive statistics on one or more numeric columns:

- Count
- Sum
- Average
- Variance
- Standard deviation
- Standard error
- Minimum
- Maximum

- Range
- Coefficient of variation

For example, suppose that you have this input data:



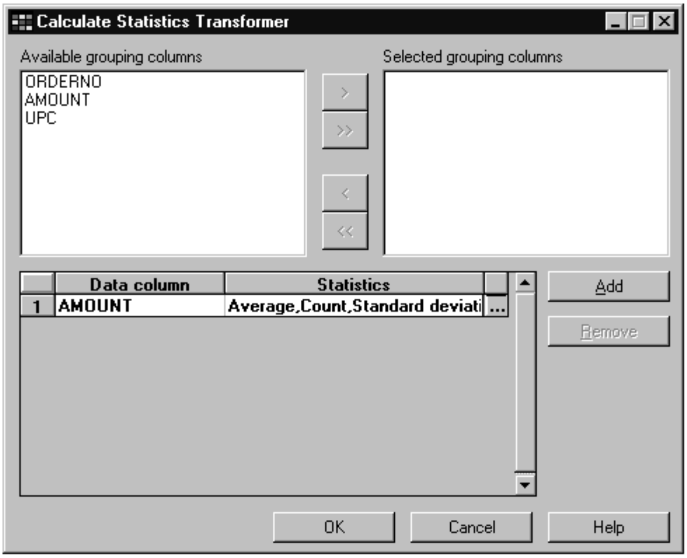
Sample Contents - ITEMS_MONTH

EXTREME - DB2 - VWTGTWHS - VWADMIN - ITEMS_MONTH

| ORDERNO | UPC | AMOUNT |
|---|---|---|
| 00334 | 100-10-00 | 4546 |
| 00334 | 100-10-01 | 32759 |
| 00334 | 100-10-02 | 2924 |
| 00334 | 200-10-00 | 606 |
| 00334 | 200-10-01 | 400 |
| 00334 | 200-10-02 | 400 |
| 00334 | 200-40-00 | 13196 |
| 00334 | 200-40-01 | 1366 |
| 00334 | 200-40-02 | 17950 |
| 00334 | 300-10-00 | 17557 |
| 00334 | 300-10-01 | 7435 |
| 00334 | 300-10-02 | 400 |
| 00334 | 300-20-00 | 451 |
| 00334 | 300-20-01 | 2919 |

*Figure 50. Input data for the Calculate Statistics transformer*

Specify these settings:



Calculate Statistics Transformer

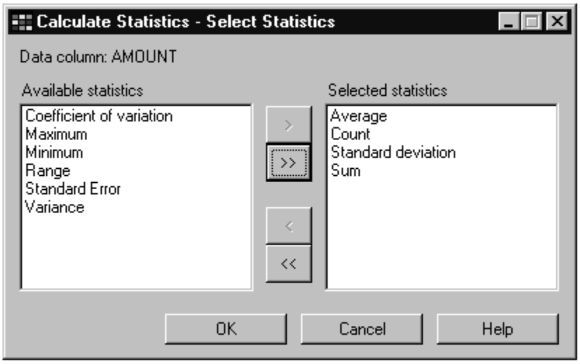
Available grouping columns: ORDERNO, AMOUNT, UPC

Selected grouping columns:

| | Data column | Statistics |
|---|---|---|
| 1 | AMOUNT | Average,Count,Standard deviati |

*Figure 51. Settings for the Calculate Statistics transformer*

Specify these statistics settings:



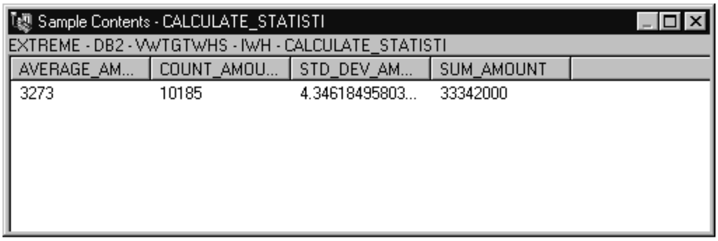
Data column: AMOUNT

Available statistics:
Coefficient of variation
Maximum
Minimum
Range
Standard Error
Variance

Selected statistics:
Average
Count
Standard deviation
Sum

Figure 52. Statistics settings for the Calculate Statistics transformer

The Calculate Statistics transformer produces the following output data:



EXTREME - DB2 - VWTGTWHS - IWH - CALCULATE_STATISTI

| AVERAGE_AM... | COUNT_AMOU... | STD_DEV_AM... | SUM_AMOUNT |
|---|---|---|---|
| 3273 | 10185 | 4.34618495803... | 33342000 |

Figure 53. Output data for the Calculate Statistics transformer

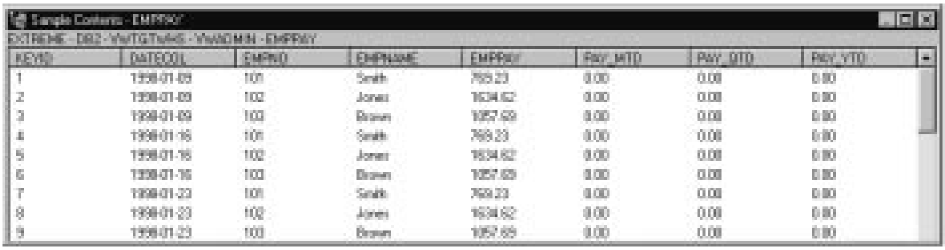
### Calculate Subtotals transformer

Use the Calculate Subtotals transformer to calculate the running subtotal for a column of numeric values. You can group the values by a period of time—either by week, semimonth, month, quarter, or year.

For accounting purposes, you might need to produce subtotals of numeric values for basic periods of time. This need arises frequently in payroll calculations where companies are required to produce month-to-date and year-to-date subtotals for various types of payroll data.

For this transformer, the source and target table are the same. The transformer adds data to the source table to create the target table. The source table must have a primary key.

For example, suppose that you have this input data:



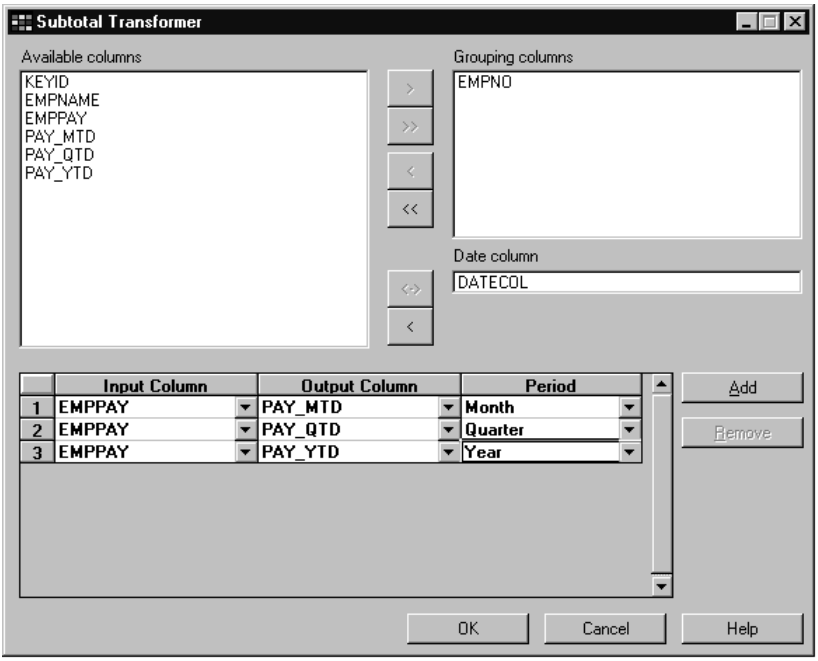
Sample Contents - EMPPAY

EXTREME - DB2 - VWTGTWHS - VWADMIN - EMPPAY

| KEYID | DATECOL | EMPNO | EMPNAME | EMPPAY | PAY_MTD | PAY_QTD | PAY_YTD |
|---|---|---|---|---|---|---|---|
| 1 | 1998-01-09 | 101 | Smith | 769.23 | 0.00 | 0.00 | 0.00 |
| 2 | 1998-01-09 | 102 | Jones | 1634.62 | 0.00 | 0.00 | 0.00 |
| 3 | 1998-01-09 | 103 | Brown | 1057.69 | 0.00 | 0.00 | 0.00 |
| 4 | 1998-01-16 | 101 | Smith | 769.23 | 0.00 | 0.00 | 0.00 |
| 5 | 1998-01-16 | 102 | Jones | 1634.62 | 0.00 | 0.00 | 0.00 |
| 6 | 1998-01-16 | 103 | Brown | 1057.69 | 0.00 | 0.00 | 0.00 |
| 7 | 1998-01-23 | 101 | Smith | 769.23 | 0.00 | 0.00 | 0.00 |
| 8 | 1998-01-23 | 102 | Jones | 1634.62 | 0.00 | 0.00 | 0.00 |
| 9 | 1998-01-23 | 103 | Brown | 1057.69 | 0.00 | 0.00 | 0.00 |

*Figure 54. Input data for the Calculate Subtotals transformer*

Specify these settings:



Subtotal Transformer

Available columns: KEYID, EMPNAME, EMPPAY, PAY_MTD, PAY_QTD, PAY_YTD

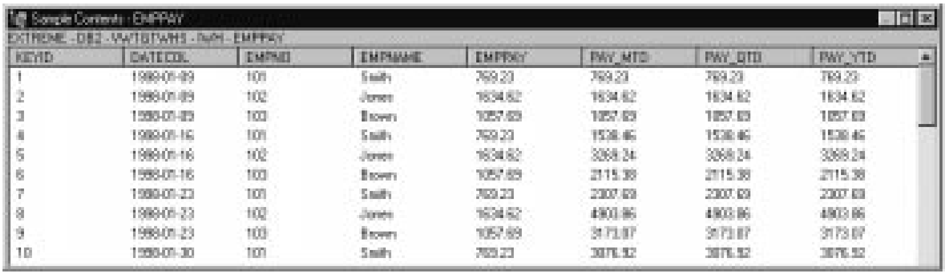
Grouping columns: EMPNO

Date column: DATECOL

|  | Input Column | Output Column | Period |
|---|---|---|---|
| 1 | EMPPAY | PAY_MTD | Month |
| 2 | EMPPAY | PAY_QTD | Quarter |
| 3 | EMPPAY | PAY_YTD | Year |

*Figure 55. Settings for the Calculate Subtotals transformer*

The Calculate Subtotals transformer produces the following output data:



EXTREME - DB2 - VWTGTWHS - IWH - EMPPAY

| KEYID | DATECOL | EMPNO | EMPNAME | EMPPAY | PAY_MTD | PAY_QTD | PAY_YTD |
|---|---|---|---|---|---|---|---|
| 1 | 1998-01-09 | 101 | Smith | 769.23 | 769.23 | 769.23 | 769.23 |
| 2 | 1998-01-09 | 102 | Jones | 1634.62 | 1634.62 | 1634.62 | 1634.62 |
| 3 | 1998-01-09 | 103 | Brown | 1057.69 | 1057.69 | 1057.69 | 1057.69 |
| 4 | 1998-01-16 | 101 | Smith | 769.23 | 1538.46 | 1538.46 | 1538.46 |
| 5 | 1998-01-16 | 102 | Jones | 1634.62 | 3269.24 | 3269.24 | 3269.24 |
| 6 | 1998-01-16 | 103 | Brown | 1057.69 | 2115.38 | 2115.38 | 2115.38 |
| 7 | 1998-01-23 | 101 | Smith | 769.23 | 2307.69 | 2307.69 | 2307.69 |
| 8 | 1998-01-23 | 102 | Jones | 1634.62 | 4903.86 | 4903.86 | 4903.86 |
| 9 | 1998-01-23 | 103 | Brown | 1057.69 | 3173.07 | 3173.07 | 3173.07 |
| 10 | 1998-01-30 | 101 | Smith | 769.23 | 3076.92 | 3076.92 | 3076.92 |

*Figure 56. Output data for the Calculate Subtotals transformer*

### Chi-square transformer

The Chi-square transformer performs the chi-square test and the chi-square goodness-of-fit test on columns of numeric data. These tests are nonparametric tests.

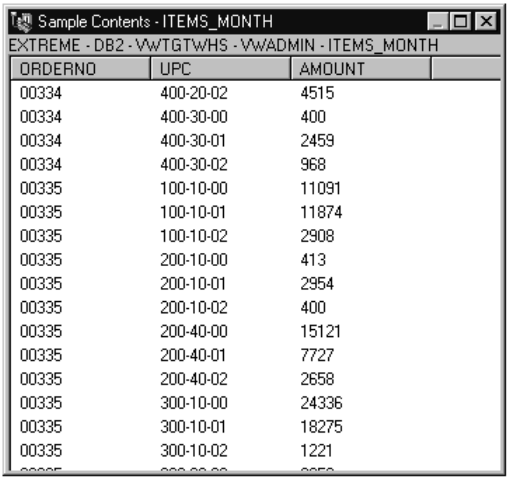
You can use the statistical results of these tests to make the following determinations:

- Whether the values of one variable are associated with 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. You can also use these tests 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 creates an additional output table. You can specify the name of this table on the Program page of the Business View notebook.

For example, suppose that you have this input data:



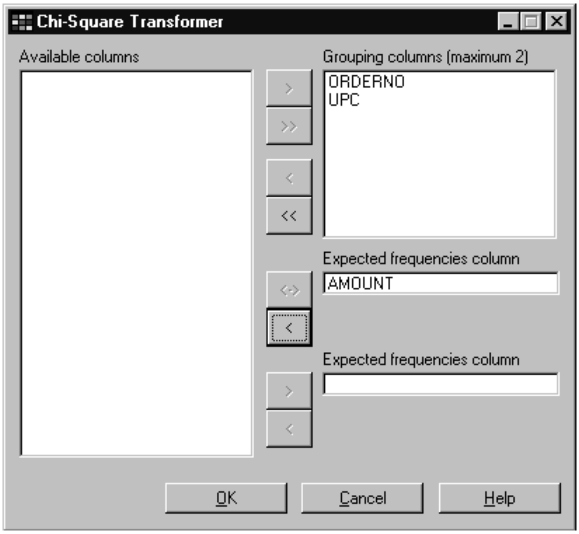
Sample Contents - ITEMS_MONTH

EXTREME - DB2 - VWTGTWHS - VWADMIN - ITEMS_MONTH

| ORDERNO | UPC | AMOUNT |
|---|---|---|
| 00334 | 400-20-02 | 4515 |
| 00334 | 400-30-00 | 400 |
| 00334 | 400-30-01 | 2459 |
| 00334 | 400-30-02 | 968 |
| 00335 | 100-10-00 | 11091 |
| 00335 | 100-10-01 | 11874 |
| 00335 | 100-10-02 | 2908 |
| 00335 | 200-10-00 | 413 |
| 00335 | 200-10-01 | 2954 |
| 00335 | 200-10-02 | 400 |
| 00335 | 200-40-00 | 15121 |
| 00335 | 200-40-01 | 7727 |
| 00335 | 200-40-02 | 2658 |
| 00335 | 300-10-00 | 24336 |
| 00335 | 300-10-01 | 18275 |
| 00335 | 300-10-02 | 1221 |

Figure 57. Input data for the Chi-square transformer

Specify these settings:



Chi-Square Transformer

Available columns:

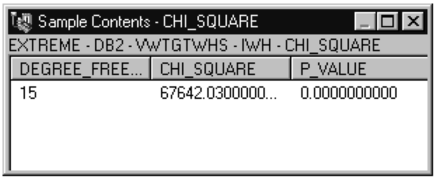
Grouping columns (maximum 2): ORDERNO, UPC

Expected frequencies column: AMOUNT

Expected frequencies column:

Figure 58. Settings for the Chi-square transformer

The Chi-square transformer produces the following output data:



Sample Contents - CHI_SQUARE

EXTREME - DB2 - VWTGTWHS - IWH - CHI_SQUARE

| DEGREE_FREE... | CHI_SQUARE | P_VALUE |
|---|---|---|
| 15 | 67642.0300000... | 0.0000000000 |

Figure 59. Output table for the Chi-square transformer

## Correlation transformer

Correlation analysis deals with determining 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 also treat the data in the input columns as a sample obtained from a larger population. You can use the Correlation transformer to test whether the attributes are correlated in the population. In this context, the *null hypothesis* asserts that the two attributes are not correlated, and the *alternative hypothesis* asserts that the attributes are correlated.

The Correlation transformer calculates any of the following correlation-related statistics on one or more pairs of columns:

**Correlation coefficient** *r*

The correlation coefficient *r* is a measure of the linear relationship between two attributes or columns of data. The correlation coefficient is also known as the Pearson product-moment correlation coefficient. The value of *r* can range from -1 to +1 and is independent of the units of measurement. A value of *r* near 0 indicates little correlation between attributes; a value near +1 or -1 indicates a high level of correlation.

When two attributes have a positive correlation coefficient, an increase in the value of one attribute indicates a likely increase in the value of the second attribute. A correlation coefficient of less than 0 indicates a negative correlation. That is, when one attribute shows an increase in value, the other attribute tends to show a decrease.

Consider two variables *x* and *y*:

- If r = 1, then *x* and *y* are perfectly positively correlated. The possible values of *x* and *y* all lie on a straight line with a positive slope in the (x,y) plane.

- If r = 0, then *x* and *y* are not correlated. They do not have an apparent linear relationship. However, this does not mean that *x* and *y* are statistically independent.
- If r = -1, then *x* and *y* are perfectly negatively correlated. The possible values of *x* and *y* all lie on a straight line with a negative slope in the (x,y) plane.

**Covariance**

Covariance is a measure of the linear relationship between two attributes or columns of data. The value of the covariance can range from –infinity to +infinity. 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 –infinity and +infinity. 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 * SQRT((n-2) / (1 - r*r))
```

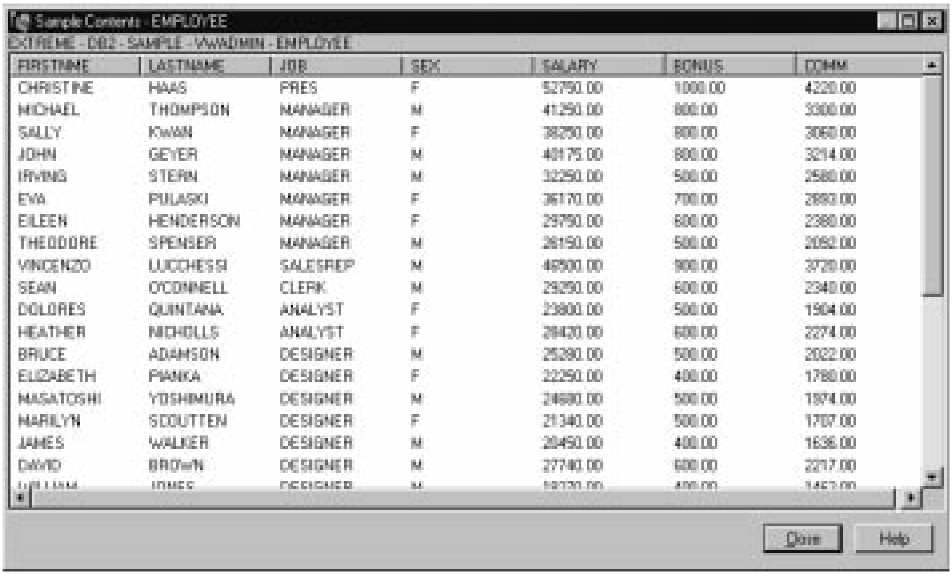
where *r* is the correlation coefficient, *n* is the number of input value pairs, and SQRT is the square root function.

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.

For example, suppose that you have this input data:



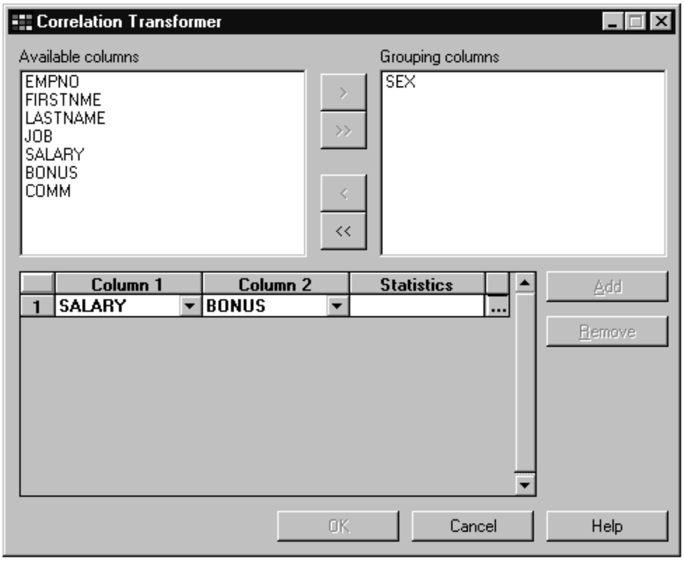
Sample Contents - EMPLOYEE

EXTREME - DB2 - SAMPLE - VWADMIN - EMPLOYEE

| FIRSTNME | LASTNAME | JOB | SEX | SALARY | BONUS | COMM |
|---|---|---|---|---|---|---|
| CHRISTINE | HAAS | PRES | F | 52750.00 | 1000.00 | 4220.00 |
| MICHAEL | THOMPSON | MANAGER | M | 41250.00 | 800.00 | 3300.00 |
| SALLY | KWAN | MANAGER | F | 38250.00 | 800.00 | 3060.00 |
| JOHN | GEYER | MANAGER | M | 40175.00 | 800.00 | 3214.00 |
| IRVING | STERN | MANAGER | M | 32250.00 | 500.00 | 2580.00 |
| EVA | PULASKI | MANAGER | F | 36170.00 | 700.00 | 2893.00 |
| EILEEN | HENDERSON | MANAGER | F | 29750.00 | 600.00 | 2380.00 |
| THEODORE | SPENSER | MANAGER | M | 26150.00 | 500.00 | 2092.00 |
| VINCENZO | LUCCHESSI | SALESREP | M | 46500.00 | 900.00 | 3720.00 |
| SEAN | O'CONNELL | CLERK | M | 29250.00 | 600.00 | 2340.00 |
| DOLORES | QUINTANA | ANALYST | F | 23800.00 | 500.00 | 1904.00 |
| HEATHER | NICHOLLS | ANALYST | F | 28420.00 | 600.00 | 2274.00 |
| BRUCE | ADAMSON | DESIGNER | M | 25280.00 | 500.00 | 2022.00 |
| ELIZABETH | PIANKA | DESIGNER | F | 22250.00 | 400.00 | 1780.00 |
| MASATOSHI | YOSHIMURA | DESIGNER | M | 24680.00 | 500.00 | 1974.00 |
| MARILYN | SCOUTTEN | DESIGNER | F | 21340.00 | 500.00 | 1707.00 |
| JAMES | WALKER | DESIGNER | M | 20450.00 | 400.00 | 1636.00 |
| DAVID | BROWN | DESIGNER | M | 27740.00 | 600.00 | 2217.00 |
| WILLIAM | JONES | DESIGNER | M | 18270.00 | 400.00 | 1462.00 |

Figure 60. Input data for the Correlation transformer

Specify these settings:



Correlation Transformer

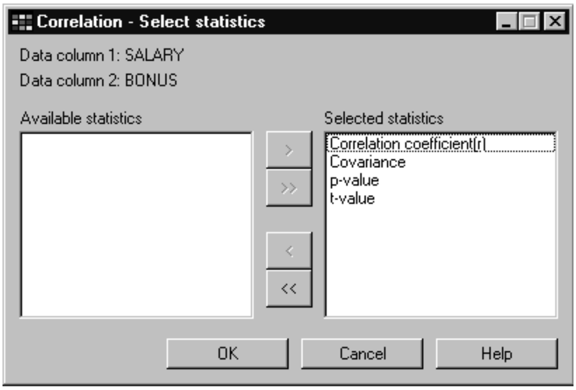
Available columns: EMPNO, FIRSTNME, LASTNAME, JOB, SALARY, BONUS, COMM

Grouping columns: SEX

| | Column 1 | Column 2 | Statistics |
|---|---|---|---|
| 1 | SALARY | BONUS | |

Figure 61. Settings for the Correlation transformer

Specify these statistics settings:



Correlation - Select statistics

Data column 1: SALARY

Data column 2: BONUS

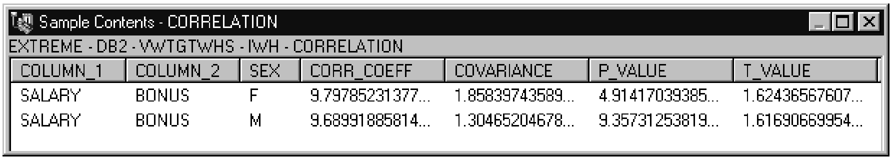
Available statistics

Selected statistics

- Correlation coefficient(r)
- Covariance
- p-value
- t-value

*Figure 62. Statistics settings for the Correlation transformer*

The Correlation transformer produces the following output data:



Sample Contents - CORRELATION

EXTREME - DB2 - VWTGTWHS - IWH - CORRELATION

| COLUMN_1 | COLUMN_2 | SEX | CORR_COEFF | COVARIANCE | P_VALUE | T_VALUE |
|---|---|---|---|---|---|---|
| SALARY | BONUS | F | 9.79785231377... | 1.85839743589... | 4.91417039385... | 1.62436567607... |
| SALARY | BONUS | M | 9.68991885814... | 1.30465204678... | 9.35731253819... | 1.61690669954... |

*Figure 63. Output data for the Correlation transformer*

### 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

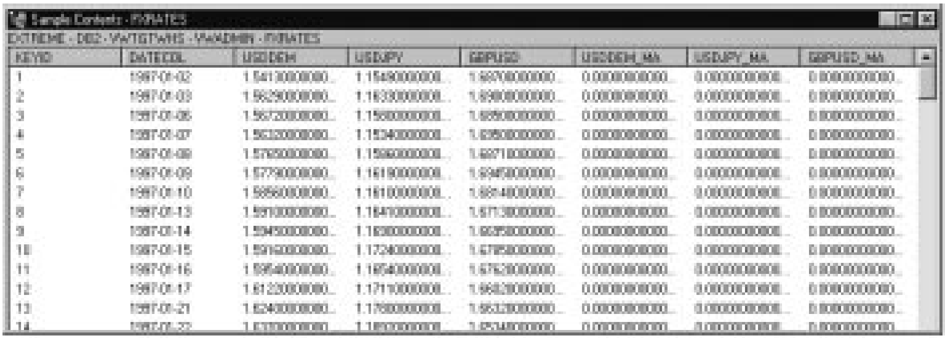
An exponential moving average is also known as an exponentially smoothed moving average.

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

For this transformer, the source and target table are the same. The transformer adds data to the source table to create the target table. The source table must have a primary key.

For example, suppose that you have this input data:



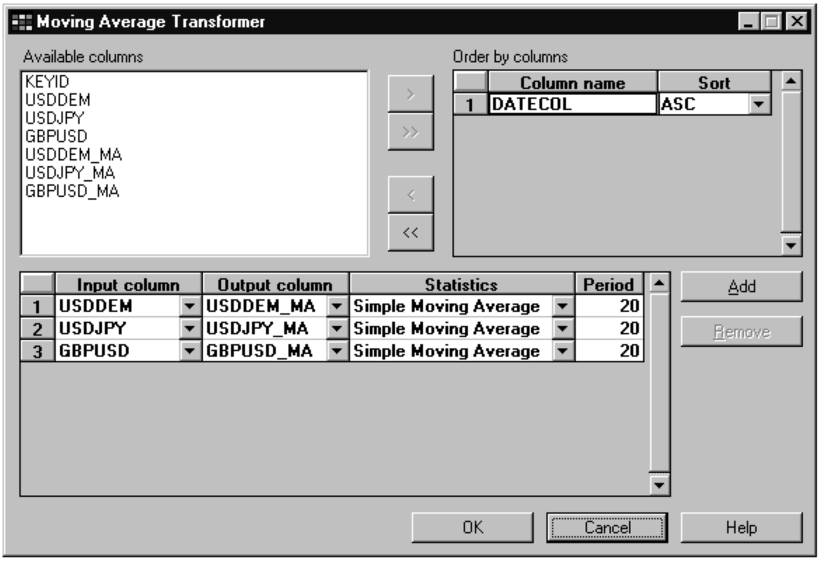
Sample Contents - FXRATES

EXTREME - DB2 - VWTGTWHS - VWADMIN - FXRATES

| KEYID | DATECOL | USDDEM | USDJPY | GBPUSD | USDDEM_MA | USDJPY_MA | GBPUSD_MA |
|---|---|---|---|---|---|---|---|
| 1 | 1997-01-02 | 1.5413000000... | 1.1549000000... | 1.6970000000... | 0.0000000000... | 0.0000000000... | 0.0000000000... |
| 2 | 1997-01-03 | 1.5629000000... | 1.1633000000... | 1.6980000000... | 0.0000000000... | 0.0000000000... | 0.0000000000... |
| 3 | 1997-01-06 | 1.5672000000... | 1.1560000000... | 1.6890000000... | 0.0000000000... | 0.0000000000... | 0.0000000000... |
| 4 | 1997-01-07 | 1.5632000000... | 1.1534000000... | 1.6950000000... | 0.0000000000... | 0.0000000000... | 0.0000000000... |
| 5 | 1997-01-08 | 1.5765000000... | 1.1566000000... | 1.6971000000... | 0.0000000000... | 0.0000000000... | 0.0000000000... |
| 6 | 1997-01-09 | 1.5779000000... | 1.1619000000... | 1.6945000000... | 0.0000000000... | 0.0000000000... | 0.0000000000... |
| 7 | 1997-01-10 | 1.5856000000... | 1.1610000000... | 1.6814000000... | 0.0000000000... | 0.0000000000... | 0.0000000000... |
| 8 | 1997-01-13 | 1.5910000000... | 1.1641000000... | 1.6713000000... | 0.0000000000... | 0.0000000000... | 0.0000000000... |
| 9 | 1997-01-14 | 1.5945000000... | 1.1630000000... | 1.6635000000... | 0.0000000000... | 0.0000000000... | 0.0000000000... |
| 10 | 1997-01-15 | 1.5916000000... | 1.1724000000... | 1.6785000000... | 0.0000000000... | 0.0000000000... | 0.0000000000... |
| 11 | 1997-01-16 | 1.5954000000... | 1.1654000000... | 1.6762000000... | 0.0000000000... | 0.0000000000... | 0.0000000000... |
| 12 | 1997-01-17 | 1.6122000000... | 1.1711000000... | 1.6682000000... | 0.0000000000... | 0.0000000000... | 0.0000000000... |
| 13 | 1997-01-21 | 1.6240000000... | 1.1760000000... | 1.6632000000... | 0.0000000000... | 0.0000000000... | 0.0000000000... |
| 14 | 1997-01-22 | 1.6330000000... | 1.1800000000... | 1.6534000000... | 0.0000000000... | 0.0000000000... | 0.0000000000... |

*Figure 64. Input data for the Moving Average transformer*

Specify these settings:


Moving Average Transformer

Available columns

KEYID
USDDEM
USDJPY
GBPUSD
USDDEM_MA
USDJPY_MA
GBPUSD_MA

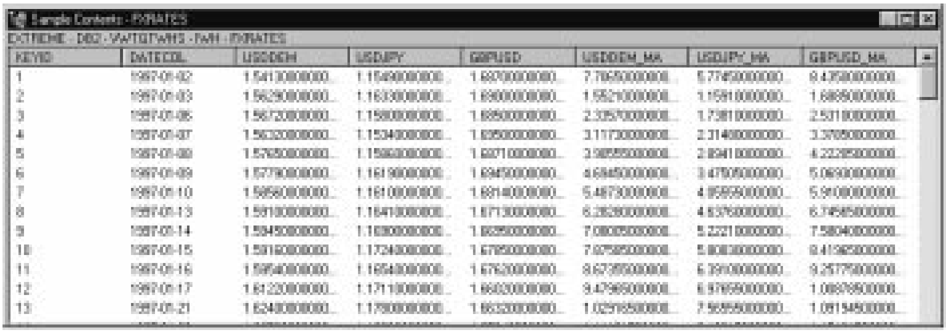
Order by columns

|   | Column name | Sort |
|---|---|---|
| 1 | DATECOL | ASC |

|   | Input column | Output column | Statistics | Period |
|---|---|---|---|---|
| 1 | USDDEM | USDDEM_MA | Simple Moving Average | 20 |
| 2 | USDJPY | USDJPY_MA | Simple Moving Average | 20 |
| 3 | GBPUSD | GBPUSD_MA | Simple Moving Average | 20 |

Figure 65. Settings for the Moving Average transformer

The Moving Average transformer produces the following output data:


Sample Contents - FXRATES

DCTREME - DB2 - VWTGTWHS - IWH - FXRATES

| KEYID | DATECOL | USDDEM | USDJPY | GBPUSD | USDDEM_MA | USDJPY_MA | GBPUSD_MA |
|---|---|---|---|---|---|---|---|
| 1 | 1997-01-02 | 1.5413000000... | 1.1549000000... | 1.6970000000... | 7.7065000000... | 5.7745000000... | 8.4850000000... |
| 2 | 1997-01-03 | 1.5629000000... | 1.1633000000... | 1.6900000000... | 1.5521000000... | 1.1591000000... | 1.6895000000... |
| 3 | 1997-01-06 | 1.5672000000... | 1.1580000000... | 1.6850000000... | 2.3357000000... | 1.7381000000... | 2.5311000000... |
| 4 | 1997-01-07 | 1.5632000000... | 1.1534000000... | 1.6950000000... | 3.1173000000... | 2.3148000000... | 3.3705000000... |
| 5 | 1997-01-08 | 1.5765000000... | 1.1566000000... | 1.6971000000... | 3.9055500000... | 2.8941000000... | 4.2228500000... |
| 6 | 1997-01-09 | 1.5779000000... | 1.1619000000... | 1.6945000000... | 4.6945000000... | 3.4750500000... | 5.0690000000... |
| 7 | 1997-01-10 | 1.5858000000... | 1.1610000000... | 1.6814000000... | 5.4873000000... | 4.0555500000... | 5.9100000000... |
| 8 | 1997-01-13 | 1.5910000000... | 1.1641000000... | 1.6713000000... | 6.2828000000... | 4.6376000000... | 6.7458500000... |
| 9 | 1997-01-14 | 1.5945000000... | 1.1690000000... | 1.6695000000... | 7.0800500000... | 5.2221000000... | 7.5804000000... |
| 10 | 1997-01-15 | 1.5916000000... | 1.1724000000... | 1.6795000000... | 7.8758500000... | 5.8083000000... | 8.4156500000... |
| 11 | 1997-01-16 | 1.5954000000... | 1.1654000000... | 1.6762000000... | 8.6735500000... | 6.3910000000... | 9.2577500000... |
| 12 | 1997-01-17 | 1.6122000000... | 1.1711000000... | 1.6603000000... | 9.4796500000... | 6.9765500000... | 1.0087850000... |
| 13 | 1997-01-21 | 1.6240000000... | 1.1790000000... | 1.6632000000... | 1.0291650000... | 7.5655500000... | 1.0815450000... |

Figure 66. Output data for the Moving Average transformer

## Regression transformer

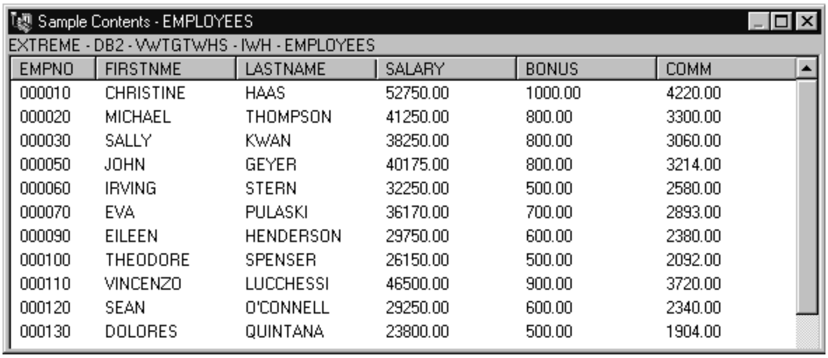
Use the Regression transformer to identify the relationships between a dependent variable and one or more independent variables, and to show how closely they are correlated. This transformer performs a backward, full-model regression.

You can use this transformer to:

- Show the effect of a change in pricing on demand for a product
- Show the effect of location on effectiveness for advertising
- Show how closely two seemingly random sets of data are related

This transformer creates two additional output tables. You can specify the names of these tables on the Program page of the Business View notebook.

For example, suppose that you have this input data:



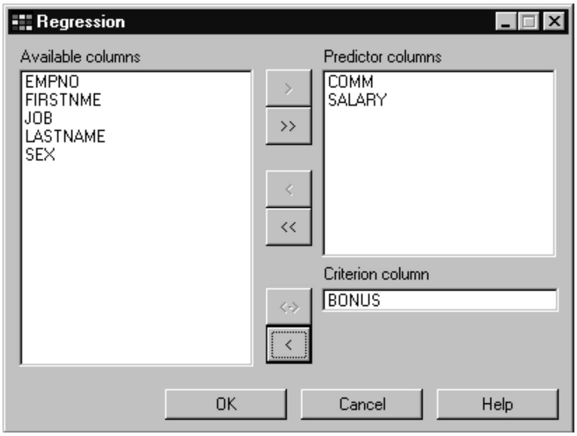
Sample Contents - EMPLOYEES

EXTREME - DB2 - VWTGTWHS - IWH - EMPLOYEES

| EMPNO | FIRSTNME | LASTNAME | SALARY | BONUS | COMM |
|---|---|---|---|---|---|
| 000010 | CHRISTINE | HAAS | 52750.00 | 1000.00 | 4220.00 |
| 000020 | MICHAEL | THOMPSON | 41250.00 | 800.00 | 3300.00 |
| 000030 | SALLY | KWAN | 38250.00 | 800.00 | 3060.00 |
| 000050 | JOHN | GEYER | 40175.00 | 800.00 | 3214.00 |
| 000060 | IRVING | STERN | 32250.00 | 500.00 | 2580.00 |
| 000070 | EVA | PULASKI | 36170.00 | 700.00 | 2893.00 |
| 000090 | EILEEN | HENDERSON | 29750.00 | 600.00 | 2380.00 |
| 000100 | THEODORE | SPENSER | 26150.00 | 500.00 | 2092.00 |
| 000110 | VINCENZO | LUCCHESSI | 46500.00 | 900.00 | 3720.00 |
| 000120 | SEAN | O'CONNELL | 29250.00 | 600.00 | 2340.00 |
| 000130 | DOLORES | QUINTANA | 23800.00 | 500.00 | 1904.00 |

*Figure 67. Input data for the Regression transformer*

Specify these settings in the Transformer window:



Regression

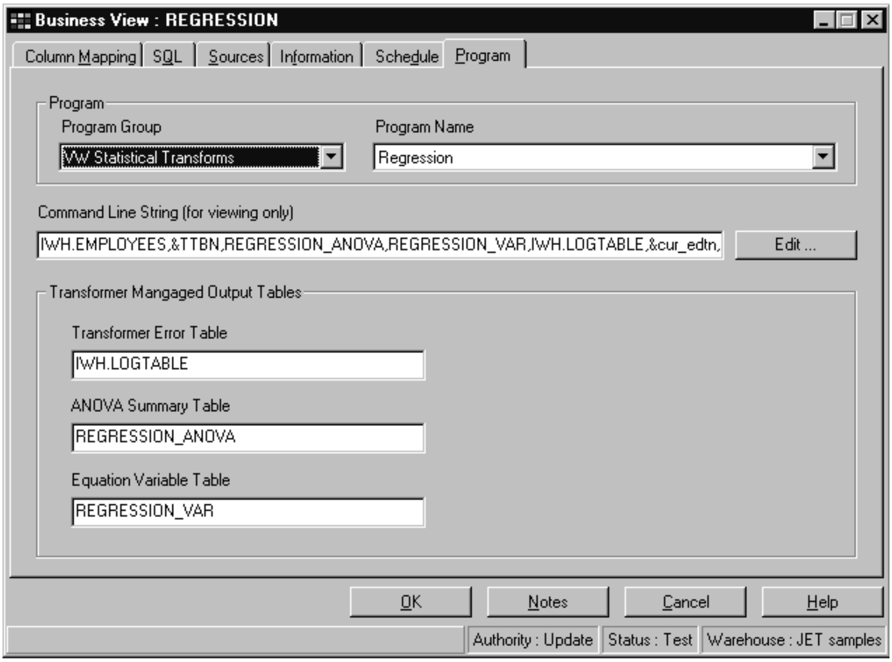
Available columns: EMPNO, FIRSTNME, JOB, LASTNAME, SEX

Predictor columns: COMM, SALARY

Criterion column: BONUS

*Figure 68. Settings for the Regression transformer*

Specify these settings on the Program page of the Business View notebook:



Business View : REGRESSION

Program

Program Group: VW Statistical Transforms

Program Name: Regression

Command Line String (for viewing only): IWH.EMPLOYEES,&TTBN,REGRESSION_ANOVA,REGRESSION_VAR,IWH.LOGTABLE,&cur_edtn,

Transformer Mangaged Output Tables

Transformer Error Table: IWH.LOGTABLE

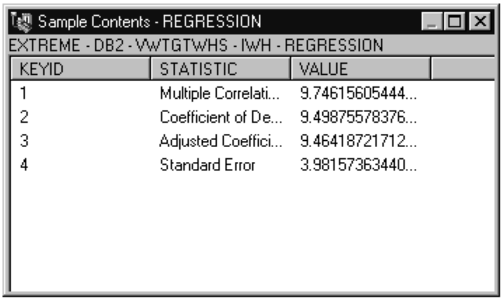
ANOVA Summary Table: REGRESSION_ANOVA

Equation Variable Table: REGRESSION_VAR

Authority : Update | Status : Test | Warehouse : JET samples

Figure 69. Settings for the Regression transformer output tables

The Regression transformer produces the following output data:



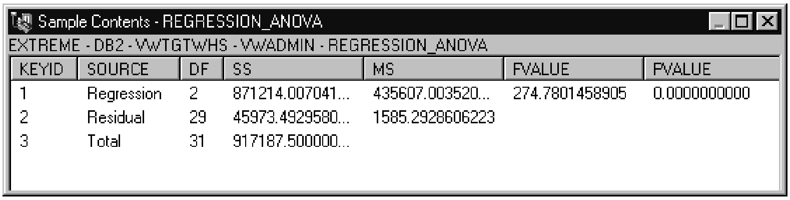
Sample Contents - REGRESSION

EXTREME - DB2 - VWTGTWHS - IWH - REGRESSION

| KEYID | STATISTIC | VALUE |
|---|---|---|
| 1 | Multiple Correlati... | 9.74615605444... |
| 2 | Coefficient of De... | 9.49875578376... |
| 3 | Adjusted Coeffici... | 9.46418721712... |
| 4 | Standard Error | 3.98157363440... |

Figure 70. Output data for the Regression transformer

The ANOVA table for this transformer looks like this:

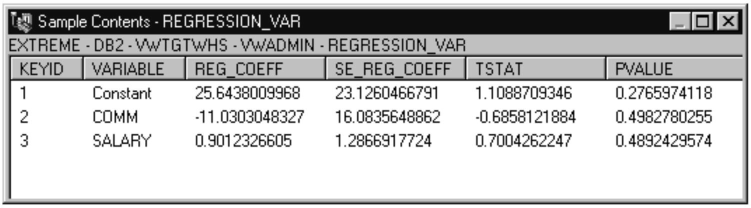


EXTREME - DB2 - VWTGTWHS - VWADMIN - REGRESSION_ANOVA

| KEYID | SOURCE | DF | SS | MS | FVALUE | PVALUE |
|---|---|---|---|---|---|---|
| 1 | Regression | 2 | 871214.007041... | 435607.003520... | 274.7801458905 | 0.0000000000 |
| 2 | Residual | 29 | 45973.4929580... | 1585.2928606223 | | |
| 3 | Total | 31 | 917187.500000... | | | |

*Figure 71. ANOVA output table for the Regression transformer*

The variance table looks like this:



EXTREME - DB2 - VWTGTWHS - VWADMIN - REGRESSION_VAR

| KEYID | VARIABLE | REG_COEFF | SE_REG_COEFF | TSTAT | PVALUE |
|---|---|---|---|---|---|
| 1 | Constant | 25.6438009968 | 23.1260466791 | 1.1088709346 | 0.2765974118 |
| 2 | COMM | -11.0303048327 | 16.0835648862 | -0.6858121884 | 0.4982780255 |
| 3 | SALARY | 0.9012326605 | 1.2866917724 | 0.7004262247 | 0.4892429574 |

*Figure 72. Variance table for the Regression transformer*

## User-defined function: Format Data and Time transformer

Use the Format Data and Time transformer to change the format of a date field in your source table that your business view is copying to the default target table. For example, you can use this transformer to make a date field Year 2000 ready. You can run this transformer with any other transformer or Visual Warehouse program.

The Format Data and Time 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`.

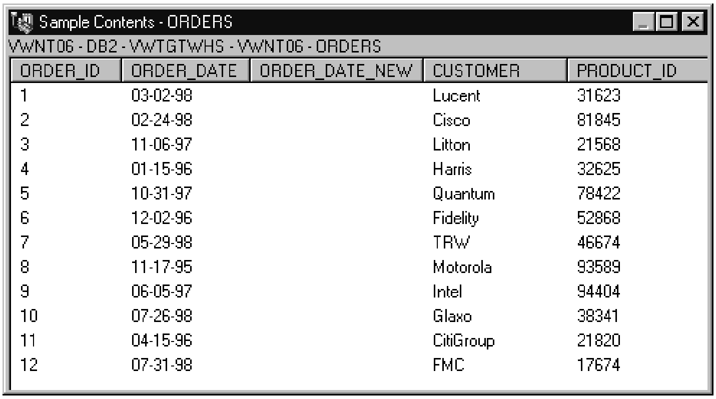
To use the Format Data and Time transformer:

1. Create a business view in a warehouse where the transformers are installed.
2. Select a source table.

3. Click **OK**.
4. On the Column Mapping page of the Business View notebook, click **...** (the column definition assistance button) for the output column you want to format.
5. In the Column Definition window, double-click **Format Date** in the **Functions** list.

   The Format Data and Time transformer window opens.

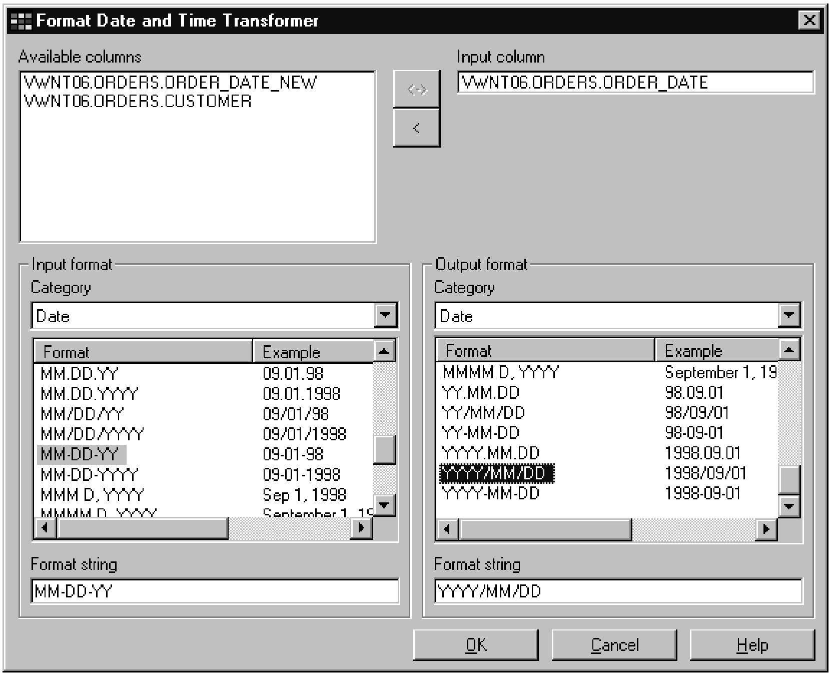
For example, suppose that you have this input data:



| ORDER_ID | ORDER_DATE | ORDER_DATE_NEW | CUSTOMER | PRODUCT_ID |
|---|---|---|---|---|
| 1 | 03-02-98 | | Lucent | 31623 |
| 2 | 02-24-98 | | Cisco | 81845 |
| 3 | 11-06-97 | | Litton | 21568 |
| 4 | 01-15-96 | | Harris | 32625 |
| 5 | 10-31-97 | | Quantum | 78422 |
| 6 | 12-02-96 | | Fidelity | 52868 |
| 7 | 05-29-98 | | TRW | 46674 |
| 8 | 11-17-95 | | Motorola | 93589 |
| 9 | 06-05-97 | | Intel | 94404 |
| 10 | 07-26-98 | | Glaxo | 38341 |
| 11 | 04-15-96 | | CitiGroup | 21820 |
| 12 | 07-31-98 | | FMC | 17674 |

*Figure 73. Input data for the Format Data and Time transformer*

Specify these settings:



Available columns

VWNT06.ORDERS.ORDER_DATE_NEW
VWNT06.ORDERS.CUSTOMER

Input column: VWNT06.ORDERS.ORDER_DATE

Input format

Category: Date

| Format | Example |
|---|---|
| MM.DD.YY | 09.01.98 |
| MM.DD.YYYY | 09.01.1998 |
| MM/DD/YY | 09/01/98 |
| MM/DD/YYYY | 09/01/1998 |
| MM-DD-YY | 09-01-98 |
| MM-DD-YYYY | 09-01-1998 |
| MMM D, YYYY | Sep 1, 1998 |
| MMMM D, YYYY | September 1, 19 |

Format string: MM-DD-YY

Output format

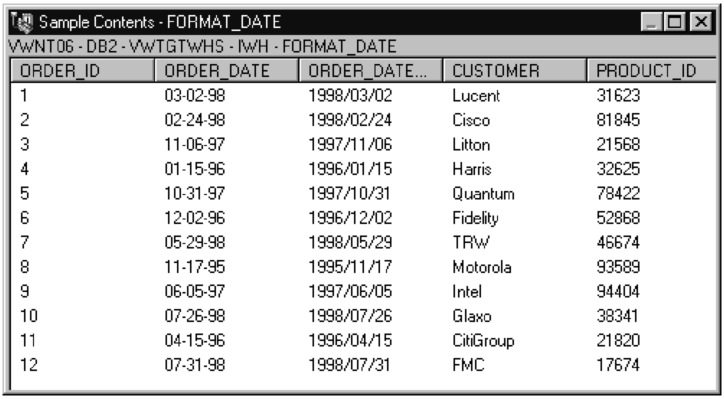
Category: Date

| Format | Example |
|---|---|
| MMMM D, YYYY | September 1, 19 |
| YY.MM.DD | 98.09.01 |
| YY/MM/DD | 98/09/01 |
| YY-MM-DD | 98-09-01 |
| YYYY.MM.DD | 1998.09.01 |
| YYYY/MM/DD | 1998/09/01 |
| YYYY-MM-DD | 1998-09-01 |

Format string: YYYY/MM/DD

Figure 74. Settings for the Format Data and Time transformer

The Format Data and Time transformer produces the following output data:


Sample Contents - FORMAT_DATE

VWNT06 - DB2 - VWTGTWHS - IWH - FORMAT_DATE

| ORDER_ID | ORDER_DATE | ORDER_DATE... | CUSTOMER | PRODUCT_ID |
|---|---|---|---|---|
| 1 | 03-02-98 | 1998/03/02 | Lucent | 31623 |
| 2 | 02-24-98 | 1998/02/24 | Cisco | 81845 |
| 3 | 11-06-97 | 1997/11/06 | Litton | 21568 |
| 4 | 01-15-96 | 1996/01/15 | Harris | 32625 |
| 5 | 10-31-97 | 1997/10/31 | Quantum | 78422 |
| 6 | 12-02-96 | 1996/12/02 | Fidelity | 52868 |
| 7 | 05-29-98 | 1998/05/29 | TRW | 46674 |
| 8 | 11-17-95 | 1995/11/17 | Motorola | 93589 |
| 9 | 06-05-97 | 1997/06/05 | Intel | 94404 |
| 10 | 07-26-98 | 1998/07/26 | Glaxo | 38341 |
| 11 | 04-15-96 | 1996/04/15 | CitiGroup | 21820 |
| 12 | 07-31-98 | 1998/07/31 | FMC | 17674 |

Figure 75. Output data for the Format Data and Time transformer

## Replicating data with Data Propagator Relational

Instead of using Visual Warehouse to write source data to the warehouse database, you can use Data Propagator™ Relational (DPropR) with Visual Warehouse to replicate data. DPropR uses staging tables to replicate data. By using staging tables, you can create a three-tiered warehousing environment more efficiently. Using staging tables also reduces the load on your network.

# Chapter 7. Maintaining Visual Warehouse

This chapter explains concepts and procedures that are associated with maintaining Visual Warehouse. This chapter covers the following topics:

- "Backing up Visual Warehouse" on page 131
- "Exporting and importing Visual Warehouse metadata" on page 132
- "Initializing a second Visual Warehouse control database" on page 139
- "Configuring Visual Warehouse" on page 141

## Backing up Visual Warehouse

Back up your Visual Warehouse control database, as well as all your warehouse databases, at regular intervals.

To back up the control database, use the standard procedures for DB2 backup and recovery.

You need to shut down Visual Warehouse before backing up the control database. To shut down Visual Warehouse:

1. Click **Settings —> Control Panel —> Services**. The Services window opens.
2. Select a Visual Warehouse service and click **Stop**.
3. Repeat this step for each Visual Warehouse service that is listed in the Services window.

   The Visual Warehouse logger and server are linked; if you stop one of them, the other also stops.
4. When you have stopped all Visual Warehouse services, click **Close**.

To back up your warehouse databases, use the backup and recovery procedures provided by your database management system (DBMS).

## Expanding your datamart

Visual Warehouse lets you expand your network configuration as your datamart grows. You can add new machines with new agents and administrative clients, or delete agents and administrative clients from your existing configuration.

You can also create a new control database. If you create a new control database, you must run the initialization process again. For more information, see “Initializing a second Visual Warehouse control database” on page 139.

To add administrative clients or agents to your Visual Warehouse configuration, follow the installation process for those components. To delete administrative clients or agents from your Visual Warehouse configuration, remove those components.

For more information about installing and removing Visual Warehouse components, see *Planning and Installing Visual Warehouse and DataGuide*.

## Exporting and importing Visual Warehouse metadata

You can use Visual Warehouse’s export and import capabilities to export and import object definitions within the Windows NT operating system. You can also export object definitions from Visual Warehouse directly into DataGuide. This provides end users and system analysts with a fast path to seeing what is in the datamart.

Export processes and import processes use a large amount of system resources. You may 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 Visual Warehouse to another. If you want to migrate Visual Warehouse, see *Planning and Installing Visual Warehouse and DataGuide*.

### Exporting metadata to another Visual Warehouse datamart

When you export metadata to a tag language file, Visual Warehouse 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 Visual Warehouse system.

You can export object definitions for information resources, warehouses, subjects, business views, and Visual Warehouse programs to a tag language file.

Any objects that are related to the object being exported, such as agent sites and Visual Warehouse program groups, are also exported. Some security groups might not be exported because a security group associated with exported sources or targets must be related to a business view that uses those sources or targets. If the security groups that you want your sources and targets to be related to are not imported, you need to add the security groups to the new warehouse.

### Information resources

You can export any number of information resource definitions for source databases and files. When you export an information resource definition, Visual Warehouse exports all of the tables, files, or segments, and columns or fields that are associated with it.

### Warehouses and subjects

You can export any number of warehouse definitions or subject definitions. When you export a warehouse or subject definition, you can export any or all of the business views within the warehouse or subject.

Before you export business view definitions with a warehouse or a subject definition:

- If the information resources used by the business views already exist in the target datamart, do not export them with the business views.

  If the information resource definitions do not already exist in the target datamart, export them before you export the business views that use them. Otherwise, you might export the same data source definition for more than one business view.
- To increase the performance of export and import, or to decrease the size of tag files, or both:
  - Export your information resource definitions.
  - Change the export settings so that Visual Warehouse will export the business view definitions without their dependent source properties. To do this, clear the **Export Dependent Source Properties** check box in the Settings window.

    If these definitions already exist in the target datamart, do not export the data source definitions that are used by the business view definitions you plan to export. Instead, change the export settings so that Visual Warehouse will export the business view definitions without their dependent source properties.

### Visual Warehouse programs

You can export Visual Warehouse program definitions in two ways:

- Individually, when they are attached to business view definitions that you select to export

- Collectively, regardless of whether they are attached to any business view definitions

Visual Warehouse automatically exports program definitions when you export the business views definitions to which they are attached. You cannot export a business view definition that uses a program definition without exporting the program definition.

To export all of your Visual Warehouse program definitions collectively:

1. Click **Settings** in the Export Warehouse window.

   The Export to Tag File: Settings window opens.
2. Select the **Export All Program Definitions** check box.
3. Click **OK**.

When you export Visual Warehouse program definitions collectively, you do not need to export any information resources, warehouses, or subjects with them.

### Exporting the metadata to a tag language file

To export the metadata to a tag language file:

1. Log on to the Visual Warehouse desktop.
2. Click **File –> Export to Tag File**.
3. Type the name of the tag language file.
4. Select the Visual Warehouse objects for which you want to export metadata.

For more information about exporting metadata to a tag language file, see the online help.

## Exporting to DataGuide

You can export the metadata that defines information resources, warehouses, subjects, and business views from Visual Warehouse to DataGuide. When you export to DataGuide, Visual Warehouse finds the subjects, warehouses, and business views that you selected, and imports their metadata directly into DataGuide. If the objects you selected have related source databases or files, the metadata for the source databases or files is exported as well. The business views are then registered in the DataGuide information catalog that you specified when you exported.

To successfully export metadata to DataGuide, you must observe the following rules:

**Warehouse metadata**

When you export warehouse metadata, you must select business view metadata to export with the warehouse. You can choose to export the mapping of the source tables to the target table at the table level or at the column level.

**Subject metadata**
When you export subject metadata, you can select business view metadata to export with the subject. You can choose to export the mapping of the source tables to the target table at the table level or at the column level. However, you do not have to export business view metadata with the subject.

**Business view metadata**
You cannot export business view metadata by itself. To export business view metadata, you must export the warehouse or subject metadata to which the business view metadata belongs.

When a business view is exported to DataGuide, any source tables, target table, Visual Warehouse program and Visual Warehouse program output that the business view uses is exported as well.

Table 12 provides the mapping between Visual Warehouse and DataGuide object types. Visual Warehouse uses this mapping when you export metadata to DataGuide.

*Table 12. Mapping between Visual Warehouse and DataGuide object types*

| Visual Warehouse object type | DataGuide object type |
|---|---|
| Business view | Transformation |
| Column | Columns or fields |
| Information resource (data source or warehouse) | Databases, files, or IMS database definitions |
| Subject | Business subject areas |
| Table | IMS segments, or relational tables and views |

## Exporting the metadata to DataGuide

To export the metadata to DataGuide:

1. Log on to the Visual Warehouse desktop.
2. Click **File –> Export to DataGuide**.
3. Type the name of the trace file.
4. Select the Visual Warehouse objects for which you want to export metadata.

For more information on exporting metadata to DataGuide, see the online help.

## Synchronizing metadata between Visual Warehouse and DataGuide

After Visual Warehouse registers the metadata for an imported object in the DataGuide information catalog, you can synchronize the metadata between Visual Warehouse and DataGuide. This process is called *metadata synchronization*.

Metadata synchronization occurs when a business view that uses one of the synchronization Visual Warehouse programs runs. To run the synchronization programs, you must install DataGuide Administrator on the same workstation as the Visual Warehouse administrative client and the Visual Warehouse Windows NT agent.

When you use metadata synchronization, the metadata for an object that is registered in the DataGuide information catalog is automatically updated each time you change the object in Visual Warehouse. The metadata for business views is updated when you promote the business view to production status. Metadata is not updated in DataGuide under the following situations:

- When an object is created in Visual Warehouse.

  To confirm that the object is to be available to end users, you must export the new object to DataGuide before it can be included in metadata synchronization.
- When the name of an object in Visual Warehouse that you previously exported to DataGuide is changing.

  Visual Warehouse uses the name of the object as a unique key, so if the name changes, Visual Warehouse treats the object as a new object.

When an object is deleted in Visual Warehouse, information about the deleted object is stored in the Visual Warehouse control database. During metadata synchronization, Visual Warehouse propagates these deletions to the DataGuide information catalog before importing other changes into the information catalog. When metadata synchronization has completed successfully, DataGuide removes the entries from the history table. Because DataGuide removes the entries from the history table after synchronization, Visual Warehouse can propagate deletes to only one information catalog. If you need to make the deletions to a second information catalog, you must make them manually.

If you do not intend to use metadata synchronization, you can drop the history table, FLG.HISTORY, from the control database.

If you change the name of an object in Visual Warehouse that you previously exported to DataGuide, you must export the object again to update the information in DataGuide. If the object you previously exported is a business view, you can export the business view with the same warehouse or subject that you exported previously. The business views that exist in the previous version of the warehouse or subject in the DataGuide information catalog are not overwritten (unless you choose to export new business views that have the same names). Both the business view with the changed name and the previous version of this business view now exist in the DataGuide information catalog.

If you have added information to the information catalog, you can keep metadata synchronization from overwriting the information you added. For more information, see the online help for the metadata synchronization windows.

For information on the objects you can export to DataGuide, see “Exporting to DataGuide” on page 134.

For information on metadata synchronization, see “Exchanging metadata with other products” in *Managing DataGuide*.

## Pre-import tasks and considerations

Before you can import a tag language file, you need to initialize the control database. You also need to ensure that the target database does not already have a warehouse with the same name. Further, you must ensure that you import object definitions in the correct order.

### Initializing the control database

If you are using the import utility to establish a new Visual Warehouse datamart, you must initialize a new control database in the target system. After you complete this task, you can import as many tag language files as you like.

For more information on how to initialize a new control database, see “Initializing a second Visual Warehouse control database” on page 139.

### Naming considerations

If you import an information resource or warehouse that has the same name as an information resource or warehouse that already exists in the target database, it will replace the information resource or warehouse that currently exists in the target database.

If you import a subject that has the same name as a subject that already exists in the target database, it will replace the subject that currently exists in the target database.

If you attempt to import a business view into a target datamart that has the same name as a development-level business view that already exists there, the business view that you are importing will replace the business view that currently exists in the target datamart.

If you attempt to import a business view into a target datamart that has the same name as a test or production-level business view that currently exists there, an error will occur.

### Ordering objects

If information resources and business views are in separate tag language files, always import source definitions before you import business view definitions.

### Importing the metadata into Visual Warehouse

To import the metadata into a Visual Warehouse:

1. Log on to the Visual Warehouse desktop.
2. Click **File –> Import from Tag File**.
3. Type the name of the tag language file.

For more information about importing metadata into Visual Warehouse, see the online help.

## Post-import considerations

Before you promote business views that you imported, enter the passwords again for the agent site and warehouse database that the business view uses and the agent site that the warehouse uses.

Before you run business views that you imported, enter the passwords again for any source databases that the business view uses and the agent sites used by the related information resources.

If the security groups that you want your sources and targets to be related to are not imported, you need to add the security groups to the new warehouse.

For more information on post-import considerations, see the online help for the Import Warehouse window.

## Re-importing tag language files

You can import most metadata in tag language files more than once without errors. However, if the tag language file includes metadata for primary and foreign keys, you must delete the metadata for the keys before importing the file again. You do not need to delete the metadata for other objects before importing the metadata again. You must demote business views to development status before importing them again, or an error will occur.

If you have changed the names or order of columns in a table or file, such as redefining a source table or a target table, then delete the source table or file, or target table before importing the table or file again. If you do not delete the table, verify after importing that the column or field information is as you expected.

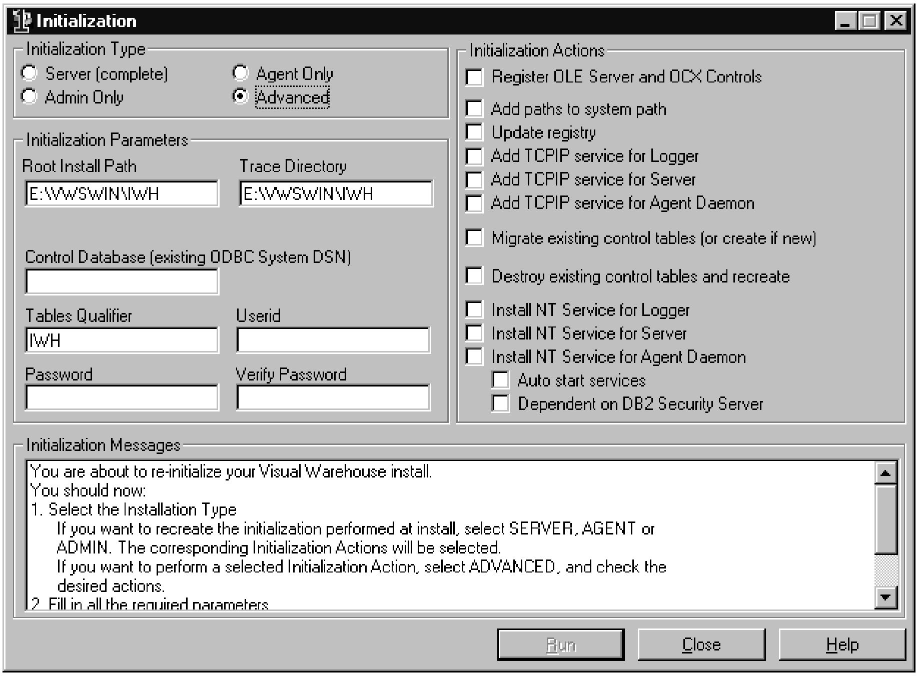
## Initializing a second Visual Warehouse control database

When you install Visual Warehouse, the control database that you specify during installation is initialized. *Initialization* is the process in which Visual Warehouse creates the control tables required to store Visual Warehouse metadata. If you have more than one control database, you can use the Visual Warehouse Initialization window to initialize the second control database. However, only one control database can be active at a time.

To initialize a second Visual Warehouse control database:

1. Click **Start –> Programs –> Visual Warehouse –> Visual Warehouse –> Visual Warehouse Initialization**.

   The Initialization window opens.



Initialization

Initialization Type

○ Server (complete) ○ Agent Only
○ Admin Only ● Advanced

Initialization Parameters

Root Install Path: E:\VWSWIN\IWH

Trace Directory: E:\VWSWIN\IWH

Control Database (existing ODBC System DSN):

Tables Qualifier: IWH

Userid:

Password:

Verify Password:

Initialization Actions

□ Register OLE Server and OCX Controls
□ Add paths to system path
□ Update registry
□ Add TCPIP service for Logger
□ Add TCPIP service for Server
□ Add TCPIP service for Agent Daemon
□ Migrate existing control tables (or create if new)
□ Destroy existing control tables and recreate
□ Install NT Service for Logger
□ Install NT Service for Server
□ Install NT Service for Agent Daemon
□ Auto start services
□ Dependent on DB2 Security Server

Initialization Messages

```
You are about to re-initialize your Visual Warehouse install.
You should now:
1. Select the Installation Type
     If you want to recreate the initialization performed at install, select SERVER, AGENT or
     ADMIN. The corresponding Initialization Actions will be selected.
     If you want to perform a selected Initialization Action, select ADVANCED, and check the
     desired actions.
2. Fill in all the required parameters
```

*Figure 76. Initialization window for a custom initialization*

2. Click **Advanced** in the Initialization Type area.
3. In the **Initialization Actions** area, select the applicable check boxes for your task.

| To do the following task: | Select this check box: |
|---|---|
| Migrate a control database from Visual Warehouse Version 3.1 to Version 5.2 | Select only the **Migrate existing control tables (or create if new)** check box. |
| Initialize a new control database. | Select only the **Migrate existing control tables (or create if new)** check box. |
| Update the Windows NT registry and start the Visual Warehouse services to use a new control database. | Select the **Update registry**, **Install NT Service for Logger**, and **Install NT Service for Server** check boxes. To automate starting Visual Warehouse, select the **Auto Start Services** and **Dependent on DB2 Security Server** check boxes. |

For more information about the initialization actions, click **Help** on the Initialization window.

4. Click **Run** to run the initialization process. Click **Close** when the initialization processes finishes running.

## Configuring Visual Warehouse

You can use the configuration tool to change global settings for your Visual Warehouse installation. For example, you can use the configuration tool to specify the default behavior of a processing business view when the agent finds no rows in the source table or file. You can override this global setting in a particular business view.

You can use the configuration tool only if the Visual Warehouse server is installed on the workstation (as well as the administrative client).

Use the configuration tool to change the following settings:

- The trace level of the components
- The server schedule mode
- The server host name
- The trace directory

### Using the configuration tool

To change the Visual Warehouse configuration:

1. Select **Start –> Programs –> Visual Warehouse Solution –> Visual Warehouse –> Visual Warehouse –> Visual Warehouse Configuration**.

   A logon window opens.
2. Type:
   a. The host name of the server.
   b. The name of the control database.
   c. The table qualifier for the control database.
      The default is IWH.
   d. The user ID and password for the control database.
3. Click **OK**.

   The Configuration notebook opens.
4. Change the appropriate values on each page of the notebook.

# Appendix A. Troubleshooting

This appendix outlines problems you might experience while using Visual Warehouse. Problems are listed by type with specific categories, such as installation issues. Each problem outline contains a solution that you can use to resolve the problem.

## Prerequisite or related product issues

This section details errors that occur with products that are installed and configured with Visual Warehouse.

### Cannot find the System DSN button on the ODBC Administrator window

You double-clicked the ODBC icon in the Windows NT Control Panel. The System DSN button either does not work or is not displayed.

#### Cause

You installed an ODBC application that does not have a System DSN button and overwrites the old ODBC administrator.

#### Resolution

Copy the file ODBCAD32.EXE from the ODBC32 subdirectory of the Visual Warehouse installation directory to your Windows NT system directory.

### A DataJoiner Version 2 CREATE SERVER MAPPING statement for an Oracle data source fails

DataJoiner Version 2 CREATE SERVER MAPPING SQL statements fail on a workstation that has Visual Warehouse installed.

You might have received the message `DB2SYSC.EXE -- Ordinal not found` or one that implies that the SERVER MAPPING failed because it could not load the SQL*NET data access module.

### Cause

The CREATE SERVER MAPPING statement fails because DataJoiner accesses the ORA73.DLL provided with Visual Warehouse instead of the ORA73.DLL in the ORANT\BIN directory. This condition occurs because DataJoiner looks for the Oracle DLL via the Path system variable. The system path is updated during a Visual Warehouse installation so that it includes a directory containing a ORA73.DLL provided by Visual Warehouse.

### Resolution

Update the Path system variable. Ensure that the directory containing the Oracle ORA73.DLL (in ORANT\BIN) is specified first in the path (ahead of the VWSWIN\IWH\ODBC32 directory).

## Installation issues

This section details errors that can occur when you install or initialize the Visual Warehouse product.

### Windows NT cannot install services for Visual Warehouse components

When you install Visual Warehouse, the Visual Warehouse logger and Visual Warehouse server files are not installed in the Windows NT Services folder.

#### Cause

You are installing from a Windows NT user member user ID.

#### Resolution

Log off of Windows NT and log on with a user ID that has administrative authority.

### Initialization fails at ″Add install path to NT service path″

Visual Warehouse initialization stops at the `Add install path to NT service path` message.

#### Cause

You are installing from a Windows NT user member user ID.

#### Resolution

Log off of Windows NT and log on with a user ID that has administrative authority.

### Initialization fails with a ″3043 Disk or Network Error″

Visual Warehouse initialization cannot write to the TEMP or TMP directories.

### Cause

Either the TEMP or TMP directory (or both) is not defined.

### Resolution

You must have the following valid statements in the **User Environment Variables** list in your System folder.

```
TEMP = drive:\TEMP
TMP  = drive:\TEMP
```

where `drive` is any drive with available space for temporary files.

The System folder is located in the Control Panel folder. The TEMP directory must exist in the **User Environment Variables** list in the System folder.

To direct the Visual Warehouse initialization process to write to the TEMP or TMP directories:

1. Create the TEMP directory to which the statements refer.
2. Ensure that the statements are in your User Environment Variables folder.
3. Restart your machine.
4. Initialize Visual Warehouse again.

## Visual Warehouse files disappear after you install the product again

You installed Visual Warehouse again, but you cannot locate the product files.

### Cause

You installed the product without restarting after the removal procedure.

### Resolution

You must restart your machine after you remove the product. See *Planning and Installing Visual Warehouse and DataGuide* for more information.

## Visual Warehouse will not start

You receive the error message, `SYS0005: Access denied`. As a result, you cannot access any Visual Warehouse services.

### Cause

You tried to access Visual Warehouse by using a different user ID than the one you used when you installed the product.

### Resolution

Log on to your NT workstation under the same user ID that you used when you installed Visual Warehouse.

## Initialization fails with an Unable to connect message

You receive the following message, `Unable to connect to Database: IWHDB01...` after a successful installation, even if you can connect to the database by using the DB2 Command Line Processor.

### Cause

There are several reasons why you might get this message:

- The database is not registered as an ODBC system DSN.
- You did not start the DB2 security server.
- You installed DB2 for Windows NT with a user ID that did not have administrator authority.

### Resolution

Try one of the following corresponding solutions:

- Register the database as an ODBC system DSN.
- Start the DB2 security server from Windows NT Services.
- Install DB2 for Windows NT again under an appropriate user ID.

## Initialization fails with a 9159: Win32 function RegOpenKeyEx failed message

You attempted to initialize Visual Warehouse after you installed Visual Warehouse from a Windows NT user ID that does not have Windows NT Administration authority.

### Cause

Visual Warehouse could not open a key in the Windows NT registry.

### Resolution

Install Visual Warehouse again from a Windows NT user ID that has NT Administrator authority.

## Unable to connect to a DB2 UDB database through TCP/IP

You are unable to access a DB2 source or target database.

### Cause

There are several reasons why you might have this problem:

- Your DB2 client is configured incorrectly.
- Your Visual Warehouse control database is not registered as an ODBC system DSN on your Visual Warehouse server and on your administrative client workstation.
- Your information source database and warehouse database are not registered as ODBC system DSNs on your agent site workstation.
- Your TCP/IP service is not started, or TCP/IP connectivity problems exist.

### Resolution

Try the following solutions:

- Verify that the workstation name is entered as `NNAME` in your DB2 client database manager configuration.
- Ensure that the node and database are cataloged correctly, as follows:

| **Verify that the DB2 client catalog:** | **Matches the host DB2 catalog:** |
|---|---|
| node | `NNAME` |
| host name | TCP/IP host name |
| service name | SVCENAME of the database manager host configuration |
| database name | database name |

- Ensure that the DB2 security service is started correctly and that the DB2 service in DB2 UDB is started correctly. Check the DB2DIAG.LOG file in your DB2 installed directory for any startup errors.
- Verify that your Visual Warehouse control database is registered as an ODBC system DSN on your Visual Warehouse server workstation and all your administrative client workstations.
- Verify that your information resource database and warehouse database are registered as ODBC system DSNs on your agent site workstations.
- Ensure that the TCP/IP service is started at both the client and host. To do this, check the Services application of the Control Panel program group on both workstations.
- Verify TCP/IP connectivity. Ensure that all cables are connected, that the TCP/IP service is started on both workstations, and that the LAN is functioning.

  If these configurations are correct, ping the server workstation from the client workstation command prompt.

  If the ping command succeeds, ensure that the server's DB2COMM environment variable is set to `TCPIP`.

If the ping command fails:

- Verify that the server and the client have the same entry for the server TCP/IP port number in the TCP/IP SERVICES file.
- Ensure that the entries for the DB2 server in the client SERVICES file and the server SERVICES file do not contain tab characters.
- Ensure that the DB2 server recognizes the port number. If not, update the DB2 server configuration:
  1. Enter the following command from the DB2 Command Line Processor:

     ```
     DB2> update dbm cfg using svcename your-svc
     ```

     where `your-svc` is the server's SERVICE name from the SERVICES file.
  2. Stop and restart DB2.
- If you do not have a domain name server, verify that the TCP/IP host name is entered in the client TCP/IP HOSTS file for the server workstation.
- If you have a domain name server, or an entry for the server in the TCP/IP HOSTS file, select the network icon in the Client NT workstation's Control Panel. Verify that the TCP/IP configuration is correct. If it is correct, click the **DNS** push button and verify that the host and domain names are correct.

## Unable to use TCP/IP to connect to AS/400 from Windows NT

You are trying to use TCP/IP to connect to the AS/400 from a DB2 Connect for Windows NT workstation.

### Cause

This configuration is not supported.

### Resolution

Try connecting with TCP/IP from a client workstation to the DB2 Connect for Windows NT workstation, and then to the host using APPC. For more information, run the following trace on the DB2 Connect for Windows NT workstation:

1. At a command prompt, enter:

   ```
   db2trc on -l 8000000
   ```
2. Reproduce the problem.
3. At a command prompt, enter:

   ```
   db2trc dmp db2.trc
   ```

4. To turn the trace off, at a command prompt, enter:

   ```
   db2trc off
   ```

## Product function issues

These problems can occur while you use Visual Warehouse.

### A Visual Warehouse component is unable to communicate with another component

The following message information is displayed with this error:

```
Return code = 6106
Message = 6117
Method VWSocket::Send
Secondary Code = 10061
System Message = 10061
```

#### Cause

A Visual Warehouse component was unable to communicate with another Visual Warehouse component. The most common occurrence of this error is when the server cannot communicate with the agent daemon because the agent daemon is not running.

#### Resolution

Open the Windows NT Control Panel and double-click the Services icon. Scroll down the list of services until you find **Visual Warehouse Server**, **Visual Warehouse Logger**, and **Visual Warehouse Agent**. If the services are stopped, start them. (The server and logger are linked so that both start when you start one of them.)

### A Visual Warehouse agent stops processing when promoting a business view

The business view promotion process stops, but you can still access other parts of Visual Warehouse.

#### Cause

The agent is in a lock-wait state on a target warehouse.

### Resolution

You need to free the agent workstation. To do this:

1. Open a DB2 the Command Line Processor window and enter:

   ```
   DB2 LIST APPLICATIONS
   ```

   The DB2 agent ID for the locked agent is listed.
2. Record the DB2 agent IDs for the database source and the target warehouse.
3. Enter:

   ```
   DB2 FORCE APPLICATIONS (XXX)
   ```

   where XXX is the DB2 agent ID for the database source.
4. Repeat the previous step for the target warehouse.

## Creation of an AIX agent process fails

There is no response from the AIX agent, or the AIX agent is unable to connect to a source or target database.

### Cause

There are several reasons why you might have this problem:

- The libdba2.a file might be missing from the LIBPATH.
- The agent might not be able to communicate with the server by its local name.
- The default port numbers that are assigned to the Visual Warehouse TCP/IP services might be in use by other services.

### Resolution

Try the following solutions:

- Verify that the libdba2.a file is in the LIBPATH. If it is not, copy the file to the LIBPATH. To find the libdba2.a file, enter `echo $LIBPATH` at a command prompt.
- Ensure that the agent can communicate with the server by its local name. Update the /etc/hosts file in the AIX host as follows:

  ```
  ipaddress localname entire.name
  ```

  where:

  `ipaddress` is the server IP address, `localname` is the local name of the server, and `entire.name` is the entire name for the server.

- Verify that the port numbers that the Visual Warehouse TCP/IP services uses in the Services file are not in use by other services. The default port numbers for the Visual Warehouse TCP/IP services are:
  - vwkernel 11000/tcp
  - vwd 11001/tcp
  - vwlogger 11002/tcp

  If other TCP/IP services currently use these port numbers, change the port numbers assigned to the Visual Warehouse TCP/IP services to port numbers that are not currently in use. Change the port numbers on both the agent site and the server workstation.
- Check the agent log for diagnostic information.

## Visual Warehouse stops processing if you run too many business views at the same time

Visual Warehouse stops processing; there are no status changes in the Operations Work In Progress window for currently running business view editions.

### Cause

Your target warehouse is locking up because too many agents are trying to update the control table at the same time.

### Resolution

Change the database configuration parameter LOCKS TIMEOUT. The default value is -1, which means that DB2 will wait forever. If you are experiencing locking problems, use the DB2 Control Center or the DB2 Command Line Processor to set the wait time to 30 seconds.

## Cannot import tables from AS/400

Visual Warehouse fails to access DB2 for OS/400 tables.

### Cause

There are some known problems that can occur when you use Visual Warehouse for Windows NT with AS/400. For example:

- Requests from any ODBC application for tables from DB2 for AS/400 can fail.
- The table filter might not work during table import.
- The table import performance might be slow, or Visual Warehouse might not appear to return any tables when importing.

### Resolution

Apply DB2 server fixpack US8125D or later.

## Prepare of SQL statement fails

### Cause

Not enough storage is available in the application heap to process the statement.

### Resolution

You need to increase the application heap size for the Visual Warehouse control database. See *Planning and Installing Visual Warehouse and DataGuide*.

## The Start/End Date/Time is incorrect in the Operations Work In Progress window

The date and time are incorrect in the **Scheduled** and **Completed** columns of the Operations Work In Progress window.

### Cause

Your Windows NT system clock is set incorrectly.

### Resolution

Open the Regional Settings application in the Control Panel and change the clock setting to the correct time and date. Include the correct time zone.

## DB2 reports an SQL0973N error when you open and close business views

The DB2 application heap and package cache default values can only handle a limited number of business views.

### Cause

Not enough storage space is allocated to the DB2 application heap and package cache. The default settings for a database are:

**applheapsz**
128

**pckcachesz**
36

### Resolution

Change the default settings for the DB2 application heap and package cache:

**applheapsz**
512

**pckcachesz**
128

## A business view that uses Microsoft Access as a source fails

You repeatedly try to run a business view that uses Microsoft Access. The business view fails every time.

### Cause

There are several reasons that the error might occur:

- The SELECT statement in your business view definition is using an unsupported function.

  For example, you are accessing data from a MEMO text field in an Access database and placing it in a LONG VARCHAR field in your business view. You need to use the Left() function, which is supported by Microsoft access, instead of the Substr() function.
- The import column definition failed because the driver is not capable of providing column definitions.

### Resolution

Try one of the following solutions:

- Check the SELECT statement in your business view definition to ensure that functions are supported by the Microsoft Access driver.

  For a mapping of the ANSI SQL data types that are supported by Microsoft Access, open the Microsoft Access **Help Topics** window and search for **Equivalent ANSI** on the **Find** tab. You will find a help topic called Equivalent ANSI SQL Data Types.
- The Microsoft Access ODBC driver does not support import of column definitions. You can manually define the column definition in the Business View notebook.

## Column is no longer visible in column grid

When you widen a column in a grid and then edit the column, the first column is no longer visible in the grid.

For example, when you widen and edit the **Parameter Text** column in the Parameters page of the Program notebook, the **Parameter Name** column is no longer visible in the grid.

### Cause

The combined width of the columns in the grid is larger than the area available to display the columns.

### Resolution

The missing column still exists; it just isn’t visible. Press the left or right arrow key to scroll to the missing column.

# Appendix B. Visual Warehouse logging and trace data

Visual Warehouse provides three levels of logging capability:

- Basic logging function
- Program trace
- Start error trace

To assist you in diagnosing problems, each of these logging levels contains a variety of features. This chapter describes the Visual Warehouse logging levels and features.

## The basic logging function

The basic logging function in Visual Warehouse captures all the events and errors that occur during run time and build time. This information is stored in the Visual 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 (business view processing) errors:

1. Open the Visual Warehouse desktop.
2. Click **Operations –>Work in Progress**.

   The Operations Work in Progress window opens.
3. Select the business view for which you want to view errors.
4. Click **Log**.

   The Log Viewer opens and displays the run-time errors for the selected business view.

To view build-time (table import, object creation, and business view promotion) errors:

1. Open the Visual Warehouse desktop.
2. Click **Operations –>Log**.

   The Log Viewer opens and displays the build-time errors for the datamart.

For detailed explanations of return codes and error messages that the log viewer displays, see *Messages and Reason Codes*.

The log holds records until a designated count limit is reached. When the count limit is reached, Visual Warehouse automatically deletes the log.

To change the count limit for the log, see the online help for the Configuration notebook.

**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 through the Windows NT Application Events, which is available through the Event Viewer application.

## Component trace data

The second level of logging capability is component trace data. The component trace data is inactive by default. Do not activate it unless IBM Software Support instructs you to do so.

The component trace data function writes information to text files. These files are located in the directory specified by the VWS_LOGGING environment variable. The default value of VWS_LOGGING is *x*:\vwswin\logging\ on Windows NT and OS/2, and /var/IWH on UNIX®. 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, Visual Warehouse writes the log files to the new directory you specified, but the value of VWS_LOGGING does not change.

The component trace data function provides information about the following components:

**Server** The component trace data function writes information to IWH2SERV.LOG in the directory specified in the VWS_LOGGING environment variable.

**Logger**
The component trace data function writes information to IWH2LOG.LOG in the directory specified in the VWS_LOGGING environment variable.

**Control**
The component trace data function writes information to IWH2EOLE.LOG in the directory specified in the VWS_LOGGING environment variable.

**Windows NT Agent**
The component trace data function writes trace information to AGNT*nnnn*.LOG and the agent’s environment settings to

AGNT*nnnn*.SET, in the directory specified in the VWS_LOGGING environment variable, where *nnnn* is the process ID.

For information about starting an agent trace, see the online help.

**AIX Agent**

The component trace data function writes trace information to AGNT*nnnn*.LOG and the agent's environment settings to AGNT*nnnn*.SET, in the directory specified in the VWS_LOGGING environment variable, where *nnnn* is the process ID.

For information about starting an agent trace, see the online help.

**OS/2 agent**

The component trace data function writes information to IWH2AGNT.*nnnn*, in the OS/2 root directory, where *nnnn* is the process ID.

For information about starting an agent trace, see the online help.

For information about program trace data for the AS/400 agent, see *Installing and Using the Visual Warehouse AS/400 Agent*.

There is a trace level for each component. Use the Configuration notebook to activate a trace level. The amount of data that is logged is dependent on the trace level that you select. For example, a level 1 trace gathers the least amount of information. You can run a level 1 trace on any component. The behavior of trace levels above 1 is component-specific. For example, the logger component uses only level 1 and level 2 traces; if you select trace level 3 or 4, you will receive the same output that you receive when you select level 2.

## Visual Warehouse programs and transformers

The supplied Visual Warehouse programs and transformers write errors to log files.

**Visual Warehouse programs**

The supplied Visual 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.

For more information, see the online help for the specific Visual Warehouse program.

**Transformers**

You can log errors that are generated when you use the transformers. You enable logging by specifying a log table name on the Program page of the Business View notebook and adding 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`.

## Start error trace files

Visual Warehouse creates three log files automatically when the logger is not running. The log file names are IWH2LOGC.LOG, IWH2LOG.LOG, and IWH2SERV.LOG. Visual Warehouse stores the files in the directory 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 Visual Warehouse server and the OLE server write to this file. The file will only exist 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 Visual Warehouse log. This is important in situations where DB2 is not starting or the file system is full. If you hear five beeps when the logger stops, you should look in this file. The server cannot start if the logger cannot start.

**IWH2SERV.LOG**
The server log contains the startup message, and will grow when the server trace is on.

# Appendix C. Registering an ODBC database

Use the ODBC drivers to register the source, target, and control databases that Visual Warehouse will access.

For information about installing and configuring the ODBC drivers for each type of agent site, see *Planning and Installing Visual Warehouse and DataGuide*.

## Registering a database source name

You must register a database as a system database source name (DSN) in ODBC:

- When you create the Visual Warehouse control database.
- Every time you define a new source database or target warehouse.

The following sections describe how to register a DSN in ODBC for Windows NT 4.0, AIX, and OS/2.

### Using Windows NT 4.0

If you are using DB2 for Windows NT 2.1.0, register the DB2 ODBC driver by double-clicking on the **ODBC Installer** icon in the **DB2** program group. You need to do this only once for each system.

To register a DSN for an ODBC driver:

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

   If you do not have a System DSN button, see “Appendix A. Troubleshooting” on page 143.
4. Click **Add**.
5. Select the ODBC driver that you want to register and click **OK**.
6. Select the desired database alias and click **OK.**
7. Close the ODBC windows.

### Using AIX

After you install the agent and configure the environment, configure the DB2 nodes and databases:

1. Log on to the DB2 instance ID.

2. Catalog the DB2 nodes and databases that you plan to access. See *Planning and Installing Visual Warehouse and DataGuide* for more information.

   If you have multiple DB2 instances on the same workstation, catalog any database that was not created on the instance you set DB2INSTANCE to in the IWH.environment file as a remote database.
3. For Intersolv ODBC users: When you define the DB2 data source, ensure that the DSN is the true name of the DB2 database name or database alias name. The user ID that starts the agent must have the DB2INSTANCE set in the .profile file.

**Tip:** If you are using Intersolv ODBC drivers, see the sample .odbc.ini file. The name of the file is /opt/IBMiwh/odbc/odbc.ini.intersolv.

For information about installing the AIX agent and configuring the environment, see *Planning and Installing Visual Warehouse and DataGuide*.

## Using Sun Solaris

After you install the agent and configure the environment, configure the DB2 nodes and databases:

1. Log on to the DB2 instance ID.
2. Catalog the DB2 nodes and databases that you plan to access. See *Planning and Installing Visual Warehouse and DataGuide* for more information.

   If you have multiple DB2 instances on the same workstation, catalog any database that was not created on the instance you set DB2INSTANCE to in the IWH.environment file as a remote database.
3. For Intersolv ODBC users: When you define the DB2 data source, ensure that the DSN is the true name of the DB2 database name or database alias name. The user ID that starts the agent must have the DB2INSTANCE set in the .profile file.

**Tip:** If you are using Intersolv ODBC drivers, see the sample .odbc.ini file. The name of the file is /opt/IBMiwh/odbc/odbc.ini.intersolv.

For more information about installing the Sun Solaris agent and configuring the environment, see *Planning and Installing Visual Warehouse and DataGuide*.

## Using OS/2

To register a DSN for an ODBC driver:

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 **Visual Warehouse Agent** folder.
2. Click **Add**.

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

## Viewing help for Intersolv ODBC drivers

You can use the ODBC Administrator to view help for each Intersolv ODBC driver that is included with Visual Warehouse.

To view the help on Windows NT:

1. Click the **ODBC Administrator** icon.
2. Click the **System DSN** tab.
3. Click **Add**.
4. Select the driver for which you want to view help.
5. Click **Finish**.

   The Setup window for the driver opens.
6. Click **Help**.

To view the help on OS/2:

1. Click the **ODBC Administrator** icon.
2. Click **Add**.
3. Select the driver for which you want to view help.
4. Click **OK**.

   The Setup window for the driver opens.
5. Click **Help**.

# Appendix D. Setting up a datamart using Microsoft Access and Microsoft Excel data

You can use Microsoft Access and Microsoft Excel data as source data for your datamart. Visual Warehouse supports the versions of Microsoft Access and Microsoft Excel that are included in Microsoft Office 97.

The following sections describe how to set up datamarts that use this source data.

## Using 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 datamart 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 button, see "Appendix A. Troubleshooting" on page 143.

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. Optionally, 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. Click **Start —> DB2 for Windows NT —> Command Line Processor**.

   A DB2 Command Line Processor window opens.
2. Enter the following command:

   ```
   CREATE DATABASE database-name
   ```

   where *database-name* is the name of the database that is to contain the target warehouse.

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 button, see "Appendix A. Troubleshooting" on page 143.
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. Optionally, 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 datamart that uses the databases

To create the Visual Warehouse definitions for the databases you created:

1. Create an information resource for the Microsoft Access database by following the instructions in the online help. 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;
       ```

       where:

       - *database-alias* is the name under which you registered the Microsoft Access database in ODBC.
       - *userID* is the user ID you specified when you registered the Microsoft Access database in ODBC.
       - *password* is the password 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 target warehouse.
   - On the Tables page:
     a. Click **Tables** to import table definitions from your Microsoft Access database.
     b. Click **OK** to list all the tables in your database.

        Verify that the tables you created in the database are in the **Tables** list.
     c. Select the table you want to use from the **Tables** list.
     d. Click **List Columns**.
     e. Select the columns you want to use from the **Columns** list.
     f. Click **Save**.
2. Create a warehouse for the DB2 database by following the instructions in the online help.

3. Create a business view that has the following attributes:
   - It uses one or more source tables from the information resource for the Microsoft Access database.
   - It creates a target table in the DB2 warehouse database.
4. Promote the business view to test status.
5. Run the business view from the Run Business View window.
6. Verify that the data 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
   ```

   where:
   - *prefix* is the prefix of the warehouse database (such as IWH)
   - *database-name* is the name of the warehouse database

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

## Using 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 datamart 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 button, see “Appendix A. Troubleshooting” on page 143.

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. Optionally, 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**.

## Creating and cataloging a target warehouse database

To create a target warehouse database in DB2:

1. Click **Start —> DB2 for Windows NT—>DB2 Command Line Processor**.

   A DB2 Command Line Processor window opens.

2. Enter the following command:

   ```
   CREATE DATABASE database-name
   ```

   where *database-name* is the name of the database that is to contain the target warehouse.

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 button, see “Appendix A. Troubleshooting” on page 143.

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. Optionally, 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 datamart that uses the databases

To create the Visual Warehouse definitions for the databases you created:

1. Create an information resource for the Microsoft Excel spreadsheet by following the instructions in the online help. 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;
       ```

       where:

       - *database-alias* is the name under which you registered the Microsoft Excel spreadsheet in ODBC.
       - *userID* is the user ID for the Visual Warehouse control database.
       - *password* is the password for the Visual Warehouse control database.
   - On the Agent Sites page, specify the agent site on which you registered the Microsoft Excel source spreadsheet and the DB2 target warehouse.
   - On the Tables page:
     a. Click **Tables** to import the table definition from your Microsoft Excel spreadsheet.
     b. Click **OK** to list all the tables in your spreadsheet.

        Verify that the tables you created in the spreadsheet are in the **Tables** list.
     c. Select the table you want to use from the **Tables** list.
     d. Click **List Columns**.
     e. Select the columns you want to use from the **Columns** list.
     f. Click **Save**.
2. Create a warehouse for the DB2 database by following the instructions in the online help.
3. Create a business view that has the following attributes:
   - It uses one or more source tables from the information resource for the Microsoft Excel spreadsheet.
   - It creates a target table in the DB2 warehouse database.
4. Promote the business view to test status.

5. Run the business view from the Run Business View window.
6. Verify that the data you created in the Microsoft Excel spreadsheet is in the warehouse database. Enter the following command in the DB2 Command Line Processor window:

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

   where:

   - *prefix* is the prefix of the warehouse database (such as IWH)
   - *database-name* is the name of the warehouse database

   You should see the data you entered in the Microsoft Excel spreadsheet.

# Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. 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. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter 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.

Licensees of this program who wish to have information about it for the purpose of enabling: (1) the exchange of information between independently created programs and other programs (including this one) and (2) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J74/G4
555 Bailey Avenue
P.O. Box 49023
San Jose, CA 95161-9023
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including, in some cases, payment of a fee.

## Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AIX
AS/400
DB2
DB2 Connect
DB2 OLAP Server
DB2 Universal Database
DataGuide
DataJoiner
DataPropagator
DRDA
IBM
IMS
MVS
OS/2
OS/390
OS/400

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries. (For a complete list of Intel trademarks see www.intel.com/tradmarx.htm)

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

# Bibliography

For questions about how Visual Warehouse works, see the online help. Visual Warehouse provides help for specific windows and for general tasks, such as creating information resources and business views.

For information about IBM products related to Visual Warehouse, visit the IBM data management Web site at http://www.software.ibm.com/data/

## Other Visual Warehouse solution publications

*IBM Visual Warehouse for Windows NT: Planning and Installing Visual Warehouse and DataGuide*, SC26-3496

*IBM Visual Warehouse for Windows NT: Messages and Reason Codes* (HTML book included in the Visual Warehouse folder)

*IBM Visual Warehouse for Windows NT: Installing and Using the Visual Warehouse AS/400 Agent*, SC26-9468

*IBM Visual Warehouse for Windows NT: Managing DataGuide*, SC26-3362

*IBM DataGuide: Programming Guide and Reference*, SC26-3368

*IBM Visual Warehouse for Windows NT: Managing ETI•EXTRACT Conversion Programs with Visual Warehouse*, SC26-9467

*IBM DB2 OLAP Server: Using DB2 OLAP Server*, SC26-9235

*IBM DB2 OLAP Server: Database Administrator's Guide*, SC26-9238

## Publications related to connectivity

*Distributed Relational Database Architecture Connectivity Guide*, SC26-4783

*IBM DB2 Connect Enterprise Edition Quick Beginnings*, S10J–7888

*IBM DB2 Connect Personal Edition Quick Beginnings*, S10J–8162

*IBM DB2 Connect User's Guide*, S10J-8163

*The Universal Connectivity Guide to DB2*, SG24–4894

## Publications related to operating systems

*Microsoft Windows NT Installation Guide*

## Publications related to databases

The Visual Warehouse Solution includes online copies of all DB2 Database Server for Windows NT books. The following books will be particularly useful:

- **IBM DB2 Universal Database publications**

  *IBM DB2 Universal Database*: *Quick Beginnings for Windows NT*, S10J-8149

  *IBM DB2 Universal Database*: *Command Reference* (S10J-8166)

  *IBM DB2 Universal Database*: *DB2 Connect User's Guide*, S10J-8163

  *IBM DB2 Universal Database*: *DB2 Connect User's Guide* (online formats only)

  *IBM DB2 Universal Database*: *Connectivity Supplement* (online formats only)

  *IBM DB2 Universal Database*: *SQL Reference*, S10J-8165
- *IBM Distributed Relational Database Architecture Connectivity Guide*, SC26–4783

## Publications related to TCP/IP

*Microsoft Windows NT Installation Guide*

*IBM DB2 Universal Database: Quick Beginnings for Windows NT*, S10J-8149

*IBM DB2 Universal Database: Installing and Configuring DB2 Clients*, S33H-0313

# Glossary

**agent site.** A workstation on which a Visual Warehouse agent is installed and runs. Visual Warehouse supports agent sites with the following operating systems: AIX, Sun Solaris, OS/2, OS/400 (V4R2 and higher) and Windows NT.

**ANOVA.** Analysis of variance. A method for determining whether the means of several different groups are equal. This method of analysis shows the relationship between a continuous dependent variable and one or more nominal independent variables.

**business metadata.** Metadata used primarily by end users to locate information in the datamart or data warehouse.

**business view.** A logical entity in Visual Warehouse that defines the structure of the output table or file, the mechanism (SQL or a program) for populating the output table or file, and that schedule by which it is populated. In general, a business view populates an output (target) table in a warehouse database by extracting data from one or more input (source) tables or files. However, you can also define a business view for launching programs that does not specify source or target tables or files.

**business view edition.** A snapshot of the data in an information resource at a particular time. You can save multiple editions of a business view to provide a record of how the data changed over time.

**cascade.** A sequence of business view processing, composed of one or more business views that are started by another business view.

**cascading business views.** Business views that are related to each other through some business process or rule.

**coefficient of variance.** A statistic that measures the amount of variation in a group relative to the mean. It is calculated by dividing a standard deviation by its mean. This statistic can be very helpful for comparing the distributions of two variables that have different means.

**correlation coefficient.** A measure of the linear relationship between two variables. A correlation coefficient of 1 or -1 indicates a perfect linear relationship; a correlation coefficient of 0 indicates no linear relationship.

**covariance.** A measure in which two variables vary together in a linear manner. Like the *correlation coefficient*, covariance can be negative or positive; unlike the correlation coefficient, covariance does not have an upper limit or a lower limit. Thus, you should not use covariance to compare the relationships between two sets of variables that are measured on a different scale.

**datamart.** A small data warehouse that contains data for use by a department or team. A datamart can be a subset of a data warehouse for your entire organization, or the data warehouse for your entire organization can be built by merging departmental datamarts.

**data group.** The Pivot Data transformer uses data groups to describe or label information that is collected into a single column. This ensures that the original meaning of the data, as expressed by the column labels in the input table, is preserved in the output table.

**data model.** A description of the organization of data in a manner that reflects the information structure of an enterprise.

**data warehouse.** A database that contains data that is extracted and derived from existing operational systems and then optimized for end-user decision making and data analysis.

**Distributed Relational Database Architecture (DRDA).** An IBM-published architecture for providing access to relational data.

**exponential smoothing.** A group of forecasting methods that use historical values and residual values to smooth a time series or forecast future values of a series.

**extract.** (1) To select and remove from a group of items those items that meet specific criteria. (2) The selection of data from a data source and the replication of the data to the data target.

**informational data.** Data that is extracted and derived from existing operational systems and then optimized for end-user decision making.

**information catalog.** The database managed by DataGuide that contains descriptive data. This data helps users identify and locate the data and information available to them.

**information resource.** The Visual Warehouse definition of a resource from which a business view extracts data or to which a business view writes data. In general, information resources correspond to operational data stores. However, an information resource can correspond to a file to which a business view writes data for temporary storage.

**initialization.** The process in which Visual Warehouse creates the control tables required to store Visual Warehouse metadata.

**key column.** A column whose values uniquely identify a row in a table. If a value is not duplicated in any other row in that table, that value is unique.

**linear relationship.** A measure between two variables indicating that as the values in one variable increase, the values of the second variable increase or decrease in a linear fashion. When you plot the values of one variable against the values of another variable, the data points should fall into a long, narrow band. If they form some other pattern, the relationship between the two variables is probably not linear.

**metadata.** Data that describes the characteristics of stored data; descriptive data. For example, the metadata for a database table might include the name of the table, the name of the database containing the table, and the names of the columns in the table.

**moving average.** A measuring method where a new average is computed by dropping the oldest observation and including the most recent observation in a time-series data set.

**nonrelational data.** Information stored as flat files or having a structure (such as hierarchical data) that is not based on tabular relationships between stored items. VSAM and IMS data is nonrelational data.

**online analytical processing (OLAP).** Software used to quickly analyze information that has been summarized into multidimensional views and hierarchies.

**operational data.** Data produced by a variety of batch and transactional applications in the day-to-day operations of a business.

**period table.** A table that contains a column that consists of a generated date, time, or timestamp value. A period table can also contain columns that are based on the date/time value for the row, or based on parameters that the user specifies.

**relational data.** Information stored using a tabular structure between stored items.

**relational database management system (RDBMS).** Software used to create, access, and maintain relational data in the form of tables consisting of rows and columns.

**rolling sum.** A measure of cumulative values in a series. For example, in a 12-period rolling sum, the first sum is the cumulative value of the first through the 12th observations. The second rolling sum is the cumulative value of the second through the 13th observations.

**rules table.** A table that designates the values that the Clean Data transformer will use during the find-and-replace process.

**smoothing.** The process of removing fluctuations in an ordered series so that the first differences are regular and the higher order differences small. The smoothing process approximates true values and reduces errors in observation.

**source database.** A database from which Visual Warehouse or Visual Warehouse program extracts data for further processing

**source file.** A file from which Visual Warehouse or a Visual Warehouse program extracts data for further processing

**star schema.** A type of relational database schema consisting of a fact table and a set of dimension tables. The fact table holds the actual data values for the database, and the dimension tables hold data about members and their relationships.

**Structured Query Language (SQL).** A command language used with relational databases. The language consists of statements to insert, update, delete, query, and protect data.

**tag language.** A set of keywords that identify the metadata for Visual Warehouse or DataGuide objects.

**tag language file.** A file that contains metadata for objects to import into Visual Warehouse or DataGuide.

**target database.** A database to which Visual Warehouse or a partner application writes the data after processing it.

**target file.** A file to which Visual Warehouse or a partner application writes the data after processing it.

**target table.** A table in the target warehouse to which Visual Warehouse writes data when a business view runs.

**target warehouse.** The Visual Warehouse definition of a database used as part of a data warehouse or datamart.

**technical metadata.** Metadata that describes the data in the datamart or data warehouse. Technical metadata includes information about source data, target data, and the rules used to extract, filter, enhance, cleanse, and transform source data to target data.

**variance.** A measure of a distribution's dispersal around a mean.

**Visual Warehouse administrative client.** A Visual Warehouse component that contains the management functions of Visual Warehouse. The administration client is used to administer the users, groups, organizations, business views, and other objects in the data warehouse.

**Visual Warehouse agent.** A Visual Warehouse component that handles the flow of data between data sources and target tables in a data warehouse. It also starts Visual Warehouse programs, issues system commands, and collects metadata.

**Visual Warehouse program.** A logical entity within Visual Warehouse that represents an application that you want Visual Warehouse to start. The Visual Warehouse agent can start a Visual Warehouse program during the population of a target table, after the population of a target table, or by itself.

**Visual Warehouse server.** A Visual Warehouse component that schedules business views, controls and monitors the Visual Warehouse agent, and logs messages.

**warehouse database.** A database that is part of a data warehouse or datamart that end-users will use to run queries and reports.

**warehouse table.** A table in the warehouse database.

# Index

Program Number: 5639-VW5

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC26-8822-02