IBM Visual Warehouse for Windows NT

Installing and Using the Visual Warehouse AS/400 Agent

Version 3 Release 1 CSD 2

SC26-9468-00

IBM Visual Warehouse for Windows NT

Installing and Using the Visual Warehouse AS/400 Agent

Version 3 Release 1 CSD 2

SC26-9468-00

Note

Before using this information and the product it supports, be sure to read the information in “Notices” on page 119.

First Edition (September 1998)

This edition applies to Version 3 Release 1 of Visual Warehouse Base Edition (5697-VW3), and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Contents

Contents

About this book

This book describes the steps that are required to install a Visual Warehouse AS/400 agent and to build a warehouse that uses the AS/400 agent.

This book does not include information about:

- Running business views (except in the appendix on the external trigger program)
- Creating and using subjects
- Performing intermediate or complex tasks associated with running a data warehouse

For information about these tasks, see the online help and *Managing Visual Warehouse*.

Who should read this book

This book is intended for administrators and datamart designers who want to use Visual Warehouse Version 3.1 to create a data warehouse that uses the AS/400 agent. This book is intended for both new and experienced users.

How to send your comments

Your feedback is important in helping to provide accurate and high-quality information. If you have any comments about this book or any other Visual Warehouse Version 3.1 documentation, visit the following Web site:

`http://www.software.ibm.com/data/vw`

There you will find a feedback page where you can enter and submit your comments.

Chapter 1. Introducing the Visual Warehouse agent

Visual Warehouse can safely extract operational data from single or multiple sources, transform it, and write it to a target database table. End users can then use data access tools to analyze the data in the target database.

You can use Visual Warehouse to extract information from rows in tables in different databases and write the information to a new table. Using Visual Warehouse allows you to improve security and performance system-wide. Having a datamart allows you to take end-user ad hoc query processing off of transactional systems. As a result, the datamart tables can be designed to optimize query performance rather than transaction processing performance. Having end-users query extracts of transactional data is also more secure since users no longer will have access to mission-critical data stores.

The following list describes the basic concepts of Visual Warehouse:

Information resource

In Visual Warehouse, a data source is referred to as an information resource. An *information resource* is a logical collection of tables and rows that are to be extracted from a single database.

Warehouse

A target database is referred to as a warehouse. A *warehouse* is a database that contains tables and columns that are of interest to a particular group of end users. After you populate a warehouse, you can allow end users to view the data that is contained in the warehouse.

Business views

You use business views to define and schedule each step in the extraction, transformation, and writing of the data. A business view might join data from four different sources and write it to one target. You might create a series of business views that extract operational data and write it to tables in a warehouse database. These business views might refine the operational data into data that is designed for specific analysis by end users. A *business view* is a step in the transformation process from the data's source format to its target format.

Suppose that you want to extract data from different databases, convert the data to a single format, and write the data to a warehouse table. You would use separate business views to extract the data from the information resources 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.

Business views move and transform data by using SQL or user-written programs. A single business view can actually use both methods.

A basic business view performs the following tasks:

- It extracts data from at least one source table or file.
- It uses Visual Warehouse's SQL processing to transform the data.
- It writes the transformed data to a target table in a warehouse.

Because a populated business view is a table in a database, you can use it as a source table for other business views.

Visual Warehouse programs

You can create business views that use Visual Warehouse programs to perform bulk operation transformations. A *Visual Warehouse program* is a user-written program that you specify to run either as a business view or to run after a business views runs successfully. For example, you can write a program to export data from a table, manipulate that data, and write it to an interim output resource or a warehouse table. The program must contain code to connect to the information resource, the output resources, or both. Visual Warehouse provides the connection parameters that the program requires, including the names of the warehouse tables and information resource tables, user ID, and password.

The following sample Visual Warehouse programs are shipped with the AS/400 agent:

- VWPLOADI
- VWPLOADR
- VWPRCPY
- VWPFTP

For information about these programs, see "Appendix A. The sample Visual Warehouse programs" on page 71 and "Appendix B. Examples of sample Visual Warehouse program definitions" on page 95.

Running a business view

To start the transfer of data between the source and the target, and any transformation of that data, you must run your business view. You can run a business view manually, or you can schedule a business view to run at a set time.

When you attach a schedule to a business view, Visual Warehouse runs the business view 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. You can also schedule other business views to run after another business view runs successfully.

This book describes how to define a business view so that you can run it manually, run it at a scheduled time, or schedule it and keep the option to run it manually. It also describes how to schedule a business view to run after another business view runs successfully.

Business view edition

When you run a business view, you can create a snapshot of the data in the information resource at any particular time. This snapshot of data is called a *business view edition*.

Visual Warehouse stores each business view edition in the target table of the business view. You can save one or more editions of the same business view.

Visual Warehouse assigns a status to each business view edition, such as scheduled, successful, or failed. For more information about business view editions, including how to view their statuses, see the online help for the Operations Work in Progress window. The online help also provides information about running business view editions manually.

Visual Warehouse agent

The actual transfer of data between the source and the target, and any transformation of that data, is performed by the *Visual Warehouse agent*. A Visual Warehouse agent can also start Visual Warehouse programs and perform administrative tasks, such as collecting metadata.

The Visual Warehouse agent receives SQL commands from the Visual Warehouse server. Then, the agent passes the SQL on to the source database or the target database.

Visual Warehouse server

The Visual Warehouse server manages the Visual Warehouse agent. The Visual Warehouse server tells the agent what it should do, based on information that you supply.

Visual Warehouse Agent Daemon

The Visual Warehouse agent daemon is a continuously running background job that waits for requests from the Visual Warehouse server. When the daemon receives a request, it starts an agent process to service the request.

Agent process
Visual Warehouse starts one agent instance, or job, for each unit of work that needs to be performed on the AS/400. Each agent instance is referred to as an *agent process*.

Agent site
The machine on which an agent runs is referred to as an *agent site*. The agent site can be on the machine that contains either the source database or the target database.

Local agent
An agent that is installed on the same machine with the Visual Warehouse server is referred to as a *local agent* (or the *default agent*).

Remote agent
An agent that is installed on a machine other than the Visual Warehouse server is referred to as a *remote agent*.

Since the Visual Warehouse server only runs on Windows NT, the Visual Warehouse AS/400 agent is always considered a remote agent.

Prerequisite tasks

Before you can install the AS/400 agent, you should complete the following tasks:

- Plan your warehousing solution. For information about planning your warehouse, see *Managing Visual Warehouse*.
- Determine the location of your agent site. You can choose from one of three different configurations. “Agent site configurations” on page 8 describes these configurations.
- Set up access to your data. “Connectivity requirements for the Visual Warehouse server and the AS/400 agent” on page 7 and “Connectivity requirements for remote databases” on page 8 describe the connectivity requirements that must be met before you can use the AS/400 agent.
- Verify that all AS/400 software is at the required code level and the required maintenance level. For more information, see “Software requirements” on page 7.
- Obtain a user ID with *ALLOBJ authority so that you can install the AS/400 agent. After installation is complete, you only need a user ID that has authority to create, destroy, and execute AS/400 objects to use the product.
- Install Visual Warehouse Version 3 Release 1 CSD 2 or higher. For more information, see *Installing Visual Warehouse and DataGuide*.

- Define security and privileges for your warehouse. For information about defining warehouse security, see *Managing Visual Warehouse*.

You might also 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 references, see "Bibliography" on page 121.

Task overview for installing the agent and creating a data warehouse

To create a data warehouse that uses the AS/400 agent, you must complete the following tasks:

1. Install the AS/400 agent.
2. Establish connectivity to local and remote databases. You also must complete any communication setup tasks that are required for connectivity to your remote databases.
3. Start the agent daemon.
4. Verify TCP/IP connectivity between the Visual Warehouse server machine and the AS/400 agent site machine.
5. Define the AS/400 agent site to Visual Warehouse.
6. Create and define an information resource.
7. Create and define a warehouse.
8. Create a Visual Warehouse program definition for each program that you want the Visual Warehouse AS/400 agent to start for you.
9. Create and define business views to use in your data warehouse. You can define the following types of business views:
   - A business view that uses SQL processing
   - A business view that uses a Visual Warehouse program

   Your data warehouse can contain business views of either or both types.

The chapters that follow describe these tasks. After you complete these tasks, you will be ready to run your business views.

For general information about running business views, see the online help and *Managing Visual Warehouse*.

For general information about starting a business view from the AS/400 command line, see “Appendix D. Using the external trigger program” on page 107.

For information about promoting a business view, see “Appendix C. Changing the status of a business view” on page 103.

Chapter 2. Planning for and installing the AS/400 agent

This chapter describes the information that you need to plan for the AS/400 agent and the information you need to install it.

Software requirements

To install and use the AS/400 agent, you must have a user ID that has ALLOBJ and JOBCTL authority. This level of authority is required for both the AS/400 RSTLICPGM command and the Visual Warehouse STRVWD and ENDVWD commands.

The user profile that starts the agent daemon should also have *PGMR (bind capability) as the User Class.

To use the AS/400 agent, you must have the following products:

- Visual Warehouse for Windows NT Version 3 Release 1 CSD 2 or later.
- OS/400 Version 4 Release 2 or later. Ensure that you apply all prerequisite software fixpacks and PTFs. For a current maintenance list, see the README file.

To determine the software that is installed on your AS/400 machine, enter `DSPSFWRSC` at an AS/400 command line.

Connectivity requirements for the Visual Warehouse server and the AS/400 agent

The Visual Warehouse server uses TCP/IP to communicate with the AS/400 agent and the agent daemon. For this communication to take place, the Visual Warehouse server must be able to recognize the fully qualified host name of the AS/400 agent. Also, the AS/400 agent must be able to recognize the fully qualified host name of the Visual Warehouse server.

By default, the Visual Warehouse server sends messages to the AS/400 agent daemon on port 11001 and receives responses on port 11000. The Visual Warehouse logger uses port 11002.

If another application uses one of the default Visual Warehouse port numbers, you can change the port number that Visual Warehouse uses. If you need to change a port number, see “Visual Warehouse and the AS/400 agent are unable to communicate” on page 111.

Connectivity requirements for remote databases

You can access remote databases via the AS/400 agent only through Systems Network Architecture (SNA) connectivity that uses IBM Distributed Relational Database Architecture (DRDA). Currently, the AS/400 agent does not support DRDA over TCP/IP.

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

- DB2 for OS/400
- DB2 for OS/390
- DB2 Universal Database

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

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

Tip: You should be able to connect to a remote database from Visual Warehouse and query it if the following conditions are met:

- You can connect to the remote database from the AS/400 agent.
- You can query the remote database from the AS/400 interactive SQL facility (STRSQL).

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

Agent site configurations

The AS/400 agent receives SQL commands from the Visual Warehouse server, and then passes the commands to the source or target databases.

You can set up the AS/400 agent in one of three configurations. The following list describes each configuration.

The agent, source, and target are located on the same machine.
The AS/400 agent is installed on a machine that contains both the information resource table and the warehouse table in the same database, as shown in Figure 1 on page 9.

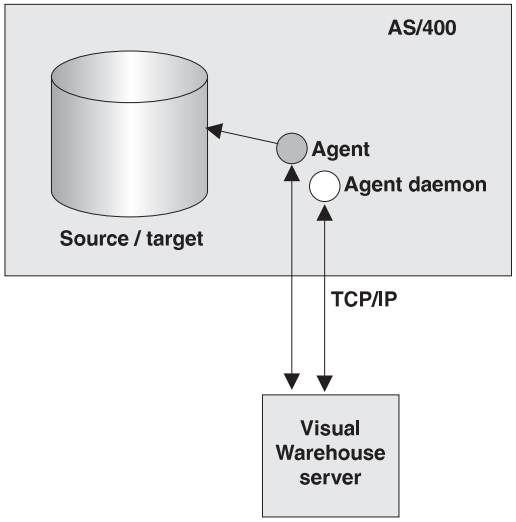


Figure 1. The agent, source, and target are located on the same machine.

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

The agent and target are located on the same machine.
The AS/400 agent is installed on the machine that contains the warehouse table, as shown in Figure 2.

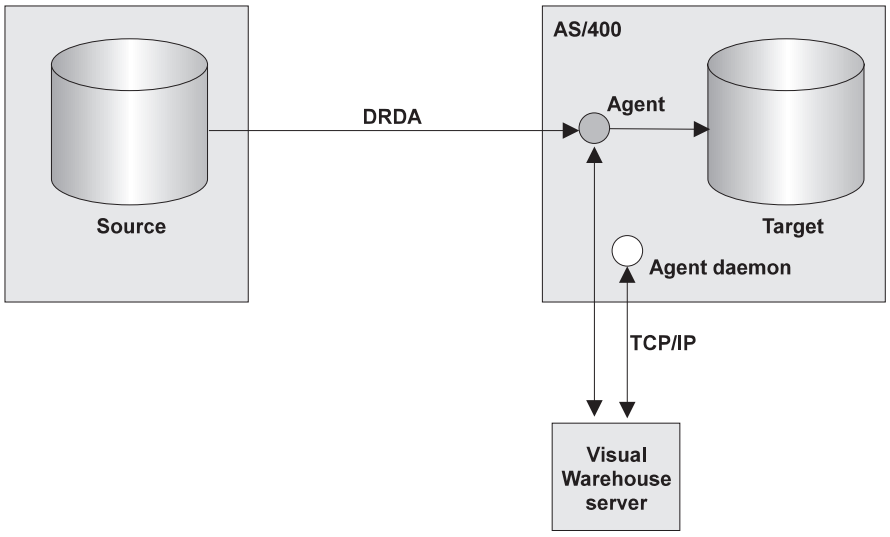


Figure 2. The agent and target are located on the same machine.

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

This configuration offers the best performance when the source and target reside on different AS/400 machines.

The agent and source are located on the same machine.
The AS/400 agent is installed on the machine that contains the information resources, as shown in Figure 3.

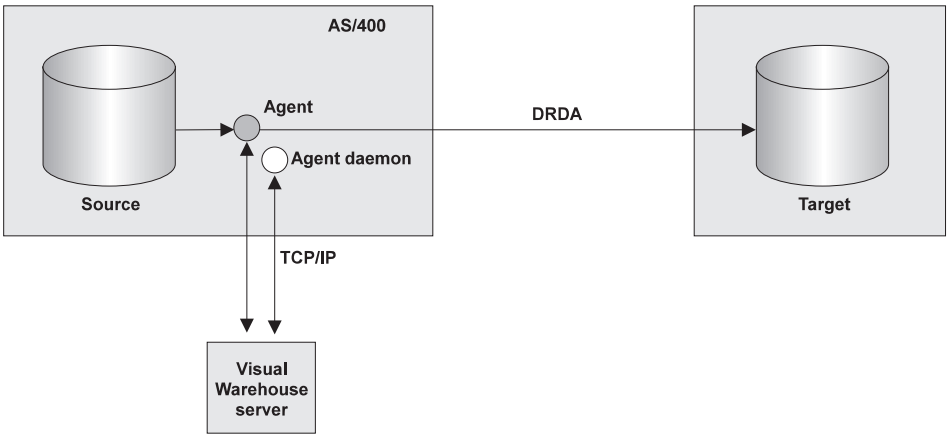


Figure 3. The agent and source are located on the same machine.

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

After you set up access to your data and you determine the location of your agent, you must define security for your warehouse. For information about defining warehouse security, see *Managing Visual Warehouse*.

Installing the AS/400 agent

To install and use the AS/400 agent, you must have a user ID that has ALLOBJ and JOBCTL authority. This level of authority is required for both the AS/400 RSTLICPGM command and the Visual Warehouse STRVWD and ENDVWD commands.

To install the AS/400 agent:

1. Log on to OS/400.
2. At an AS/400 command line, enter:
   ```
   RSTLICPGM LICPGM(5639VW5) DEV(OPT01)
   ```

   where OPT01 is the name of the device that holds the Visual Warehouse CD-ROM.

The /QIBM/UserData/IWH subdirectory and the /QIBM/ProdData/IWH subdirectory are created by the install process. Traces are written to the /QIBM/UserData/IWH subdirectory.

Chapter 3. Setting up the AS/400 agent

After you install the AS/400 agent, you must complete the following tasks:

1. Establish connectivity to local and remote databases.
2. Start the agent daemon.
3. Verify communication between the Visual Warehouse server and the AS/400 agent.
4. Define the AS/400 agent site to Visual Warehouse.

These tasks are described in this chapter.

Establishing connectivity to local and remote databases

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

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

For example, Fred is creating a data warehouse. He wants to catalog the database names for a database that is named Sales and a database that is named Expenses. The database named Sales is located on the same machine as the AS/400 agent. The database named Expenses is located on the remote machine that the agent will access. Table 1 describes how Fred should catalog each database on each machine.

Table 1. How to catalog local and remote database names

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

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

Setting up the AS/400 agent

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

```
ADDRDBDIRE databasename locationname
```

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

Ensure that you supply both the name of the database and the name of the location, even if they are the same name.

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

Attention: If you change the name of a database in the Remote Database Directory, you must update each information resource that refers to it. Failure to do so will result in information resource database connection errors.

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

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

Starting the agent daemon

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

The user profile that starts the agent daemon should have *PGMR (bind capability) as the User Class.

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

To verify that the agent daemon started:

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

The agent daemon runs as a background job.

Verifying communication between the Visual Warehouse server and the AS/400 agent

Visual Warehouse uses TCP/IP to communicate with a remote agent site machine. For this communication to take place, the Visual Warehouse server must be able to recognize the fully qualified host name of the AS/400 agent site machine. Also, the AS/400 agent site machine must be able to recognize the fully qualified host name of the Visual Warehouse server.

To verify that one machine recognizes the fully qualified host name of the other machine, use the **ping** command from a command prompt.

For example, the fully qualified host name for an AS/400 agent site machine is `abc.xyz.commerce.com`. To verify that the Visual Warehouse server recognizes the fully qualified host name of the agent site machine, from a DOS command prompt, enter:

```
ping abc.xyz.commerce.com
```

Ensure that you verify communication from both the AS/400 agent site machine to the Visual Warehouse server machine and vice-versa.

Defining the AS/400 agent site to Visual Warehouse

You must now define the agent site to Visual Warehouse so that Visual Warehouse can access the agent site.

To define an agent site to Visual Warehouse:

1. Start the Visual Warehouse desktop.
2. Select **Definitions → Agents Definition**.
   The Agent Site window opens.
3. Select **File → New**.
   The Agent Site notebook opens.

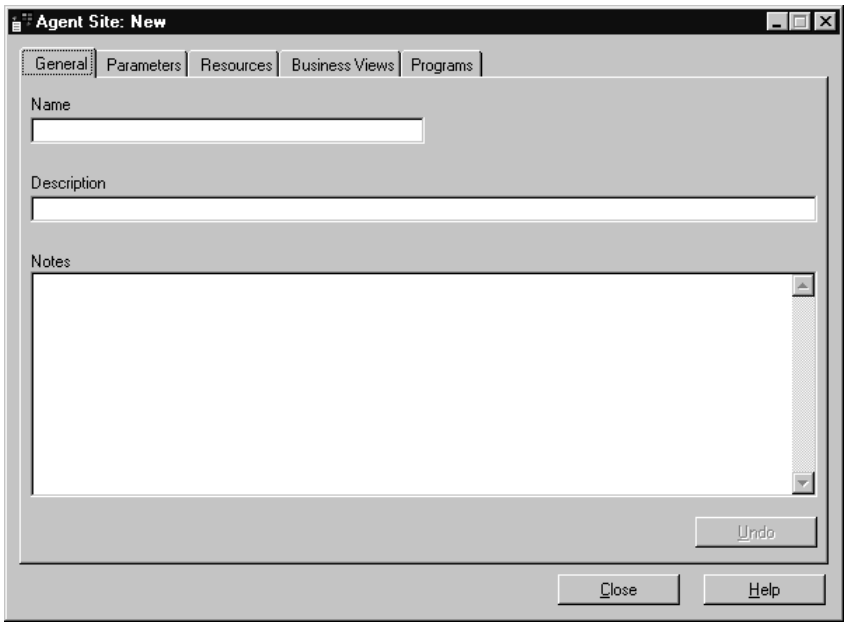


Figure 4. Agent Site notebook

4. In the **Name** field, type a name for the agent site. This is a logical name that is used only by Visual Warehouse to identify the agent site. The name can be up to 80 characters long and is case sensitive.
5. In the **Description** field, type information about your Visual Warehouse agent site. For example, Fred might enter `AS/400 agent for testing new warehouse. Contact Fred, ext. 1234`.

   This step is optional.
6. Click the **Parameters** tab.

   Figure 5 on page 17 shows an example of a completed Parameters page. In this example, Fred has entered information for an AS/400 agent site that is located in San Jose.

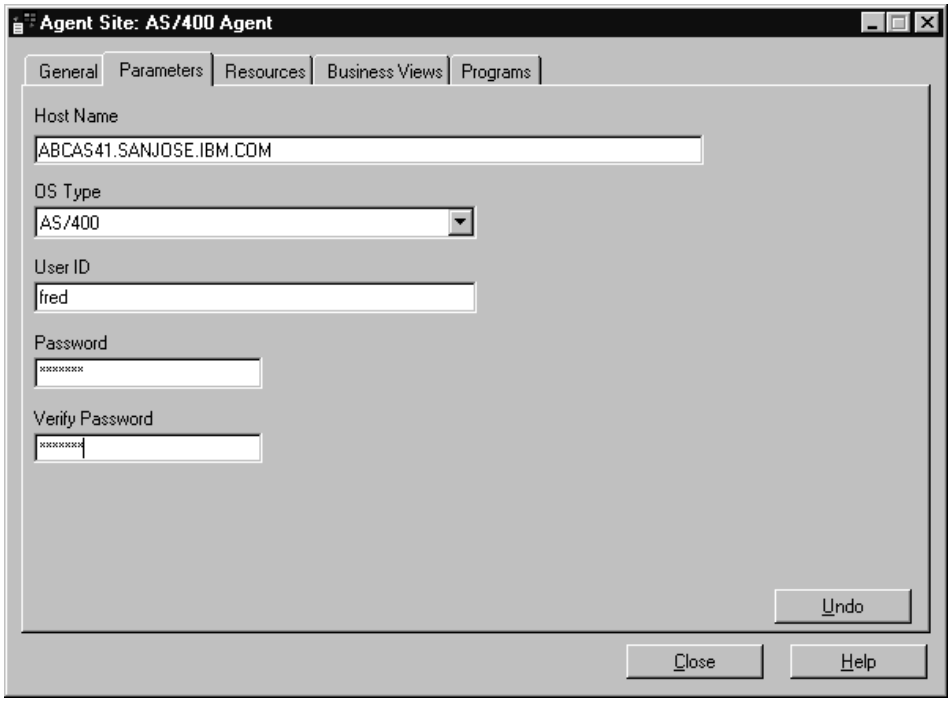


Figure 5. An example of a completed parameters page

7. In the **Host Name** field, type the fully qualified TCP/IP host name of the agent site.
8. From the **OS Type** drop-down list, select **AS/400**.
9. In the **User ID** field, type the valid user ID under which the Visual Warehouse agent daemon will run in uppercase letters. You must use uppercase letters because OS/400 only allows uppercase user IDs.

   A user ID can be up to 8 characters long.
10. In the **Password** field, type the password for the AS/400 user ID under which the agent daemon will run in uppercase letters. You must use uppercase letters because the AS/400 only allows uppercase passwords.

    A password can be up to 10 characters long.
11. In the **Verify Password** field, type the valid password again, in uppercase letters.
12. When you are finished entering your information, click **Close**.

If you are an experienced Visual Warehouse user and you have already created a data warehouse that you plan to use with the AS/400 agent, ensure that the following Visual Warehouse objects are defined properly:

- Any AS/400-based information resource that the AS/400 agent will access. Specify the database type for your information resource as **AS/400 RISC** for AS/400 systems Version 3 Release 6 and above, or **AS/400 CISC** for AS/400 systems Version 3 Release 1 and Version 3 Release 2. To select your database type, click the **Database** tab in the Information Resource notebook.
- Any AS/400-based warehouse to which the AS/400 agent will write data. Specify the database type for your warehouse as **AS/400 RISC** for AS/400 systems Version 3 Release 6 and above, or **AS/400 CISC** for AS/400 systems Version 3 Release 1 and Version 3 Release 2. To select your database type, click the **Database** tab in the Warehouse notebook.
- Any Visual Warehouse programs that the AS/400 agent will run.
- Any business view that will interact will the AS/400 agent.

If you are a new user, you can now create your data warehouse. To create a data warehouse, you must complete the following tasks:

1. Create an information resource, as described in "Chapter 4. Creating and defining an information resource" on page 19.
2. Create a warehouse, as described in "Chapter 5. Creating and defining a warehouse" on page 27.
3. Define a program, as described in "Chapter 6. Defining a Visual Warehouse program that runs on OS/400" on page 33. This step is not necessary if you do not want to use a Visual Warehouse program.
4. Create business views, as described in "Chapter 7. Creating and defining a business view that uses Visual Warehouse SQL processing" on page 41 and "Chapter 8. Creating and defining a business view that uses a program" on page 59.

Chapter 4. Creating and defining an information resource

In Visual Warehouse, a data source is referred to as an information resource. You use the Information Resource notebook to specify information about the source data that you will use to create the business views to populate your data warehouse.

Before you can build a data warehouse, you must define the information resources that Visual Warehouse can access. Before you can define an information resource, you must create it.

To create an information resource:

1. Select the **Sources** tab on the Visual Warehouse desktop.
2. Select **File → New**.

   The Resource Type window opens.
3. In the **Resource Type** list, select **DB2 Family**, as shown in Figure 6.

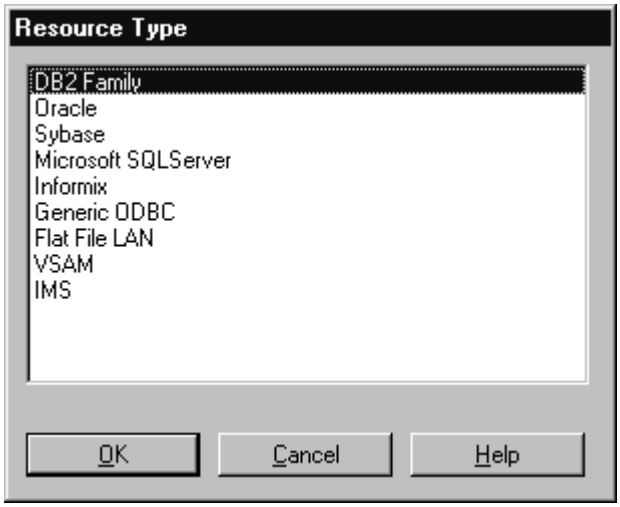


Figure 6. Select DB2 Family from the Resource Type list.

4. Click **OK**.

The Information Resource notebook opens, as shown in Figure 7 on page 20.

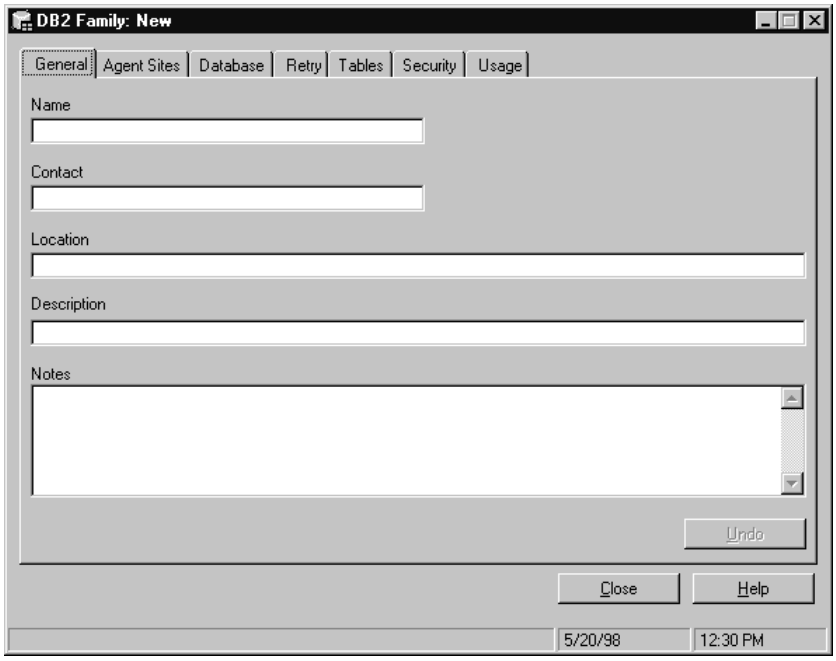


Figure 7. The Information Resource notebook

As you enter information and move from page to page in the notebook, your information is automatically saved.

The sections that follow describe each page of the Information Resource notebook (except for the Usage page, which is mainly referential). These sections provide the information you need to define an information resource that an AS/400 agent can access.

Providing general information about the information resource

When you open an Information Resource notebook, the General page is displayed. Use this page to enter basic information about the Information Resource that you are defining:

1. In the **Name** field, type a business name for the information resource.

   This field is required. You will use this name to refer to your information resource throughout the Visual Warehouse system. Use a descriptive name that Visual Warehouse administrators, business analysts, and other Visual Warehouse users will understand.

If your information resource and your warehouse are defined in the same database, they cannot have the same name. For example, you can name your information resource DB2400_SRC and name your warehouse DB2400_TGT.

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

2. In the **Contact** field, type the name of a person to contact about the information resource.
3. In the **Location** field, type the location of the information resource. For example, this location might be the department that owns the information resource, or it might be the room where the database server is located.
4. In the **Description** field, type a description of the information resource.
5. In the **Notes** field, type any additional information about the information resource.
6. Click the **Agents Sites** tab. The Agent Sites page is displayed.

Selecting an agent site

You use the Agents Sites page to select the AS/400 agent site that you will use to extract data from your information resource.

1. From the **Available Agent Sites** list, select the name of the AS/400 agent site that you want to use to access your information resource.
2. Click **Add**. Your agent site is added to the **Selected Agent Sites** list.
3. From the **Selected Agent Sites list**, select **Default VW Agent Site** and click **Remove**.

   The default agent site takes precedence over remote agent sites. To prevent Visual Warehouse from using the default agent site, you must ensure that the default agent site is not selected for both your information resource and your business view.
4. Click the **Database** tab. The Database page is displayed.

Selecting a database

Use the Database page to select the database that the information resource will use.

For example, Fred is creating an information resource that will use a DB2 for AS/400 database. The database is registered on the remote database directory of the AS/400 agent site as COST. Figure 8 on page 22 shows the information that Fred types on the Database page.

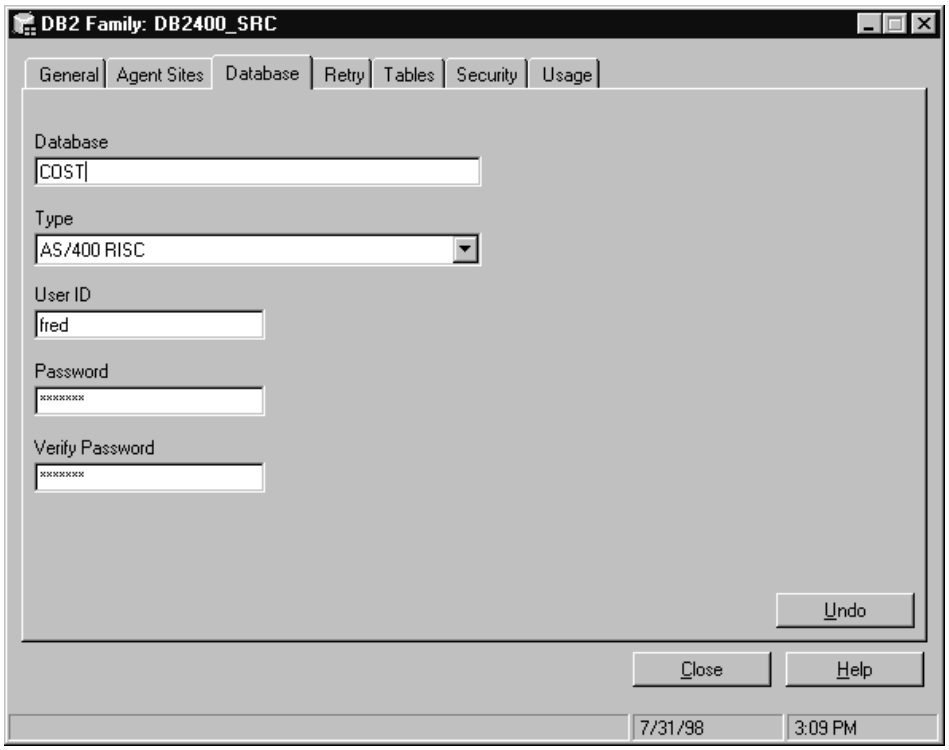


Figure 8. An example of a completed Database page

To select a database for your information resource:

1. In the **Database** field, type the name of the database as it is registered in the remote database directory on the AS/400 agent site. The database name for an information resource is always the database name as it is known on the agent site that will access it.

   Ensure that the name that you enter in the **Database** field has a maximum of 18 characters. While this field has space for a name up to 40 characters in length, the remote database directory on the AS/400 machine uses database names that are a maximum of 18 characters in length.
2. From the **Type** list, select **AS/400 RISC** for AS/400 systems Version 3 Release 6 and above, or **AS400 CISC** for AS/400 systems Version 3 Release 1 and Version 3 Release 2.
3. In the **User ID** field, type the user ID that will access the database from the AS/400 agent site. If the database that the user ID accesses is an AS/400 database, type the user ID in uppercase letters. You must use uppercase letters because OS/400 only allows uppercase user IDs.

If the database that the user ID accesses is not an AS/400 database, type the user ID in accordance with the authentication rules for the database to which you are connecting.

4. In the **Password** field, type the password for the user ID that will access the database. If the password belongs to a user ID that accesses an AS/400 database, type the password in uppercase letters. You must use uppercase letters because OS/400 only allows uppercase passwords.

   If the password belongs to a user ID that accesses a non-AS/400 database, type the password in accordance with the authentication rules for the database to which you are connecting.
5. In the **Verify Password** field, type the password again.
6. Click the **Tables** tab. The Tables page is displayed.

Defining tables

Before you can access data from a table in an information resource, you must define the table to Visual Warehouse. You use the Tables page to define tables in one of two ways:

- Import table definitions into an information resource
- Create a table definition

In most cases, you will use only one of these methods to define your tables. You are not required to complete both tasks to define tables to Visual Warehouse.

Attention: Columns that are defined as BINARY WITH PRECISION are interpreted as INTEGER by SQL SELECT. This is a permanent AS/400 restriction. For more information, see “Columns defined as BINARY WITH PRECISION are interpreted as INTEGER” on page 112.

Importing table definitions

To import a table definition to Visual Warehouse:

1. Click **Tables**.

   The Catalog Import window opens.
2. If you want to select from tables with a certain library or collection, in the **Table Qualifier** field, type the name of the library or collection. The **Table Qualifier** field is case sensitive.

   If you know the name of the table that you want to import, in the **Table Name** field, type the name. The **Table Name** field is case sensitive.

   If you do not know the name of the table, leave the **Table Name** field blank.

Visual Warehouse excludes system tables from the table list by default. To include system tables in the list, select the **Include system tables** check box.

3. Click **OK**.

   If you typed the name of a library or collection in the **Table Qualifier** field, the Table List window displays all the tables in that library or collection at the data source. Use this window to select tables to include in your information resource.

   If you typed a name for the table in the **Table Name** field, the Column List window opens. Use this window to select columns from a source table to include in your information resource.

   If you did not type a name in either field, the Table List window displays all the tables at the data source.
4. From the **Tables** list, select the table that you want to include.
5. Click **List**.
6. Click **Save**.
7. If the **Table name** field is blank or you typed a library name or collection name in the **Table Qualifier** field:
   a. In the Table List window, click **Tables**.
   b. Click **OK**, then click **Close**. The Table List window closes.
   c. In the Catalog Import window, click **Close**.

   The table is now displayed in the **Tables** list on the Tables page of the Information Resource notebook.

Creating a table definition

If you can't import a table definition into your information resource, you must create one. To create a table definition for an information resource:

1. Click **New**.

   The Table notebook opens.
2. On the General page, type the name of the new table in the **Name** field. Because the name will be used in SQL statements, it must be a fully qualified name.

   Table names can be case sensitive, or have other restrictions, for some data source types. See the documentation for your data source type for information about naming conventions.
3. In the **Description** field, type a description for the database.
4. In the **Notes** field, type any additional notes.
5. Click the **Columns** tab. The names of the table columns are listed in the large display field on this page.
6. To create a column description, click **New**.

Figure 9 shows an example of a completed Column description. In this example, Fred is creating a description for a column named `Serial Number` in a table named `EQUIPMENT`. The column has a CHAR data type and a length of 9, contains text, and does not allow nulls.

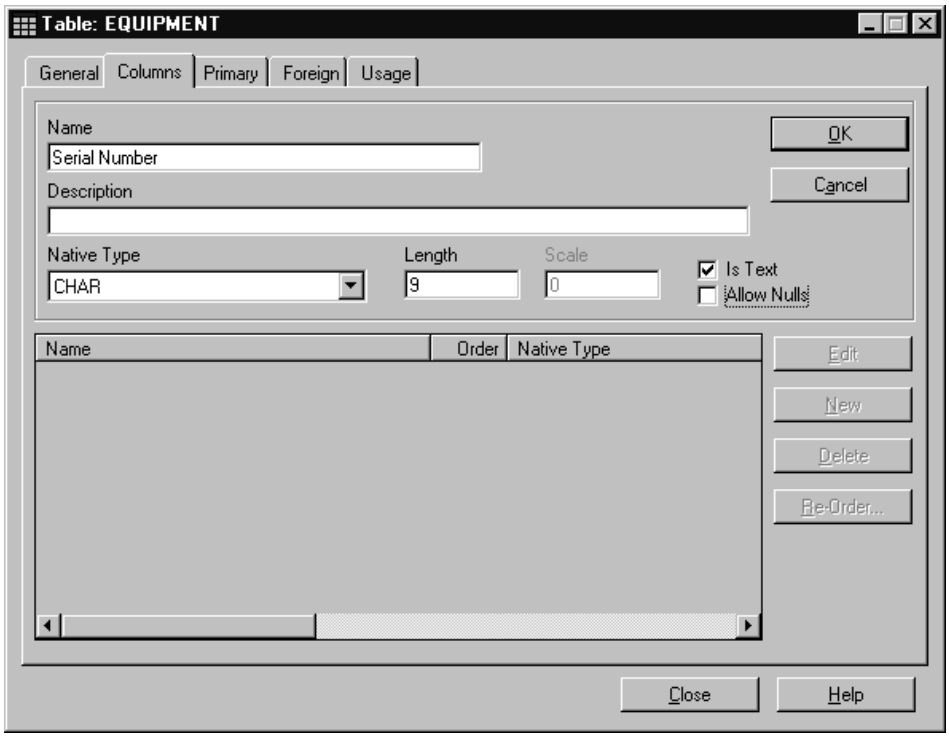


Figure 9. An example of a completed column description

7. In the **Name** field, type the name of the column.

   Column names can be case sensitive for some data source types. See the documentation for your data source type for information about naming conventions.
8. In the **Description** field, type a description for the column.
9. Select a data type from the **Native Type** list. The types shown in the list depend on the type of database that you specified on the Database page of the Information Resource notebook.
10. Specify length information in the **Length** or **Precision and Scale** field depending on the data type of the column. For example, you must specify a length for CHAR data types.
11. To ensure that the code page is translated properly, check the **Is Text** check box if the CHAR or VARCHAR field contains text.

12. If null (nonexistent or unknown) values are allowed in the table, check the **Allow NULL** box.
13. Click **Close**.
14. Click the **Security** tab. The Security page is displayed.

Defining security

If you did not define security groups for Visual Warehouse, this page will be blank. For information about defining security groups, see *Managing Visual Warehouse*.

Use the Security page to authorize a group of users to access this information resource definition. The users can then create business views that use this information resource. You can also use this page to deny access to a group of users to this information resource.

To define which groups can create business views that use this information resource:

1. From the **Available Security Groups** list, select the name of the group that you want to authorize.
2. Click **Add**.

The group is added to the **Selected Security Groups** list.

To deny access to a group of users, ensure that the name of the group is not on the **Selected Security Groups** list. If you want to remove the name of a group from the **Selected Security Groups** list:

1. Select the name of the group for which you want to deny access to the information resource definition.
2. Click **< Remove**.
3. When you are finished, click **OK**.

You have defined an information resource that the AS/400 agent can access. You must now define a warehouse.

Chapter 5. Creating and defining a warehouse

A warehouse is a DB2 database that contains business views that are of interest to a particular group of end users. To define a warehouse, you use the Warehouse notebook.

You create a warehouse to contain your business views. To create a warehouse:

1. Select the **Warehouses** tab from the Visual Warehouse desktop.
2. Select **File → New Warehouse**.

The Warehouse notebook opens, as shown in Figure 10. Use this notebook to specify information about your new warehouse.

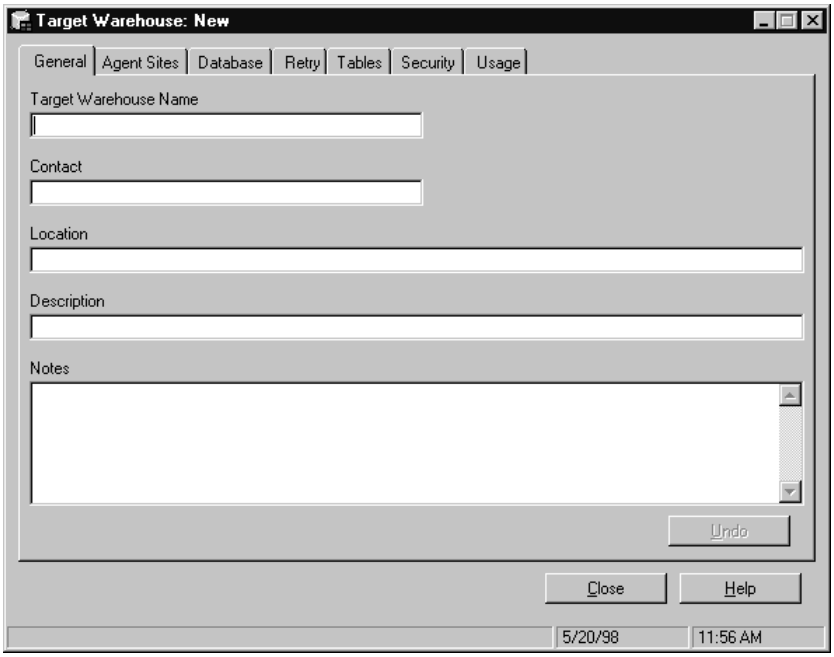


Figure 10. The Warehouse notebook

As you enter information and move from page to page in the notebook, your information is automatically saved.

Creating a warehouse

The sections that follow describe tasks that you must complete using certain pages of the Warehouse notebook. These sections provide the information you need to define a warehouse that accesses an AS/400 agent.

Providing general information about the warehouse

When you open a Warehouse notebook, the General page is displayed. Use this page to enter basic information about the warehouse that you are defining:

1. In the **Target Warehouse Name** field, type a name for your warehouse.

   The name can be up to 80 characters long and is case sensitive.

   The first character of the name must be alphanumeric.

   If your information resource and your warehouse are defined in the same database, they cannot have the same name. For example, you can use `DB2400_SRC` to describe your information resource and `DB2400_TGT` to describe your warehouse.
2. In the **Contact** field, type the name of a person who will serve as a contact for this warehouse.
3. In the **Location** field, type the location of the warehouse. For example, this can be the room number where the database server for the warehouse is located.
4. In the **Description** field, type a description of the warehouse.
5. In the **Notes** field, type any additional information about the warehouse.
6. Click the **Agent Sites** tab. The Agent Sites page is displayed.

Selecting an agent site

Use the Agent Sites page to select the AS/400 agent site that you will use to write data to your warehouse. This agent site should be the same agent site that you selected when you defined your information resource.

When you add an AS/400 agent site, the agent site can access the warehouse only if the agent uses DRDA connectivity.

1. From the **Available Agent Sites** list, select the name of the site that you want to use to access your warehouse.
2. Click **Add Site**.

   Your agent site is added to the **Selected Agent Sites** list.

   The agent site can now access the warehouse.
3. Click the **Database** tab. The Database page is displayed.

Selecting a database

Use the Database page to select the database that your warehouse will use.

The database that you use as a warehouse must be cataloged in the DB2 directory at the agent site machine. If you did not already catalog this database, do so now. For information about cataloging the database name, see "Establishing connectivity to local and remote databases" on page 13.

To select a database:

1. In the **Database** field, type the name of the database as it is registered in the remote database directory on the AS/400 agent site. The database name for a warehouse is always the database name as it is known on the agent site that will access it.

   Ensure that the name that you enter in the **Database** field has a maximum of 18 characters. While this field has space for a name up to 40 characters in length, the remote database directory on the AS/400 machine uses database names that are a maximum of 18 characters in length.
2. From the **Type** list, select **AS/400 RISC** for AS/400 systems Version 3 Release 6 and above, or **AS/400 CISC** for AS/400 systems Version 3 Release 1 and Version 3 Release 2.
3. In the **User ID** field, type the user ID that will access the database from the AS/400 agent site. If the database that the user ID accesses is an AS/400 database, type the user ID in uppercase letters. You must use uppercase letters because OS/400 only allows uppercase user IDs.

   If the database that the user ID accesses is not an AS/400 database, type the user ID in accordance with the authentication rules for the database to which you are connecting.
4. In the **Password** field, type the password for the user ID that will access the database. If the password belongs to a user ID that accesses an AS/400 database, type the password in uppercase letters. You must use uppercase letters because OS/400 only allows uppercase passwords.

   If the password belongs to a user ID that accesses a non-AS/400 database, type the password in accordance with the authentication rules for the database to which you are connecting.
5. In the **Verify Password** field, type the password again.
6. Click the **Tables** tab. The Tables page is displayed.

Importing an existing table definition

Use the Tables page to import the definition of an existing table (if you want to use one) from your warehouse database to Visual Warehouse. Visual Warehouse can then use this definition as the target table for your warehouse.

If you do not want to import an existing table definition, leave this page blank.

To import a table:

1. Click **Tables**.

   The Catalog Import window opens.
2. If you want to select from tables with a certain library or collection name, in the **Table Qualifier** field, type the name of the library or collection. The **Table Qualifier** field is case sensitive.

   If you know the name of the table that you want to import, in the **Table Name** field, type the name. The **Table Name** field is case sensitive. If you do not know the name of the table, leave the field blank.

   Visual Warehouse excludes system tables from the table list by default. To include system tables in the list, select the **Include system tables** check box.
3. Click **OK**.

   If you typed a name of a library or collection in the **Table Qualifier** field, the Table List window displays all the tables in that library or collection at the data source. Use this window to select tables to include in your information resource.

   If you typed a name for the table in the **Table Name** field, the Column List window opens. Use this window to select columns from a source table to include in your information resource.

   If you did not type a name in either field, the Table List window displays all the tables at the data source.
4. If you typed a library name or collection name in the **Table Qualifier** field or you left the **Table name** field blank, select the table that you want to include from the **Tables** list. Otherwise, go to step 7.
5. Click **Columns**.
6. In the **Columns** list, select the columns that you want to include. The default selects all column names.
7. Click **Save**.
8. If you typed a library name or collection name in the **Table Qualifier** field or you left the **Table name** field blank:
   a. In the Table List window, click **Import**.
   b. Click **OK**.

9. Click the **Security** tab. The Security page is displayed.

Attention: To specify an existing table as a target table, you must clear the **Visual Warehouse Created Table** check box when you define information for your business view.

Defining security

Use the Security page to define which groups of users have authority to create business views to populate the warehouse:

1. From the **Available Groups** list, select the name of the group that you want to authorize.
2. Click **Add Group**.

   The group is added to the **Selected Security Groups** list.
3. Click **OK**. You have defined a warehouse that will use the AS/400 agent.

You must now define a Visual Warehouse program that runs on the AS/400 agent site, if you plan to use one. For more information, see “Chapter 6. Defining a Visual Warehouse program that runs on OS/400” on page 33.

If you do not plan to use a Visual Warehouse program, you must now create and define a business view that uses Visual Warehouse SQL processing. For more information, see “Chapter 7. Creating and defining a business view that uses Visual Warehouse SQL processing” on page 41.

Chapter 6. Defining a Visual Warehouse program that runs on OS/400

This chapter describes how to define a program that runs on OS/400 to Visual Warehouse. If you do not want to use Visual Warehouse programs in the operation of your data warehouse, see “Chapter 7. Creating and defining a business view that uses Visual Warehouse SQL processing” on page 41.

Before you can use an AS/400 agent to run a Visual Warehouse program, you must store the program on the AS/400 agent site and define the program to Visual Warehouse. If you did not store your program on your AS/400 agent site, do so now.

After you define your Visual Warehouse program, you must ensure that the program definition is added to your AS/400 agent site definition. Then, you can define a business view that will use the program.

“Verifying that a Visual Warehouse program is added to your AS/400 agent site definition” on page 40 describes how to ensure that your program definition is added to your AS/400 agent site definition.

“Chapter 8. Creating and defining a business view that uses a program” on page 59 describes how to define a business view that uses a Visual Warehouse program.

The following sample Visual Warehouse programs are shipped with the AS/400 agent:

- VWPLOADI

  This program LOADs data from a flat file into a DB2 table. The LOAD appends new data to the end of existing data in the table. In addition, prior to doing the append, VWPLOADI exports the original data to a backup file so that the original state of the data can be restored if the LOAD operation fails.
- VWPLOADR

  This program LOADs data from a flat file into a DB2 table. The LOAD completely replaces the existing data in the table.
- VWPRCPY

  This program transfers a text file from one system to another.
- VWPFTP

  This program transfers files from one system to another.

For more information about these programs, see “Appendix A. The sample Visual Warehouse programs” on page 71.

For examples of the information that you provide when you define sample programs to Visual Warehouse, see “Appendix B. Examples of sample Visual Warehouse program definitions” on page 95.

You use the Program notebook to define a program to Visual Warehouse. To open the Visual Warehouse program notebook:

1. From the Visual Warehouse desktop, select **Definitions → Visual Warehouse Programs**.

   The Programs window opens.
2. Select **File → New**.

   The Program notebook opens, as shown in Figure 11.

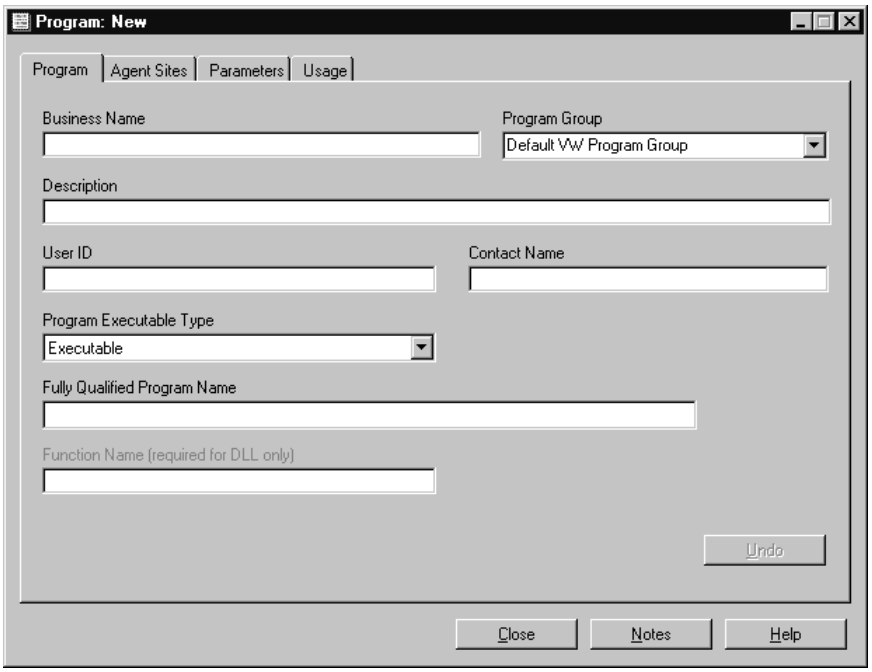


Figure 11. The Program notebook

The sections that follow describe each page of the Program notebook (except the Usage page, which you use for reference purposes). These sections provide the information that you need to define a program that runs on an AS/400 agent site.

At any time, to save the information you enter into the Program notebook, click **OK**. Visual Warehouse saves the information in the Program notebook, and the notebook closes.

To open the notebook again, double-click the program icon in the Programs window.

Providing general information about the program

Use this page to define information about the Visual Warehouse program that you stored on the AS/400 agent site.

For example, Fred wants to define a program that he created to Visual Warehouse. Figure 12 shows the information that Fred types on the Program page:

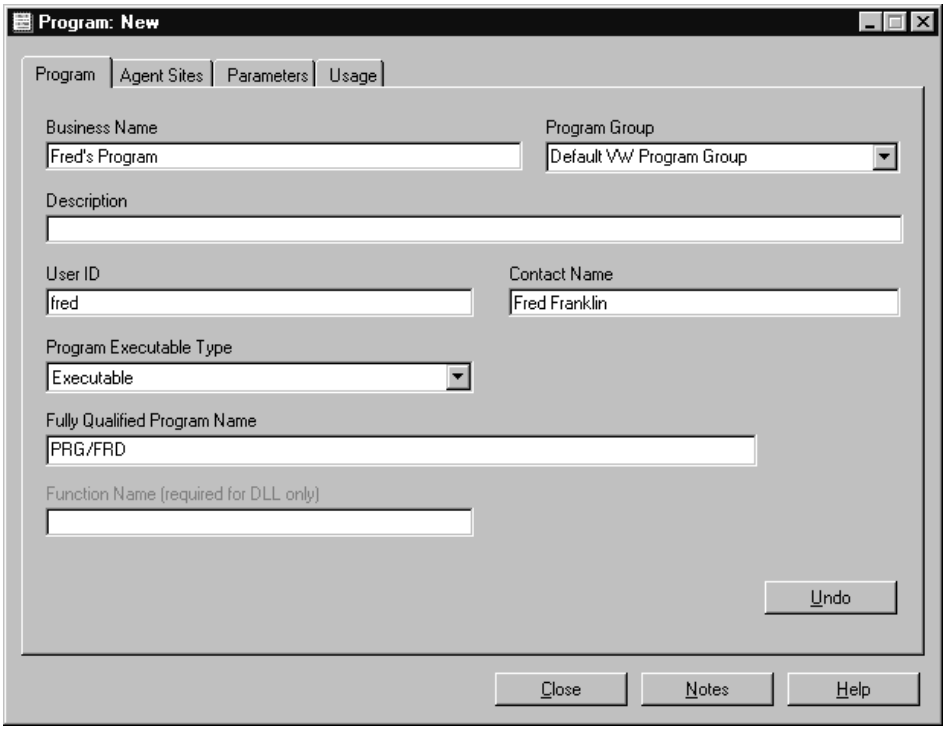


Figure 12. An example of a completed Program page

To define information about a program to Visual Warehouse:

Defining a program

1. In the **Business Name** field, type a name for the Visual Warehouse program.

   The name can be up to 80 characters in length and is case sensitive.

   The first character of the name must be alphanumeric.
2. From the **Program Group** list, select a program group in which to define the program. A *program group* is a logical grouping of programs.

   To add a new program group, select a field in the **Program Group** list. Then type the name of the new group in the field. To change the name of an existing group, type the new name of the group over the old one.
3. Select a program type from the **Program Executable Type** list.

   On OS/400, Visual Warehouse programs can be executables (program objects) or command programs (CL programs). DLLs are not supported.
4. In the **Fully Qualified Program Name** field, type the fully qualified name of the program.

   For programs that are stored on OS/400, the format for the fully qualified name is *library/program*. For example, to run a program called TRX in a library that is called DAT, you specify the path to Visual Warehouse as DAT/TRX.
5. Click the **Agent Sites** tab. The Agent Sites page is displayed.

Adding an agent site

Use the Agent Sites page to add the AS/400 agent site to the Visual Warehouse program:

1. From the **Available Agents** list, select the AS/400 agent site on which the program is installed.
2. Click **Add**.
3. Click the **Parameters** tab. The Parameters page is displayed.

Tip: It is recommended that you only use one agent site per each Visual Warehouse program definition. The agent site that you select must be the site on which the Visual Warehouse program will run.

Setting parameters

Use the Parameters page to set parameters for your Visual Warehouse program. The **Parameter List** displays the parameters and tokens that Visual Warehouse will pass on to your program. You can select system parameters or system tokens for your program to use, or you can define your own parameters.

Adding system parameters or tokens to the Parameter List

To add system parameters and tokens to the Parameter List:

1. Click **Add**.

   The Add Parameters window opens.
2. From the **Select Parameter Type** list, select either **System Parameters** or **System Tokens**.
3. Select a parameter or token from the **Available Parameters** list, and then click **Add**.

   The parameter is added to the **Parameter List** of the Program notebook.
4. Repeat steps 2 and 3 until you add all the parameters that your program requires.
5. When you are finished adding parameters or tokens, click **Close**.

Defining your own parameter to the Parameter List

To define your own parameter to the Parameter List:

1. On the Parameters page, click **Insert**.

   A new row is added to the **Parameter List**.
2. In the **Parameter Name** field, type the name of the parameter.
3. In the **Parameter Value** field, type the value of the parameter.

Removing a parameter from the Parameter List

To remove a parameter from the Parameter List:

1. Select a parameter by clicking the number to the left of the parameter name.
2. Click **Remove**.

Saving the parameter list

When you finish defining the parameter list, click **Show**. The command-line string that Visual Warehouse will pass to your program is displayed.

When you click **Show**, the parameters in the Parameter List are saved to the Visual Warehouse program, and the program definition is saved to Visual Warehouse.

For example, Fred added system parameters to the Parameter List for his program definition and clicked **Show**. Figure 13 on page 38 shows the Parameters page for his program definition, including the command-line string.

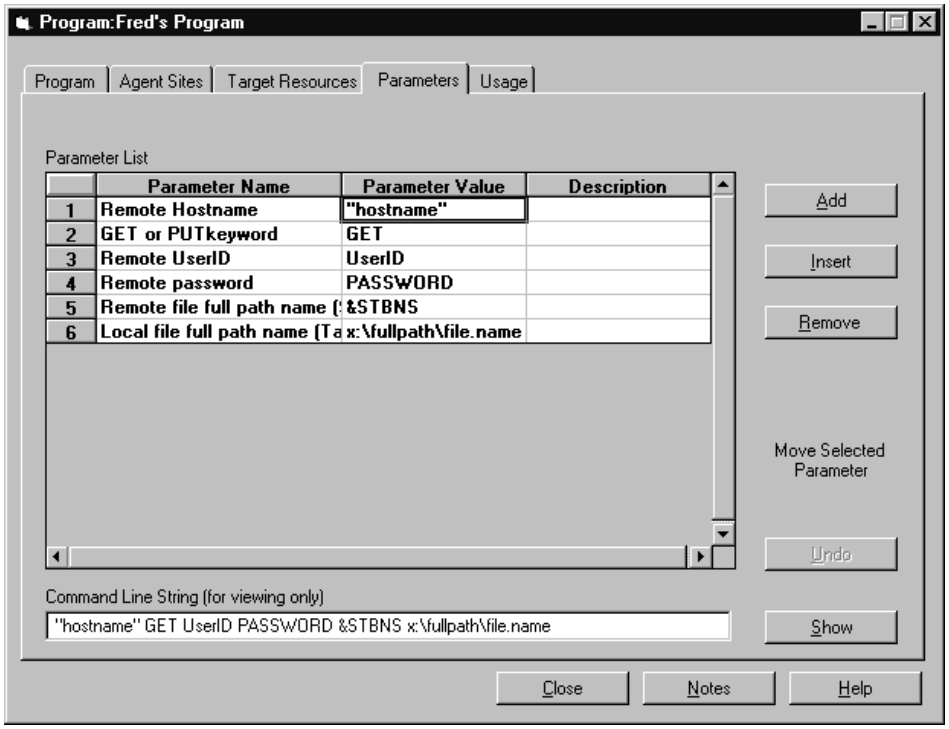


Figure 13. An example of a completed Parameters page

Providing default system parameter definitions

You will notice that the value for each parameter in the **Parameter List** has a generic definition. For example, the parameter value for the Remote password parameter is PASSWORD. If you want to create default parameter definitions for your program, you should define the values for these parameters now.

To define a default system parameter value for your program:

1. Double-click the **Parameter Value** field for the parameter that you want to define.

   You can now edit the **Parameter Value** field.
2. Clear away the generic parameter value in the **Parameter Value** field.
3. Type the correct text in the **Parameter Value** field.

   For example, Fred wants to assign default definitions to the program Fred's Program. He types the new values that he wants assigned to the parameters, as show in Figure 14 on page 39:

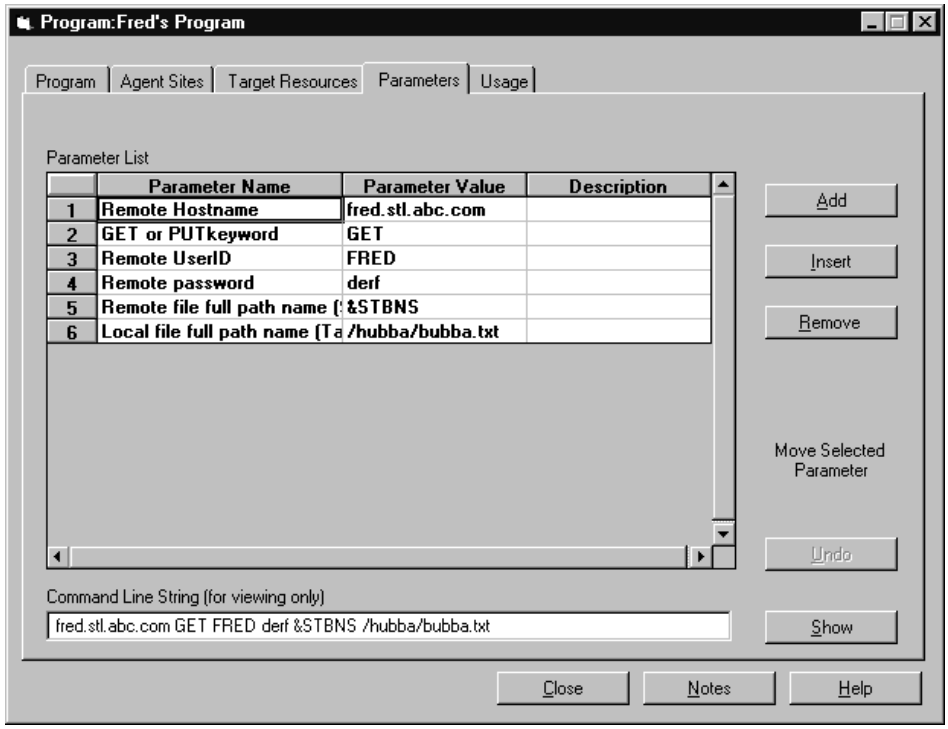


Figure 14. An example of new values for system parameters

4. When you are finished defining your program parameters, click **Show**.

   The parameter value is saved and a new command-line string is displayed.

If necessary, you can override a default parameter value on a case by case basis. For more information, see “Chapter 8. Creating and defining a business view that uses a program” on page 59.

When you are finished defining parameters, click **Close**. Your program is defined to Visual Warehouse.

You must now verify that your program is added to your agent site.

Verifying that a Visual Warehouse program is added to your AS/400 agent site definition

To verify that a program is added to your AS/400 agent site definition:

1. Select **Settings → Agent Definition** from the Visual Warehouse desktop.

   The Agent Sites window opens.
2. Double-click the icon for your AS/400 agent site.

   The Agent Site notebook opens.
3. Select the **Programs** tab.
4. If your program is not listed in the **Selected Programs** list, double-click on the program in the **Available Programs** list.

   The program moves to the **Selected Programs** list.
5. Click **OK**.

You must now create and define a business view that uses your Visual Warehouse program.

Chapter 7. Creating and defining a business view that uses Visual Warehouse SQL processing

Before you can create a business view that uses SQL processing, you must open the warehouse that is to contain it.

To open the warehouse:

1. Select the **Warehouses** tab on the Visual Warehouse desktop.
2. Double-click the name of the warehouse that you want to open.

   The Business View List window opens.

After you select a warehouse to contain your business view, you are ready to create the business view.

After you open your warehouse, you create a business view. To create a business view, you use the Create Business View window.

For example, Fred uses the Create Business View window to create a business view that is named Sum of Revenue, as shown in Figure 15 on page 42.

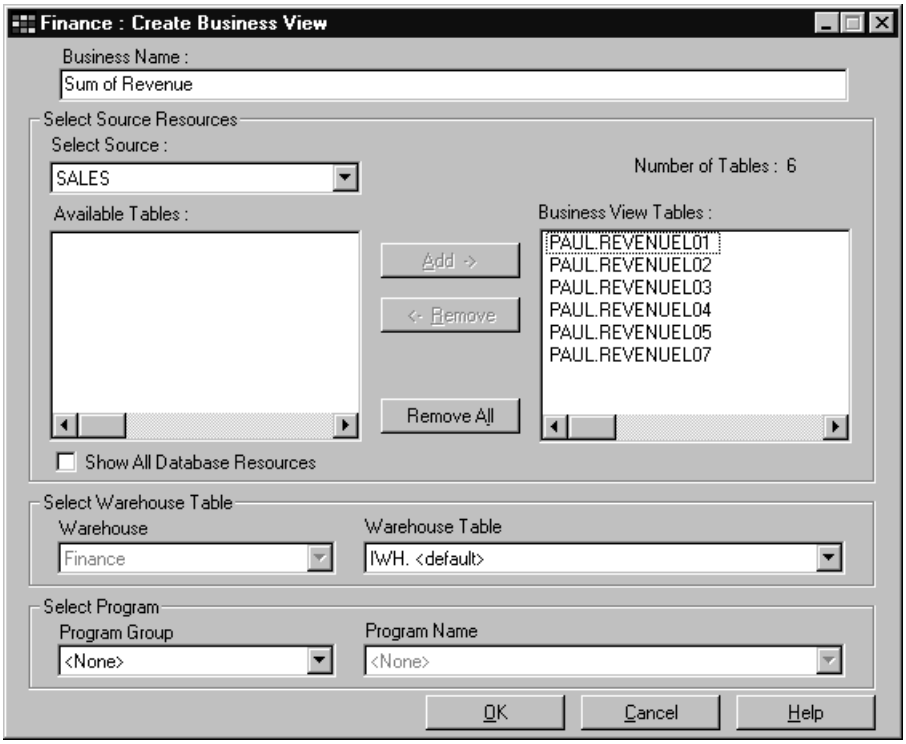


Figure 15. The Create Business View window

To create a business view:

1. Select **File → New** from the Business View List window.

   The Create Business View window opens.
2. In the **Business Name** field, type a name for your business view. This field is required.

   The name can be up to 80 characters long and is case sensitive.

   In the English language version of Visual Warehouse, the name cannot contain & as the first character. Other naming restrictions might vary depending on the language.

   The business view initially specifies a target table that is based on the business name of the business view. You can change the table to another target table in the warehouse or alter the CREATE statement for the table.
3. From the **Select Source** list, select the name of the source from which you want to extract data. You will probably see both information resources and warehouses in this list. You can use the tables that exist in a warehouse as sources for a business view.
4. From the **Available Tables** list, select one or more tables to include in your business view.

5. Click **Add**.

   The tables that you selected are displayed in the **Selected Tables** list.

6. Click **OK**.

   The business view notebook opens.

   Figure 16 shows an example of the Business View notebook for the business view that Fred creates.

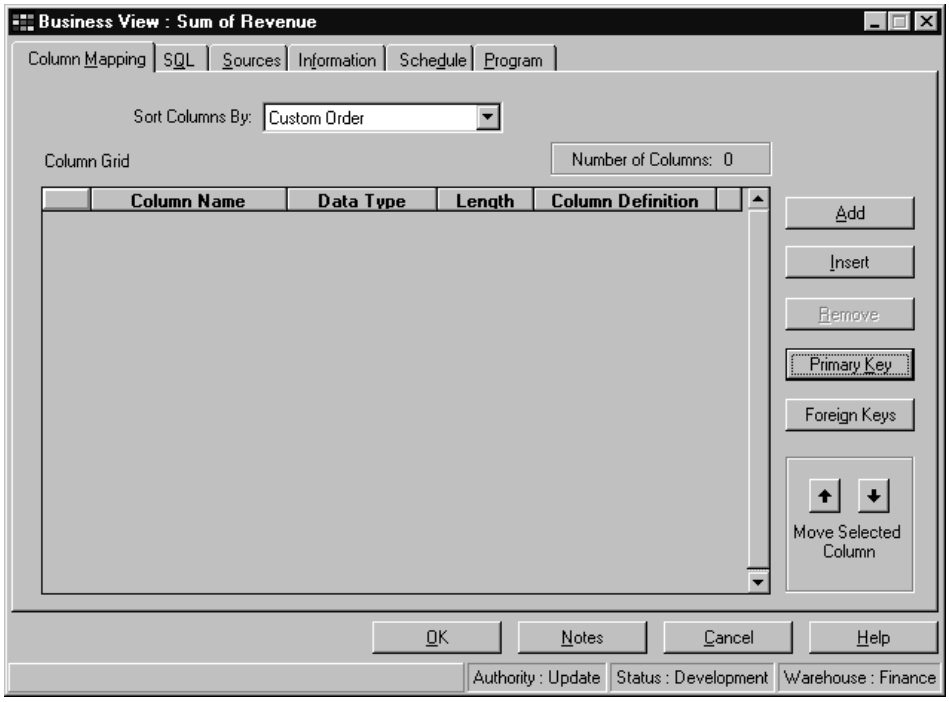


Figure 16. An example of a Business View notebook

You use the Business View notebook to define your business view.

To save the information you enter into the Business View notebook at any time, click **OK**. The information in the Business View notebook is saved, and the notebook closes.

To open the notebook again, double-click on the business view icon in the **Business View** list.

Specifying source column definitions

Use the Column Mapping page to specify the source column definitions that you want to include in your target table. You can include column definitions that come from your information resource and column definitions that do not come from your information resource.

Adding a data column definition from an information resource

From the Column Mapping page, you can access the Add Columns window. You use the Add Columns window to add column definitions to a business view.

For example, Fred wants to add columns from a table that is named PAUL.REVENUEL01 to his business view. Figure 17 on page 45 shows the Add Columns window that Fred uses:

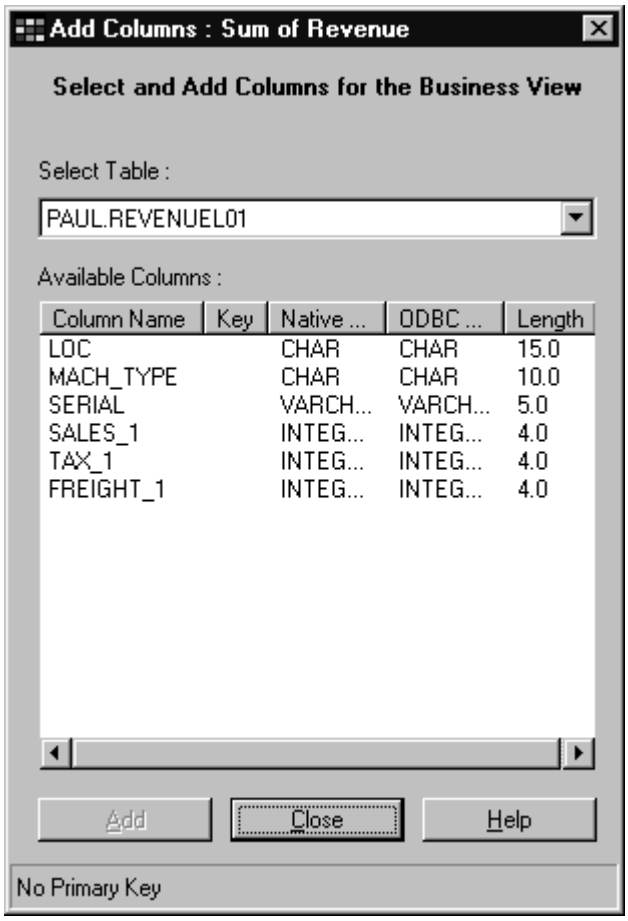


Figure 17. The Add Columns window

To add column definitions to a business view:

1. Click **Add**.

   The Add Columns window opens.
2. From the **Select Table** list, select the table from which you want to add columns.

   The columns for that table are displayed in the **Available Columns** list.
3. From the **Available Columns** list, select the column that you want to add.

   Tip: To select more than one column, hold down the Shift key and click additional column names.
4. Click **Add**.

   The columns are added to the column grid on the Column Mapping page of the Business View notebook. If the column name is a duplicate of a

column name displayed in the **Column Name** list, you will be prompted to change the name to make it unique.

When you add a column from an information resource, Visual Warehouse converts data types from the source table into the equivalent DB2 Common Server or DB2 for OS/400 data type. Visual Warehouse converts Sybase, Oracle, Microsoft SQLServer Version 6, and Informix data types into the appropriate DB2 data types. If there is no conversion available for a specific data type, a message states that there is no data type match available. You can specify or change the data type in the column grid by clicking the down arrow of the **Datatype** field.

Adding a column that does not belong to an information resource

You can use the Column Mapping page to add column definitions to your business view that do not belong to an information resource. To add column definitions, you use this page to access the Input Column Name window and the Column Definition window.

For example, Fred wants to add a column named TOTAL to his business view. The TOTAL column will be a combination of a column named SALES and a column named TAX. First, he enters information about the new column in the Input Column Name window. Then, he defines the column in the Column Definitions window. Figure 18 on page 47 shows the information that Fred enters into the Column Definition window.

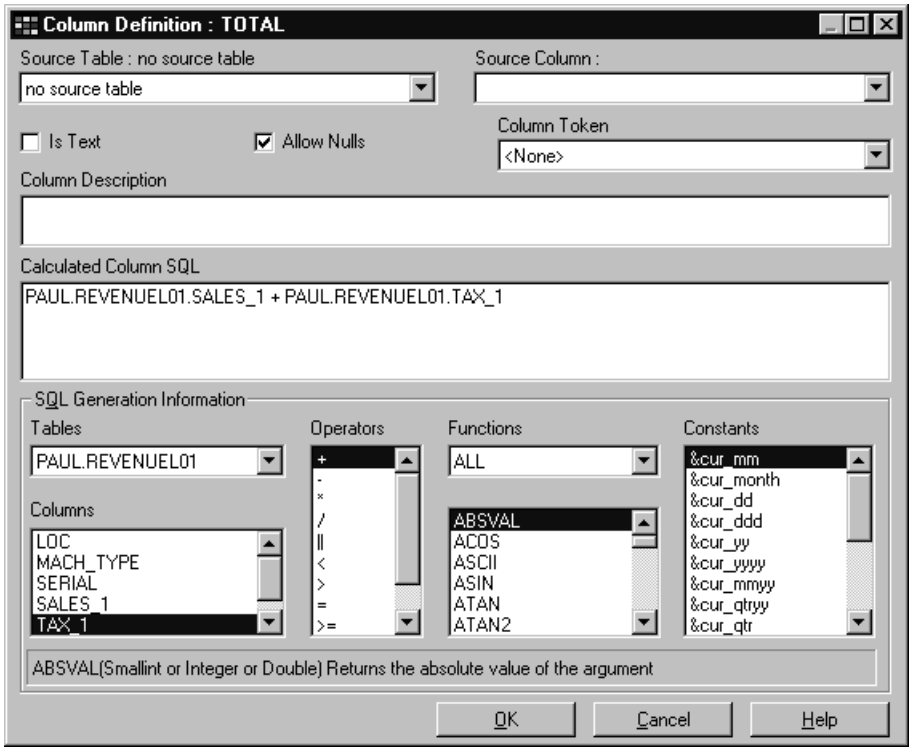


Figure 18. A completed column definition

To add a column definition that does not belong to an information resource:

1. Click **Insert**.

   The Input Column Name window opens.
2. In the **New Column Name for New Inserted Column** field, type the name of the column.
3. Click **OK**.

   A column with a data type of integer, fixed length, and SQL statement of 0, is added to the column grid.
4. If the data type of the column is not integer, click the down arrow in the **Datatype** field to select a new data type.
5. In the **Length** field, type the length of the column.
6. Click **...** (the ellipsis at the end of the column) to change the SQL statement (and other definitions) of the column.

   The Column Definition window opens.

   If you want the value of the column to equal a constant, select the constant from the **Column Token** list.

If you want the value of a column to equal the result of an SQL statement:

a. Select a table name from the **Tables** list.
b. Double-click a column name in the **Columns** list.

   The fully qualified name for the column (table name and column name) is displayed in the **Calculated Column SQL** field.
c. Double-click an item from the **Operators** list or the **Functions** list. If you select a function, the status bar displays a brief description of the function.

   The item is displayed in the **Calculated Column SQL** field.
d. Double-click an item from the **Constants** list or the **Columns** list, or type a value in the **Calculated Column SQL** field.
e. Repeat steps 6.a through 6.d until you complete the expression.

Tip: To change an SQL statement, type over the statement in the **Calculated Column SQL** field or rebuild the statement. If you do not change the SQL statement, the column will always have a value of 0.

7. Click the **SQL** tab. The SQL page is displayed.

Editing SQL

When you specify columns on the Column Mapping page, Visual Warehouse automatically generates a SELECT statement based on the columns for the business view. Use the SQL page to:

- Edit the SELECT statement directly.
- Regenerate the SELECT statement, based on changes that you make.
- Join tables to include in your business view.
- Define a selection WHERE clause.
- Generate a GROUP BY clause.
- Define a HAVING clause.

Editing the SELECT statement directly

1. Click **SQL**.

   The Modify SQL window opens.
2. Type over the SQL statement in the **SQL Statement** field.

Regenerating a SELECT statement

You can also regenerate a SELECT statement.

For example, Fred wants to regenerate a SELECT statement for his business view. To do this, he uses the Modify SQL window, as shown in Figure 19.

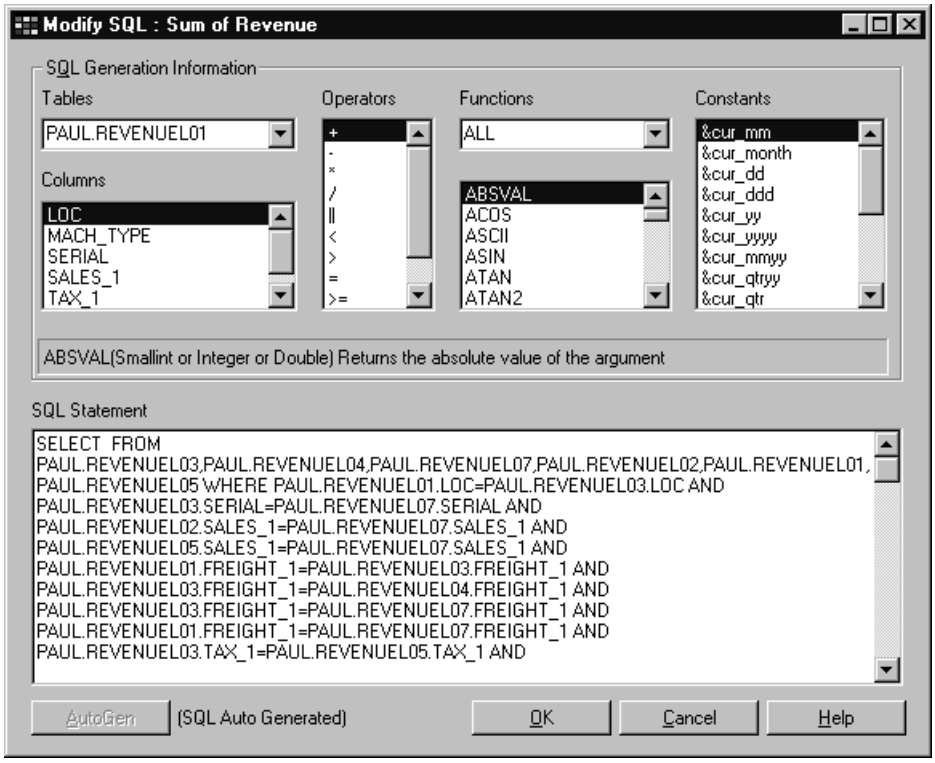


Figure 19. The Modify SQL window

To regenerate a SELECT statement:

1. Click **SQL**.

   The Modify SQL window opens.
2. Select a table name from the **Tables** list.
3. Double-click a column name in the **Columns** list.

   The fully qualified name for the column (table name and column name) is displayed in the **SQL Statement** field.
4. Double-click an item from the **Operators** list or the **Functions** list.

   The item is displayed in the **SQL Statement** field.
5. Double-click an item from the **Constants** list or the **Columns** list, or type a value in the **SQL Statement** field.
6. Repeat steps 2 through 5 until you complete the expression.

Joining tables

You can join data from two tables that share a common column and store the data in your business view table. If you selected multiple tables in the Create Business View window, you must include each table in at least one join. You can join tables if the business view status is development or test. You can join tables automatically or manually.

To automatically join all columns with the same name, click **AutoJoin**. All columns with the same name are joined.

For example, Fred wants to join the columns in his tables. When he clicks **AutoJoin**, the joins are displayed on the SQL page, as shown in Figure 20.

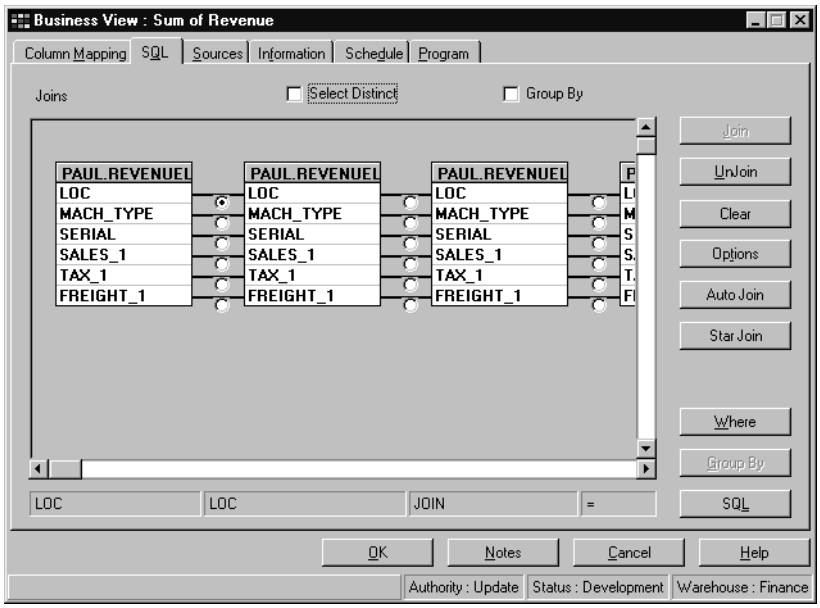


Figure 20. Columns that are joined using AutoJoin

To manually join two tables, select two columns in the **Joins** area and click **Join**.

A line is drawn between the two columns. A radio button is displayed in the line. When you click the radio button, information about the join is displayed at the bottom of the page, and the **UnJoin** button is available.

Repeat this procedure for each additional join that you want to create.

To break a join between tables:

1. In the **Joins** area, click the radio button in the line that represents the join.
2. Click **UnJoin**.

   The join is broken.

To clear all joins, click **Clear**.

Completing a selection WHERE clause

You define a WHERE clause to specify a condition that rows from the source column must meet to be included in the business view. By default, Visual Warehouse connects each clause of a WHERE clause with an AND operator and processes all clauses in random order.

To process specific clauses of a WHERE clause in a certain order, you must enter the clauses on the same line and connect them with OR operators. You can use one of two methods to do this:

- Insert an OR operator between clauses in the **Where** field.
- Select a clause from the **Where Clause** list and click **Edit**. In the Edit Where Clause window, type OR after the selected clause, then type the next clause that you want to process.

When you are finished modifying your WHERE clause, click **OK**.

To define a WHERE clause:

1. Click **Where**.

   The Selection Where Clause window opens.
2. Select a table name from the **Tables** list.
3. Double-click a column name in the **Columns** list.

   The fully qualified name for the column (table name and column name) is displayed in the **Where** field.
4. Double-click an item in the **Operators** list or **Functions** list.

   The item is displayed in the **Where** field.
5. Double-click an item in the **Constants** list or **Columns** list, or type a value in the **Where** field.
6. Repeat steps 2 through 5 until you have completed the expression.
7. Click **Add**.

   The WHERE clause is displayed in the **Where Clause** list.

Generating a GROUP BY clause

You generate a GROUP BY clause to combine rows that have the same value in one or more columns.

To generate a GROUP BY clause, select the **Group By** check box. Visual Warehouse generates a GROUP BY clause for all the columns that are not calculated with an aggregate function or constant.

To change the GROUP BY clause, click **SQL** and type over the statement in the **SQL Statement** field.

Completing a HAVING clause

You define a HAVING clause to limit values in the intermediate table that is defined in the GROUP BY statement. By default, Visual Warehouse connects each HAVING clause with an AND operator and processes all clauses in random order.

To process specific HAVING clauses in a certain order, you must enter the clauses on the same line and connect them with OR operators. You can use one of two methods to do this:

- Insert an OR operator between clauses in the **HAVING** field.
- Select a clause from the **Having Clause** list and click **Edit**. In the **Edit Group By Clause** window, type OR after the selected clause, then type the next clause that you want to process.

When you finish making changes to your HAVING statement, click **OK**.

To define a HAVING clause:

1. Click **Group By**.

   The Group By window opens.
2. Select a table name from the **Tables** list.
3. Double-click a column name in the **Columns** list.

   The fully qualified name for the column (table name and column name) is displayed in the **Having** field.
4. Double-click an item in the **Operators** list or the **Functions** list.

   The item is displayed in the **Having** field.
5. Double-click an item in the **Constants** list or **Columns** list, or type a value in the **Having** field.
6. Repeat steps 2 through 5 until you have completed the expression.
7. Click **Add**.

   The HAVING clause is displayed in the **Having Clause** list.
8. Click the **Information** tab. The Information page is displayed.

Providing information about your business view

You use the Information page to define your business view to your agent site and to define information about your warehouse table.

Figure 21 shows an example of a completed information page. In this example, Fred provides information about the agent site that runs his business view (AS/400 Agent) and about the target table in the warehouse.

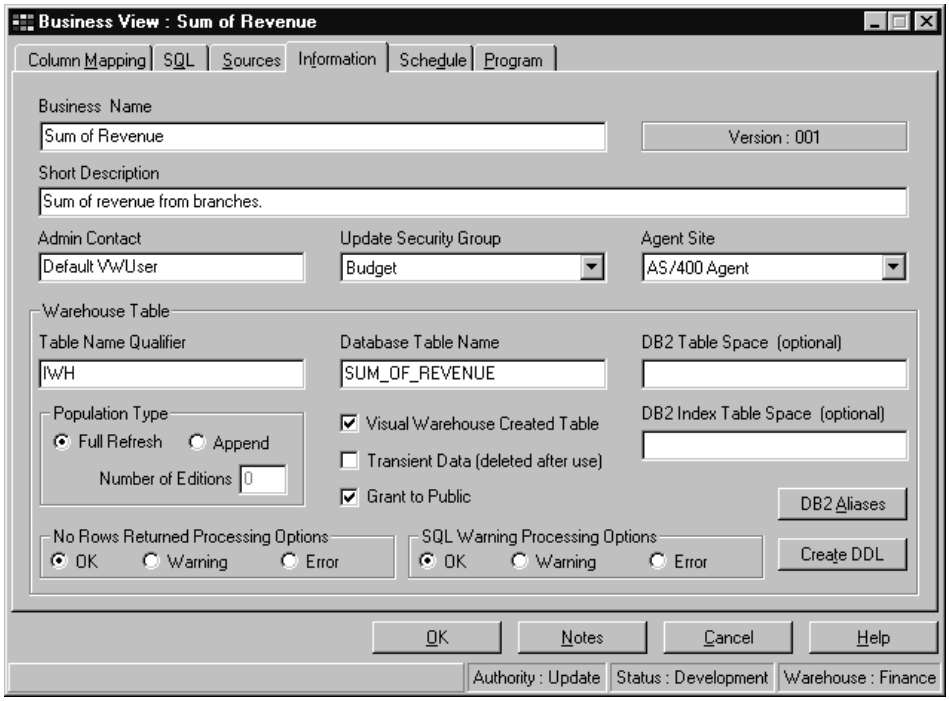


Figure 21. An example of a completed Information page

If you are using DB2 for MVS as a source or as a target, see "Using DB2 for MVS as a source or a target" on page 116 before you complete this page.

Defining your business view to your agent site

To define your business view to your agent site:

1. Select the name of your agent site from the **Agent Sites** list.
2. Click **OK**.

The agent sites lists in both the Information Resource and Business View notebooks contain an entry for the default agent site. The default agent site

takes precedence over remote agent sites. To prevent Visual Warehouse from using the default agent site, you must ensure that the default agent site is not selected for both your information resource and your business view. If the information resource and the business view are both configured to use the default agent site, you must remove the default agent site from the **Selected** list of either the information resource or the business view.

Providing information about the warehouse table

To provide information about your warehouse table, enter information into the fields that are listed in the Warehouse Table area:

1. In the **Table Name Qualifier** field, type the name of the library or collection that is associated with your table. If this library or collection does not exist on your AS/400 machine, it will be created for you.
2. In the **Database Table Name** field, type the name of the business view table. This table will be created in the library or collection that you entered in the **Table Name Qualifier** field.

   If you do not type a name for your business view table, Visual Warehouse will create a name for the table based on the name of your business view.
3. In the **Population Type** area, select the population type for the table:
   - To replace all the data in the business view table each time the business view runs, select **Full Refresh**.
   - To add a new business view edition to the business view table each time the business view runs, select **Append**.

     If you select **Append**, in the **Number of Editions** field, type the number of business view editions to save in the table. For example, if a business view runs once a month and you specify 12 editions, Visual Warehouse adds a new row of data to the table each month for one year. When the business view runs for the 13th time, the new data will replace the data in the first edition. The maximum number of editions is 9999.

     If you specify 0 as the number of editions, Visual Warehouse will append the data each time the business view runs, without replacing older data.

     If you specify one or more editions, Visual Warehouse adds the VWEDITION column, which contains a unique ID for the edition within Visual Warehouse, to the target table.

     For more information on selecting a population type for a business view, see the online help for the Information page.
4. If you do not want Visual Warehouse to create or drop the target table, clear the **Visual Warehouse Created Table** check box.

If you decide that Visual Warehouse should not create the target table, and your target table is in a DB2 for OS/400 database, ensure that the target table is being journaled.

5. If you want the data to be temporary, select **Transient Data**.
6. If you do not want each user that has access to the warehouse database to have access to the target table, clear the **Grant to Public** check box.
7. If you want to change the way Visual Warehouse handles SQL warning codes for this business view, select the appropriate radio button in the **No Rows Returned Processing Options** area:
   - **OK** to continue running the business view edition if an SQL warning code is issued. The business view edition will be displayed with a successful status icon in the Operations Work in Progress window (assuming that other errors do not occur).
   - **Warning** to continue running the business view edition if an SQL warning code is issued. The business view edition will be displayed with a warning status icon in the Operations Work in Progress window (assuming that other errors do not occur).
   - **Error** to stop running the business view edition if an SQL warning code is issued. The business view edition will be displayed with a failed status icon in the Operations Work in Progress window.
8. In the **SQL Warning Processing Options** area, select **OK**.

   The SQL Warning Processing Options determine the way that Visual Warehouse handles warnings when the agent finds no data to extract for a business view.

   When you run a business view, Visual Warehouse deletes any information about the business view that was created the last time the business view ran (unless you specify that Visual Warehouse saves this information).

   When you run a business view for the first time, there is no information for Visual Warehouse to delete. If you click **OK**, Visual Warehouse runs the business view, regardless of whether information about the business view exists. If you select one of the other options, the business view might fail.

   After you run the business view for the first time, you can change the SQL Warning Processing Options. The SQL Warning Processing Options are:

   - **OK**. The business view edition continues running if the agent finds no data to extract. The business view edition will be displayed with a successful status icon in the Operations Work in Progress window (assuming that other errors do not occur).
   - **Warning**. The business view edition continues running if the agent finds no data to extract or if the agent finds no data to delete in the target table. The business view edition will be displayed with a

warning status icon in the Operations Work in Progress window (assuming that other errors do not occur).

- **Error**. The business view edition stops running if the agent finds no data to extract. The business view edition will be displayed with a failed status icon in the Operations Work in Progress window.

9. If you want to create an alias for the target table in the warehouse, click **DB2 Aliases**.
10. If you want to view or edit the DDL for the Create Table statement, click **Create DDL**. The Create DDL window opens.

    You might want to edit the DDL for a number of reasons. For example, if you have columns that contain binary data, you must edit the DDL to add the FOR BIT DATA keyword. If you do not do this, the binary data is considered to be normal character data. The data will go through page translation on INSERT into the target column.

    When you are finished editing or viewing the DDL, click **OK**.
11. Click the **Schedule** tab. The Schedule page is displayed.

Scheduling a business view

Use the Schedule page to define a schedule for running your business view.

For example, Fred wants to schedule his business view to run every Fortnight. Figure 22 on page 57 shows the information that he selects on the Schedule page to run his business view every 14 days, or fortnight, beginning Sunday at 1 AM.

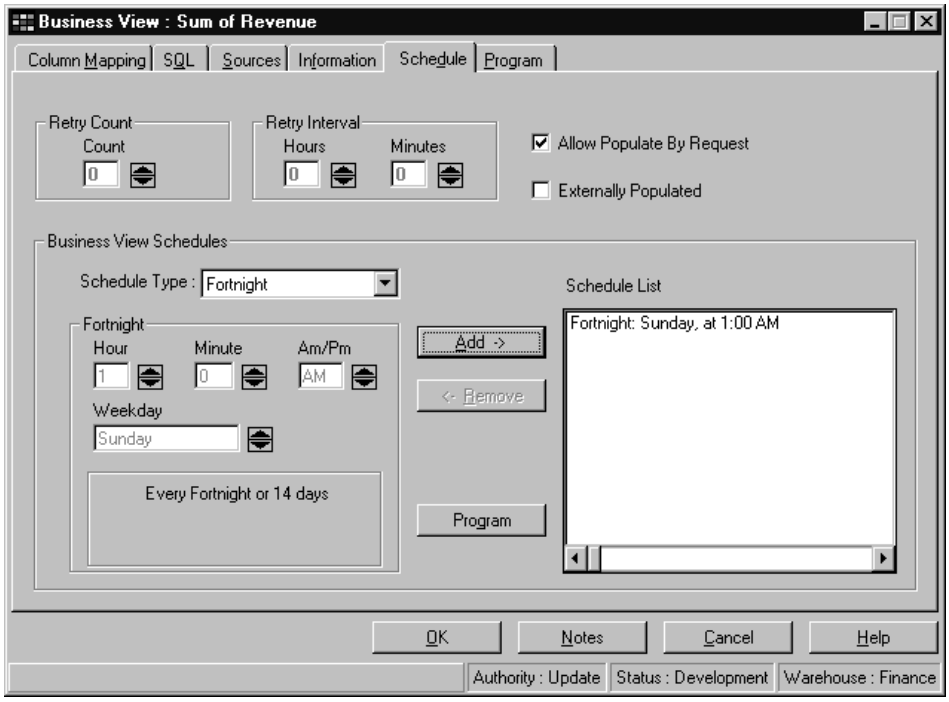


Figure 22. An example of a completed Schedule page.

You can define the schedule if the business view status is test or development.

To define a schedule:

1. Scroll through the **Schedule Type** list to select a schedule type based on how often you want to run a business view.

   Based on the type of schedule you choose, the box below the list displays certain selections.
2. Click **Add** to add the schedule to the business view. When the business view is promoted to production status, the data extract will run according to the schedule that you set.
3. If you want to run a business view at any time outside of the regular schedule, check the **Allow Populate by Request** box. Use the Operations Work in Progress window to run the business view.

   Tip: If a business view has test status, you can run it at any time, regardless of whether this box is checked.
4. If you want to populate or update the business view through external sources, such as IBM DataPropagator, check the **Externally Populated** box. If you make this selection, Visual Warehouse will not run your business view or launch any program that you designate to run your business view.

Any schedule that you define will be for informational purposes only. For more information, see the online help for the Schedule page in the Business View notebook.

5. In the **Count** field, select the number of times that you want to retry the processing of a business view edition that fails. Use the up and down arrows on the button to the right of the field to scroll through a list of available values.
6. In the **Hours** and **Minutes** fields, select the amount of time to elapse between retry attempts. Use the up and down arrows on the buttons to the right of the fields to scroll through lists of available values.
7. Click **OK**. The Business View notebook closes. You are finished defining your business view.

You can now promote your business view and run it. For information about promoting a business view, see “Appendix C. Changing the status of a business view” on page 103.

For information about running a business view manually, or for viewing the status of a scheduled business view, see the online help for the Operations Work in Progress window.

Chapter 8. Creating and defining a business view that uses a program

You can create business views that use a Visual Warehouse program to process data instead of using Visual Warehouse's SQL processing. You can include these business views in warehouses or subjects. This chapter describes the steps that are required to define a business view that uses a program and is included in a warehouse. For information about including a business view in a subject, see the online help.

When a Visual Warehouse program runs a business view, it can use the sources and targets that are defined to the business view. In order for a Visual Warehouse program to use these sources and targets, they must be defined as sources and targets in the Visual Warehouse program.

Before you can create a business view that your AS/400 agent will run, you must open the warehouse that is to contain it.

To open the warehouse:

1. Select the **Warehouses** tab on the Visual Warehouse desktop.
2. Double-click the name of the warehouse that you want to open.

   The Business View List window opens.

After you select a warehouse to contain your business view, you are ready to create the business view.

To create a business view that uses a Visual Warehouse program:

1. Select **File** → **New** from the Business View List window.

   The Create Business View window opens.
2. In the **Business Name** field, type a name for your new business view. This field is required.

   The name can be up to 80 characters long and is case sensitive.

   In the English language version of Visual Warehouse, the name cannot contain & as the first character. Other naming restrictions might vary depending on the language.

   The business view initially specifies a warehouse target table that is based on the name of the business view. You can change the table to another target table in the warehouse or modify the CREATE statement for the table.

3. From the **Program Group** list, select the program group that contains the Visual Warehouse program.
4. From the **Program Name** list, select the Visual Warehouse program that you want to assign to your business view.
5. Depending on the type of program you plan to run, select the name of an information resource or warehouse from the **Select Source** list. If your program does not require a source, do not make a selection.
6. From the **Available Tables** list, select one or more tables to include and click **Add**.

   The tables that you selected are displayed in the **Selected Tables** list.

   To remove a table from the **Selected Tables** list, select the table and click **Remove**.

   To remove all tables from the **Selected Tables** list, click **Clear**.
7. Click **OK**.

   The Business View notebook opens.

After you create the business view, you are ready to define it to use a Visual Warehouse program. You use the Business View notebook to define your business view.

To save the information that you enter into the Business View notebook at any time, click **OK**. The information in the Business View notebook is saved, and the notebook closes.

To open the notebook again, double-click on the business view icon in the Business View list.

Depending on the type of business view that you are creating, you might not need to provide information for the Column Mapping page, the SQL page, a portion of the Information page, and the Program page. However, you should review the information that is provided about these pages in the following sections to ensure that your business view definition meets its intended purpose.

Specifying source column definitions

When you define a business view that uses a program, this task is optional.

You use the Column Mapping page to add data columns from your information resource. The information that you enter on this page is the same information that you enter for a business view that uses SQL processing. For information about using this page, see “Specifying source column definitions” on page 44.

When you finish entering information on this page, click the **SQL** tab. The SQL page is displayed.

Editing SQL

When you define a business view that uses a Visual Warehouse program, the tasks you would normally complete using the SQL page are optional.

If your business view has sources and a target table, Visual Warehouse generates a default SQL statement.

Because you are using a Visual Warehouse program to manage the table for your business view, you do not need to edit the SQL statement. However, if you want Visual Warehouse to pass a SELECT statement to the Visual Warehouse program at run time, you can type the SELECT statement to use. You must specify that the Visual Warehouse program use the SELECT statement.

The steps you take to edit a SELECT statement for a business view that runs a program are the same as the steps you take to edit a business view that uses SQL processing. For information about using this page, see “Editing SQL” on page 48.

When you finish entering information on this page, click the **Information** tab. The Information page is displayed.

Providing information about your business view

Use the Information page to define your business view to your agent site and to define information about your warehouse table, if your business view uses one. You must define your business view to your agent site.

If you are using DB2 for MVS as a source or as a target, see “Using DB2 for MVS as a source or a target” on page 116 before you complete this page.

Defining your business view to your agent site

To define your business view to your agent site:

1. Select the name of your agent site from the **Agent Sites** list.
2. Click **OK**.

The agent sites lists in both the Information Resource and Business View notebooks contain an entry for the default agent site. The default agent site

takes precedence over remote agent sites. To prevent Visual Warehouse from using the default agent site, you must ensure that the default agent site is not selected for both your information resource and your business view. If the information resource and the business view are both configured to use the default agent site, you must remove the default agent site from the **Selected** list of either the information resource or the business view.

If your business view uses a warehouse table, you must define information about the table. Otherwise, you are finished defining this page.

Providing information about the warehouse table

The steps you take to define information about a warehouse table for a business view that runs a program are the same as the steps you take to define a business view that uses SQL processing. For information about completing this task, see "Providing information about your business view" on page 53.

When you finish entering information on this page, click the **Schedule** tab. The Schedule page is displayed.

Scheduling a business view

Use the Schedule page to define a schedule for running your business view.

You also use this page to specify how your Visual Warehouse program will run.

For a business view to appear in the Operations Work in Progress window, you must associate a schedule with the business view, or you must select the **Allow Populate by Request** check box on this page.

You can define the schedule for a business view if the business view status is test or development.

To define a schedule for a business view:

1. Scroll through the **Schedule Type** list to select a schedule type based on how often you want to run a business view.

   The box below the list displays certain selections based on the type of schedule that you select.
2. Click **Add** to add the schedule to the business view.

   When the business view is promoted to production status, the data extract will run according to the schedule that you set.

3. If you want to run a business view at any time outside of the regular schedule, check the **Allow Populate by Request** box. Use the Operations Work in Progress window to run the business view.

   If a business view has test status, you can run it at any time, regardless of whether this box is checked.
4. If you want to populate or update the business view through external sources such as IBM DataPropagator, check the **Externally Populated** box. If you check this box, Visual Warehouse will not start your program. Any schedule that you define will be for informational purposes only.
5. In the **Count** field, select the number of times that you want to retry the processing of a business view edition that fails. Use the up and down arrows on the button to the right of the field to scroll through a list of available values.
6. In the **Hours** and **Minutes** fields, select the amount of time to elapse between retry attempts. Use the up and down arrows on the buttons to the right of the fields to scroll through lists of available values.

You can schedule a Visual Warehouse program to run after your business view edition successfully runs. Depending on the schedule type that you select, you can create one of two types of Visual Warehouse programs:

The conditional cascade Visual Warehouse program
: After a business view runs, this type of Visual Warehouse program determines whether a condition that you specified was met. If the condition was met, the program returns a 0 return code. The 0 return code initiates a cascade that starts the next business view.

The post-processing Visual Warehouse program
: This Visual Warehouse program performs an action that you want to occur after a business view runs. This is the default Visual Warehouse program type.

For example, Fred wants to create a conditional cascade program for his business view. He selects **Conditionally Starts** from the **Schedule Type** list and clicks **Program**. The Cascade Program window opens. He then types information about his program and information about the table that is to receive the output of his program, as shown in Figure 23 on page 64. The table, FINANCE.RATE, is located in the Finance warehouse.

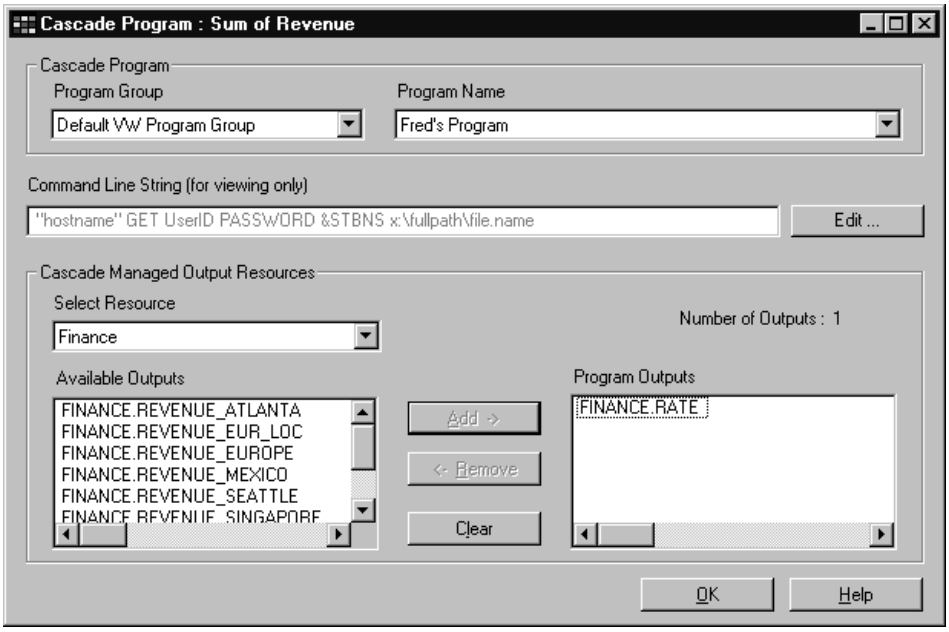


Figure 23. An example of a conditional cascade program definition

To create a conditional cascade Visual Warehouse program:

1. From the **Schedule Type** list, select **Conditionally Starts**, and click **Program**.

   The Cascade Program window opens.
2. From the **Program Group** list, select the program group that your program belongs to.
3. From the **Program Name** list, select your Visual Warehouse program.
4. From the **Select Resource** list, select the information resource or warehouse that receives your program's output.

   The list of tables in the resource is displayed in the **Available Sources** list.
5. Select one or more tables from the **Available Sources** list.

   The table moves to the **Visual Warehouse Program Outputs** list.
6. Click **OK**.

   The Cascade Program window closes.

   On the **Schedule** page, the **Conditionally Starts** list is displayed.
7. From the **Conditionally Starts** list, select the business view that you want the conditional cascade to start and click **Add**.
8. Click the **Program** tab. The Program page is displayed.

To run a conditional cascade program, you must promote its business view to production status. You must also promote to production status the business view that the cascade program will conditionally start. If you do not promote the second business view to production status, it will not run, even if the conditional cascade is successful.

If you assign a conditional cascade Visual Warehouse program to a business view, and then remove the Conditionally Starts schedule type, the program will run as a post-processing Visual Warehouse program.

For more information about the conditional cascade program, see "Appendix E. Determining the status of a conditional cascade" on page 109.

To create a post-processing Visual Warehouse program:

1. If you want to attach a schedule to the business view, select a schedule type other than **Conditionally Starts** from the **Schedule Type** list.

   You are not required to attach a schedule type for a business view that uses a post-processing program. For example, you might create a business view that you run manually.

   Any program that uses a schedule type other than **Conditionally Starts** is a post-processing program, including programs that do not use schedule types.
2. Click **Program**.

   The Cascade Program window opens.
3. From the **Program Group** list, select the program group that your program belongs to.
4. From the **Program Name** list, select your Visual Warehouse program.
5. From the **Select Resource** list, select the information resource or warehouse that receives your program's output. The list of tables in the resource is displayed in the **Available Sources** list.
6. Select one or more tables from the **Available Sources** list.

   The table moves to the **Visual Warehouse Program Outputs** list.
7. Click **OK**.
8. If you want to select a different program than the one you selected when you created the business view, click the **Program** tab. The Program page is displayed.
9. If you want to keep the program that you selected when you created the business view, then you are finished defining your business view.

   You can now promote your business view and run it.

To run a post-processing program, you must promote its business view to production status.

Defining a business view that uses a program

For information about promoting a business view, see “Appendix C. Changing the status of a business view” on page 103.

For information about running the business view manually, or for viewing the status of a scheduled business view, see the online help for the Operations Work in Progress window.

Working with a program definition

Use the Programs page to override, change, or remove the program definition that you are assigning to your business view.

Overriding default system parameters

Before you can use a program successfully, you must define the parameters that the program passes.

You can use the Program page to override the parameter values that the program will pass. Any changes you make to a Visual Warehouse program through the Program page override the default program definition for this business view. However, these changes do not affect the default Visual Warehouse program as it is defined in the Program notebook.

To illustrate, you assign a program called XPY to a business view. You want to change the value of the Remote Hostname parameter for this business view only. The default value for the Remote Hostname parameter, as it is defined on the Parameters page of the Program notebook, is doc.sd.rt.com. You use the Program page to override the value for the Remote Hostname parameter and change it to prat.sd.rt.com.

Now, suppose you assign program XPY to another business view. The value for the Remote Hostname parameter for the program assigned to the new business view is the default that is defined on the Parameters page of the Program notebook, doc.sd.rt.com.

To override a parameter value, you must use the Program page to edit the **Parameters List**.

Fred has assigned default values to each of the parameters in the Visual Warehouse program, Fred’s Program. Suppose he wants to override the value of the Remote UserID parameter for a particular business view. He wants to change it from FRED to JIM. He also wants to override the Remote password parameter value.

To override a default parameter value:

1. Click **Edit**

   The Program Parameters window opens..

   The **Parameters List** is displayed.
2. Double-click the Parameter Value of the Parameter Name that you want to define.

   You can now edit the **Parameter Value** field.
3. Clear away the generic parameter value in the **Parameter Value** field.

   For example, for the Remote UserID parameter, Fred clears away the parameter value `FRED`. He also clears away the value for the Remote password parameter.
4. Type the correct text in the **Parameter Value** field.

   For example, for the Remote UserID parameter, Fred types `JIM` in the **Parameter Value** field. He then changes the value of the Remote password parameter from `derf` to `MIJ`.
5. When you are finished defining your program parameters, click **Show**. A new command line string is displayed in the **Command Line String (for viewing only)** field.

   Figure 24 shows an example of the Program Parameters window for Fred's business view, after he has changed his parameter values and clicked **Show**.

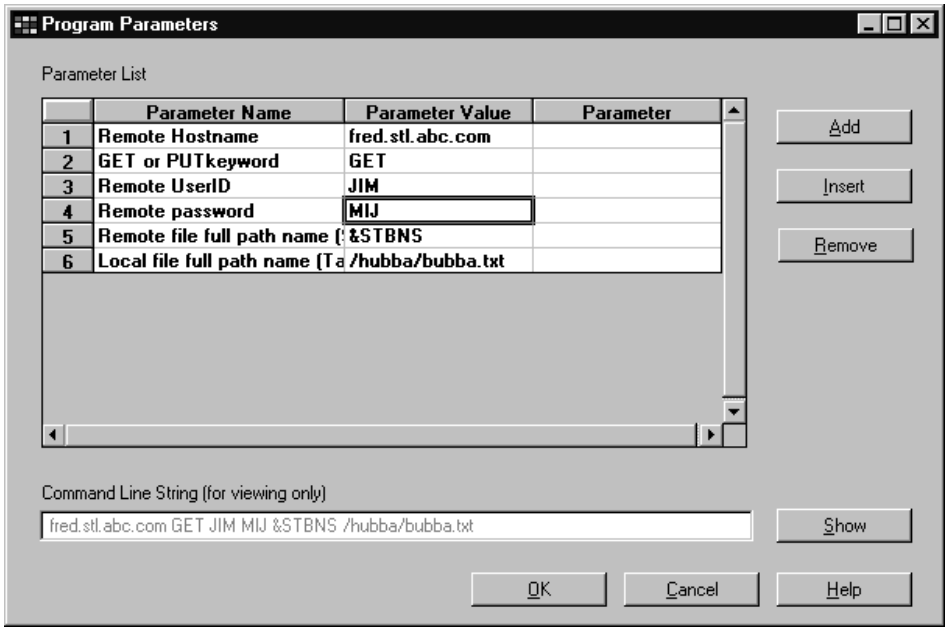


Figure 24. An example of new value definitions for system parameters

Selecting a different program

You can also use the Program page to select a different Visual Warehouse program than the one you initially specified when you created the business view. If you change the Visual Warehouse program, the selected outputs for the program do not change. You must manually change the selected output to reflect the new program.

Detaching a program

You can also use this page to detach a Visual Warehouse program from a business view. The business view will then use the SQL that is defined on the SQL page.

For information about completing these tasks, see the online help.

When you are finished entering data in the business view notebook, click **OK**. Your business view is defined to Visual Warehouse. You can now promote your business view and run it.

For information about promoting a business view, see “Appendix C. Changing the status of a business view” on page 103.

For information about running the business view manually, or for viewing the status of a scheduled business view, see the online help for the Operations Work in Progress window.

Chapter 9. Maintaining the AS/400 agent

This chapter describes other tasks that you might want to perform when you work with the AS/400 agent.

Verifying that the AS/400 agent daemon is running

Occasionally, you might want to verify that the AS/400 agent daemon that you started is still running. For example, you use the AS/400 agent throughout your work week, then go home for the weekend. When you return the following Monday, you should verify that the agent daemon is active before you initiate a new agent process.

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

1. At an AS/400 command line, enter `WRKACTJOB`. The active jobs are displayed.
2. Look for the function PGM-IWHVWD that is associated with the user ID you used when you started the daemon. If the function is not displayed, the agent is inactive.

Stopping the AS/400 agent daemon

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

To stop the agent daemon, enter `ENDVW` at an AS/400 command line.

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

Uninstalling the AS/400 agent

To delete the agent code, enter the following delete command at an AS/400 command line:

```
DLTLICPGM LICPGM(5639VW5)
```

The delete command:

- Removes the SRVTBLE entries.
- Removes the IFS directory.
- Removes the /QIBM/ProdData/IWH subdirectory.

- Ends the daemon.
- Deletes the product.

Appendix A. The sample Visual Warehouse programs

This appendix describes the sample Visual Warehouse programs that are included with the AS/400 agent:

- VWPLOADR
- VWPLOADI
- VWPRCPY
- VWPFTP

"Appendix B. Examples of sample Visual Warehouse program definitions" on page 95 provides examples of the information that you provide when you define these programs to Visual Warehouse.

VWPLOADR

The VWPLOADR program loads data from a flat file into a DB2 table. The load operation completely replaces the existing data in the table.

The source code for this program can be found in QIWH/VWPCSRC.VWPLOADR.

Prerequisites

To use this program, you must have the following PTFs applied to the AS/400 agent site:

Command PTF:	5769SS100	VRM420	SF46911
Code PTF:	5769SS100	VRM420	SF46976
Maintenance PTF:	5769SS100	VRM420	SF49466

These PTFs provide the AS/400 CPYFRMIMPF and CPYTOIMPF commands (LOAD and EXPORT). These are the commands that makes the VWPLOADR program work. These PTFs also install the online help for these commands.

Security

The user profile under which this program and the Visual Warehouse agent run must have at least read/write authority on the table that is to be loaded.

Limitations

The following list of limitations apply to the VWPLOADR program. For information about the limitations of the CPYFRMIMPF command, see the restrictions section of the online help for the CPYFRMIMPF command. To view the online help for this command, type CPYFRMIMPF on the AS/400 command line and press F1.

- This implementation of VWPLOADR differs from VWPLOADR on other platforms. Specifically, it does not delete all records loaded if the load operation fails for some reason.

  Normally, this program replaces everything in the target table each time it is run, and automatically deletes records from a failed run. However, if the load operation fails. you should avoid using data in the target table. If there is data in the target table, it will not be complete.
- The default behavior for VWPLOADR is to tolerate all recoverable data errors during LOAD (ERRLVL(*NOMAX)).

  To override this behavior, include the `ERRLVL(n)` keyword in the fileMod string parameter, where `n` = the number of permitted recoverable errors.

  You can find more information about the ERRLVL keyword in the online help for the CPYFRMIMPF command.

For information about the fileMod string parameter, see “Parameters”.

Parameters

Because this program runs on the AS/400 system under the same user profile that the Visual Warehouse agent runs under, the database, user ID, and password parameters that are required by VWPLOADR on other platforms are not required on the AS/400. As a result, the parameter list is much smaller that the one described in the online help for VWPLOADR.

The following list describes that parameters that VWPLOADR uses on the AS/400 agent site:

Source file name

Acceptable source files for the AS/400 implementation of VWPLOADR are AS/400 QSYS source file members or stream files in Integrated File System (IFS), the root file system.

When you define your parameter value, you reference files in the root file system by their full path names. For example:

```
/yourdirectory/yoursubdirectory/yourfile.yourext
```

You can also use the root file naming conventions for QSYS members:

```
/qsys.lib/yourlib.lib/yourfile.file/yourmember.mbr
```

Attention: You must define the full path name for VWPLOADR source file name parameter, beginning with the slash (/) character. This is how the program knows whether the input is a stream file or a QSYS file member.

When you define your parameter value, you reference file members in the QSYS file system by QSYS member naming rules. For example:

```
YOURLIB/YOURFILE.YOURMEMBER
```

If the data is in the first member of the file (for example, when the file is created using the SQL CREATE TABLE command) you can omit the member name:

```
YOURLIB/YOURFILE
```

Tip: You can improve both performance and storage consumption by using QSYS file members instead of stream files. This is because CPYFRMIMPF makes a copy of the entire stream file to QRESTORE and then loads the copy into your table. See the online help for CPYFRMIMPF for more information on this.

Target table

Because they are DB2 tables, target tables can reside only in the QSYS file system.

When you define your parameter value, you can only reference target tables by QSYS member naming rules. For example:

```
YOURLIB/YOURTABLE.YOURMEMBER
```

Or, if the data is to be loaded into the first member of the file (for example, when the file is created using the SQL CREATE TABLE command), you can omit the member name:

```
YOURLIB/YOURFILE
```

FileMod String (optional)

This parameter is used to modify the file characteristics that the CPYFRMIMPF command expects the input file to have. If this parameter is omitted, all of the default values that the CPYFRMIMPF command expects are assumed to be correct.

Some of the default characteristics of the input file are:

- The file is comma delimited.
- Strings and date/time values are enclosed in quotation marks.
- Date and time values are in ISO format.
- The decimal point is represented by a period character.

For more information on the default values for the CPYFRMIMPF command, see the AS/400 online help for the CPYFRMIMPF command.

The format for FileMod String is:

1. The string must contain valid CPYFRMIMPF command keywords. All of the valid keywords for the CPYFRMIMPF command are described in the online help for the command.
2. Each keyword must be followed immediately by its value. The value must be enclosed in parentheses.
3. Each keyword must be separated from the next keyword by a space.

For example, suppose the delimiter in your input file is a semicolon, the decimal point character is a comma, and the date format is European. The string you would enter for the FileMod parameter is:

```
FLDDLM(''';'') DECPNT(*COMMA) DATFMT(*EUR)
```

Attention: Certain parameters require that you enclose values in two single quote characters. For example, the FLDDLM command must have the values enclosed by two single quote characters. You must do this because Visual Warehouse generates the AS/400 CALL statement for VWPLOADR, in the form:

```
CALL PGM(QIWH/VWPLOADR) PARM('fromfile' 'totable' 'filemodstring')
```

Two single quotes together tells the AS/400 command line processor that your parameter value contains a single quote character. This prevents the command line processor from confusing your single quote character with the regular end-of-parameter marker.

Testing VWPLOADR from the command line

The call syntax on the AS/400 for VWPLOADR is:

```
CALL PGM(QIWH/VWPLOADR) PARM('fromfile' 'totable' 'filemodstring')
```

where:

fromfile
is the name of the file to get data from.

tofile is the name of the table to load.

filemodestring
is an optional parameter. It is a string containing options for the CPYFRMIMPF command.

For example:

```
CALL PGM(QIWH/VWPLOADR)
     PARM('TESTDATA/EURO.DAT' 'TESTDATA/TESTTBL.EUROPE'
          'FLDDLM('';'') DECPNT(*COMMA) DATFMT(*EUR)')
```

Tip: When you test VWPLOADR from the AS/400 command line, submit the job as a background job to get a spool file. If you run the job in the foreground, you will not get a spool file.

The command to run a job in the background is:

```
SBMJOB CMD(CALL PGM(QIWH/VWPLOADR) PARM(. . .))
```

When the Visual Warehouse program is run by the Visual Warehouse agent, it automatically runs in the background, so the spool file will always be created.

Traces and diagnostic information

VWPLOADR provides two kinds of diagnostic information:

- The return code, as documented in the VW online help
- The VWPLOADR trace

Attention: Successful completion of this program does not guarantee that the data was transferred correctly. For error handling that is more strict, use the ERRLVL parameter, as described in the Visual Warehouse online help.

Reading the VWPLOADR trace file

The VWPLOADR trace files are located in the Integrated File System in the `/QIBM/UserData/IWH` directory.

The VWPLOADR trace file has the following name format:

```
VWxxxxxxxx.VWPLOADR
```

where xxxxxxxx is the process ID of the VWPLOADR run that produced the file.

To view trace files from a workstation:

1. Use Client Access/400 to map your AS/400 root file system to a logical drive, or use FTP to copy the file to the workstation.

   For information about using Client Access/400, see “Viewing the VWPLOADR trace via Client Access/400”.

2. Open the trace file with a text editor to view the information.

Viewing the VWPLOADR trace via Client Access/400

To use Client Access/400 to map an AS/400 system to a logical drive on an NT workstation:

1. Set up a Client Access/400 connection to your AS/400 system over TCP/IP.
2. Open the Windows NT File Explorer.
3. From the Explorer menu, select **Tools → Map Network Drive**.
4. Type the pathname:

   ```
   \\hostname\.
   ```

   where `hostname` is the fully qualified TCP/IP host name of your AS/400 system.
5. Click **OK**.

Attention: If you use Client Access/400 to access the trace file, you must define the file extension `.VWPLOADR` to Client Access/400. Defining this extension allows Client Access/400 to translate the contents of files with this extension from EBCDIC to ASCII.

To define a file extension to Client Access/400:

1. From Windows NT, select **Start → Programs → IBM AS400 Client Access → Client Access Properties**.

   The Client Access notebook opens.
2. Click the **Network Drives** tab.
3. Type `.VWPLOADR` in the **File extension:** field.
4. Click **Add**.
5. Click **Apply**.
6. Click **OK**.

You should now be able to load the file into your favorite ASCII text editor or word processor.

AS/400 exceptions

If there was a failure of any of the system commands issued by VWPLOADR, then there will be an exception code recorded in the VWPLOADR trace file. To get an explanation for the exception:

1. At an AS/400 command line, enter `DSPMSGD RANGE(xxxxxxx)`, where xxxxxxx is the exception code. For example, you might enter `DSPMSGD RANGE(CPF2817)` at the AS/400 command line.

   The Display Formatted Message Text panel is displayed.
2. Select option **30** to display all information. A message similar to the following message is displayed:

   ```
   Message ID . . . . . :   CPF2817
   Message file . . . . . :   QCPFMSG
   Library  . . . . . :     QSYS
   ```

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

The second line in the VWPLOADR trace file contains the information you need to issue the WRKJOB command.

To view the spool file, you can cut and paste the name of the message file onto an AS/400 command line after the WRKJOB command and press Enter. View the spool file for the job to get additional information about any errors that you may have encountered.

VWPLOADI

The VWPLOADI program loads data from a flat file into a DB2 table. The load operation appends new data to the end of existing data in the table. In addition, prior to doing the append, VWPLOADI exports the original data to a backup file so that the original state of the data can be restored if the load operation fails.

The source code for this program can be found in QIWH/VWPCSRC.VWPLOADI.

Prerequisites

To use this program, you must have the following PTFs applied to the AS/400 agent site:

Command PTF:	5769SS100	VRM420	SF46911
Code PTF:	5769SS100	VRM420	SF46976
Maintenance PTF:	5769SS100	VRM420	SF49466

These PTFs provide the AS/400 CPYFRMIMPF and CPYTOIMPF commands (LOAD and EXPORT). These are the commands that makes the VWPLOADI program work. In addition, these PTFs install the online help for these commands.

Security

The user profile under which this program and the Visual Warehouse agent run must at least have the authority to create and delete objects and have read/write authority on the table that is to be loaded.

Limitations

The default behavior for VWPLOADI is to tolerate all recoverable data errors during LOAD (ERRLVL(*NOMAX)).

To override this behavior, include the `ERRLVL(n)` keyword in the fileMod string parameter, where `n` = the number of permitted recoverable errors.

For information about the fileMod string parameter, see "Parameters" on page 72.

For information about the limitations of the CPYFRMIMPF command, see the restrictions section of the online help for the CPYFRMIMPF command. To view the online help for this command, type CPYFRMIMPF on the AS/400 command line and press F1.

You can also find more information about the ERRLVL keyword in the online help for the CPYFRMIMPF command.

Parameters

Because this program runs on the AS/400 system under the same user profile that the Visual Warehouse agent runs under, the database, user ID, and password parameters that are required by VWPLOADI on other platforms are not required on the AS/400. As a result, the parameter list is much smaller that the one described in the online help for VWPLOADI.

The following list describes that parameters that VWPLOADI uses on the AS/400 agent site:

Source file name

Acceptable source files for the AS/400 implementation of VWPLOADI are AS/400 QSYS source file members or stream files in Integrated File System (IFS), the root file system.

When you define your parameter value, you reference files in the root file system by their full path names. For example:

```
/yourdirectory/yoursubdirectory/yourfile.yourext
```

You can also use the root file naming conventions for QSYS members:

```
/qsys.lib/yourlib.lib/yourfile.file/yourmember.mbr
```

Attention: You must define the full path name for VWPLOADI source file name parameter, beginning with the slash (/) character. This is how the program knows whether the input is a stream file or a QSYS file member.

When you define your parameter value, you reference file members in the QSYS file system by QSYS member naming rules. For example:

```
YOURLIB/YOURFILE.YOURMEMBER
```

If the data is in the first member of the file (for example, when the file is created using the SQL CREATE TABLE command) you can omit the member name:

```
YOURLIB/YOURFILE
```

Tip: You can improve both performance and storage consumption by using QSYS file members instead of stream files. This is because CPYFRMIMPF makes a copy of the entire stream file to QRESTORE and then loads the copy into your table. See the online help for CPYFRMIMPF for more information on this.

Target table

Because they are DB2 tables, target tables can reside only in the QSYS file system.

When you define your parameter value, you can only reference target tables by QSYS member naming rules. For example:

```
YOURLIB/YOURTABLE.YOURMEMBER
```

Or, if the data is to be loaded into the first member of the file (for example, when the file is created using the SQL CREATE TABLE command), you can omit the member name:

```
YOURLIB/YOURFILE
```

Backup file name

Because VWPLOADI appends data to an existing set of data in the target table, you need some form of error recovery. This parameter supplies the name of the backup file so that the existing data can be copied before the load of the new data is attempted. If there is a failure during the load operation, the backup file is used to restore the data to its original state.

The backup files used by the AS/400 implementation of VWPLOADI are AS/400 save files. You must provide save file names in the format library-name/save-file-name. For example:

```
YOURLIB/YOURSAVF
```

Attention: This parameter cannot be the name of an existing save file. It must be the name of a save file you want created on your behalf.

FileMod String (optional)

This parameter is used to modify the file characteristics that the CPYFRMIMPF command expects the input file to have. If this

parameter is omitted, all of the default values that the CPYFRMIMPF command expects are assumed to be correct.

Some of the default characteristics of the input file are:

- The file is comma delimited.
- Strings and date/time values are enclosed in quotation marks.
- Date and time values are in ISO format.
- The decimal point is represented by a period character.

For more information on the default values for the CPYFRMIMPF command, see the AS/400 online help for the CPYFRMIMPF command.

The format for FileMod String is:

1. The string must contain valid CPYFRMIMPF command keywords. All of the valid keywords for the CPYFRMIMPF command are described in the online help for the command.
2. Each keyword must be followed immediately by its value. The value must be enclosed in parentheses.
3. Each keyword must be separated from the next keyword by a space.

For example, suppose the delimiter in your input file is a semicolon, the decimal point character is a comma, and the date format is European. The string you would enter for the FileMod parameter is:

```
FLDDLM(''';'') DECPNT(*COMMA) DATFMT(*EUR)
```

Attention: Certain parameters require that you enclose values in two single quote characters. For example, the FLDDLM command must have the values enclosed by two single quote characters. You must do this because Visual Warehouse generates the AS/400 CALL statement for VWPLOADI in the form:

```
CALL PGM(QIWH/VWPLOADI) PARM('fromfile' 'totable' 'filemodstring')
```

Two single quotes together tells the AS/400 command line processor that your parameter value contains a single quote character. This prevents the command line processor from confusing your single quote character with the regular end-of-parameter marker.

Testing VWPLOADI from the command line

The call syntax on the AS/400 for VWPLOADI is:

```
CALL PGM(QIWH/VWPLOADR)
     PARM('fromfile' 'totable' 'backupfile' 'filemodstring')
```

where:

fromfile
is the name of the file to get data from.

tofile is the name of the table to load.

backupfile
is the name of the save file for the table.

filemodestring
is an optional parameter. It is a string containing options for the CPYFRMIMPF command.

For example:

```
CALL PGM(QIWH/VWPLOADR) PARM('TESTDATA/EURO.DAT'
  'TESTDATA/TESTTBL.EUROPE' 'TESTDATA/YOURSAVF'
  'FLDDLM('';'') DECPNT(*COMMA) DATFMT(*EUR)')
```

Tip: When you test VWPLOADI from the AS/400 command line, submit the job as a background job to get a spool file. If you run the job in the foreground, you will not get a spool file.

The command to run a job in the background is:

```
SBMJOB CMD(CALL PGM(QIWH/VWPLOADI) PARM(. . .))
```

When the Visual Warehouse program is run by the Visual Warehouse agent, it automatically runs in the background, so the spool file will always be created.

Traces and diagnostic information

VWPLOADI provides two kinds of diagnostic information:

- The return code, as documented in the VW online help
- The VWPLOADI trace

Attention: Successful completion of this program does not guarantee that the data was transferred correctly. For error handling that is more strict, use the ERRLVL parameter, as described in the Visual Warehouse online help.

Reading the VWPLOADI trace file

The VWPLOADI trace files are located in the Integrated File System in the `/QIBM/UserData/IWH` directory.

The VWPLOADR trace file has the following name format:

```
VWxxxxxxxx.VWPLOADI
```

where xxxxxxxx is the process ID of the VWPLOADI run that produced the file.

To view trace files from a workstation:

1. Use Client Access/400 to map your AS/400 root file system to a logical drive, or use FTP to copy the file to the workstation.

   For information about using Client Access/400, see "Viewing the VWPLOADI trace file via Client Access/400".

2. Open the trace file with a text editor.

Viewing the VWPLOADI trace file via Client Access/400

To use Client Access/400 to map an AS/400 system to a logical drive on an NT workstation:

1. Set up a Client Access/400 connection to your AS/400 system over TCP/IP.
2. Open the Windows NT File Explorer.
3. From the Explorer menu, select **Tools → Map Network Drive**.
4. Type the pathname:

   `\\hostname\.`

   where `hostname` is the fully qualified TCP/IP host name of your AS/400 system.
5. Click **OK**.

Attention: If you use Client Access/400 to access the trace file, you must define the file extension `.VWPLOADI` to Client Access/400. Defining this extension allows Client Access/400 to translate the contents of files with this extension from EBCDIC to ASCII.

To define a file extension to Client Access/400:

1. From Windows NT, select **Start → Programs → IBM AS400 Client Access → Client Access Properties**.

   The Client Access notebook opens.
2. Click the **Network Drives** tab.
3. Type `.VWPLOADI` in the **File extension:** field.
4. Click **Add**.
5. Click **Apply**.
6. Click **OK**.

You should now be able to load the file into an ASCII text editor or word processor.

AS/400 exceptions

If there was a failure of any of the system commands issued by VWPLOADI, then there will be an exception code recorded in the VWPLOADI trace file. To get an explanation for the exception:

1. At an AS/400 command line, enter DSPMSGD RANGE(xxxxxxx), where xxxxxxx is the exception code. For example, you might enter DSPMSGD RANGE(CPF2817) at the AS/400 command line.

   The Display Formatted Message Text panel is displayed.
2. Select option **30** to display all information. A message similar to the following message is displayed:

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

The second line in the VWPLOADI trace file contains the information you need to issue the WRKJOB command.

To view the spool file you can cut and paste the name of the message file on to an AS/400 command line after the WRKJOB command and press **Enter**. View the spool file for the job to get additional information about any errors that you may have encountered.

VWPRCPY

The VWPRCPY program transfers a text file from one system to another.

To transfer a text file, the program performs the following tasks:

1. The program builds a temporary file that contains FTP commands.
2. The program then redirects the standard input to the command file and issues the FTP command to move the text file. The program also redirects the FTP output messages from standard output to a file.
3. After the file is transferred, VWPRCPY deletes the command file for security reasons; the command file contains a user ID and password.

The source physical file that holds both the output and input data is QIWH/FTPCMD.

The command files are members that have the form CMxxxxxxxx, where xxxxxxxx is the process ID of the instance of VWPRCPY currently running. Because CMxxxxxxxx is deleted as soon as the job completes, it can only be seen while the program is running.

The output messages from FTP are members that have the form MSxxxxxxxx, where xxxxxxxx is the process ID of the instance of VWPRCPY currently running.

The source code for this Visual Warehouse program can be found in QIWH/VWPCSRC.VWPRCPY.

Security

Although the command file is deleted upon completion of the file transfer, the CMxxxxxxxx members are still a security exposure. As a result, you should restrict access to the QIWH library.

Limitations

The QIWH/FTPCMD source file can only hold records that are 240 bytes long. Because of this, the combined length of source and target path strings cannot be more than 221 bytes long. If you exceed this limit, the get/put command will not be written to the temporary FTP command file.

There are two ways around this limitation:

- You can create your own command file to do the transfer and use VWPFTP instead of VWPRCPY. VWPFTP is described on "VWPFTP" on page 88.
- You can drop and reallocate the FTPCMD file for longer records. This method is not recommended because you will not be able to view the contents of the FTP output members on the AS/400 (the limit is 240 bytes).

Performance considerations

VWPRCPY uses ASCII transfer type. To perform binary or EBCDIC file transfers, use VWPFTP. Using the ASCII type to transfer files makes VWPFTP generic so that it can get text files from and put text files to any operating system.

File naming conventions

Most file systems, except MVS and AS/400 QSYS, allow files to be retrieved using POSIX path and file name conventions.

To retrieve MVS files or AS/400 QSYS files using FTP/400, you must enter the file names following the conventions described below:

AS/400

QSYS files /QSYS.LIB/yourlib.LIB/yourfile.FILE/yourdata.MBR

For example:

```
/QSYS.LIB/QGPL.LIB/QCLSRC.FILE/BUBBA.MBR
```

MVS

- mid-level-qualifier.low-level-qualifier(member-name)

  For example:

  ```
  CLIST.CLIST(DEFAULT)
  ```

  gets the file CLIST.CLIST(DEFAULT)
- 'hlq.mlq.llq(member-name)'

  where:

  hlq = high-level-qualifier

  mlq = mid-level-qualifier

  llq = low-level-qualifier

  Attention: The single quotes are required on MVS when a fully qualified dataset name is used.

  For example:

  ```
  'BUBBA.CLIST.CLIST(DEFAULT)'
  ```

  gets the file BUBBA.CLIST.CLIST(DEFAULT)

Testing VWPRCPY from the command line

The call syntax on the AS/400 for VWPRCPY is:

```
CALL PGM(QIWH/VWPRCPY) PARM('host' 'cmd' 'uid' 'pwd' 'rmtf' 'locf')
```

where:

host is the fully qualified TCP/IP host name.

cmd is the FTP get or put command.

uid is the remote host user ID.

pwd is the remote host password.

rmtf is the remote file to copy from or to.

locf is the local file to copy from or to.

Attention: When you use fully qualified MVS file names, you must pass the remote file name parameter in triple single quotes. This preserves one set of

single quotes around the name, which is required for MVS to override the default high-level qualifier. For example, you would type:

```
CALL PGM(QIWH/VWPRCPY) PARM(. . . '''bob.fin.dat(checks)'''. . . )
```

Tip: When testing VWPRCPY from the AS/400 command line, submit the job as a background job to get a spool file. If you run the job in the foreground, you will not get a spool file.

The command to run a job in the background is:

```
SBMJOB CMD(CALL PGM(QIWH/VWPRCPY) PARM(. . .))
```

When the Visual Warehouse program is run by the Visual Warehouse agent, it automatically runs in the background, so the spool file is always created.

Traces and diagnostic information

VWPRCPY provides out three kinds of diagnostic information:

- The return code, as documented in the VW online help
- The FTP message log
- The VWPRCPY trace

FTP message logs are members in the QIWH/FTPCMD source physical file. They follow the naming convention MSxxxxxxxx, where xxxxxxxx is the process ID of VWPRCPY run that produced the file.

To view the FTP message logs, enter the following command at an AS/400 command line:

```
WRKMBRPDM QIWH/FTPCMD
```

Attention: Successful completion of this program does not guarantee that the data was transferred correctly. It only indicates that the FTP command ran. To ensure that the data was transferred correctly, you must check the FTP message logs for warnings and non-fatal errors.

For example, if the record length of your target physical file is too small, some records might get truncated. This information will show up in the FTP message log for that run.

Reading the VWPRCPY trace file

The VWPRCPY trace file can be found in the Integrated File System in the /QIBM/UserData/IWH directory.

The VWPRCPY trace file has the following name format:

```
VWxxxxxxxx.VWPRCPY
```

where xxxxxxxx is the process ID of the VWPRCPY run that produced the file.

To view trace files from a workstation:

1. Use Client Access/400 to map your AS/400 root file system to a logical drive or use FTP to copy the file to the workstation.

   For information about using Client Access/400 to map your AS/400 root file system to a logical drive, see "Viewing the VWPRCPY trace file via Client Access/400".

2. Open the trace file with a text editor.

Viewing the VWPRCPY trace file via Client Access/400

To use Client Access/400 to map an AS/400 system to a logical drive on an NT workstation:

1. Set up a Client Access/400 connection to your AS/400 system over TCP/IP.
2. Open the Windows NT File Explorer.
3. From the Explorer menu, select **Tools → Map Network Drive**.
4. Type the pathname:

   ```
   \\hostname\.
   ```

   where `hostname` is the fully qualified TCP/IP host name of your AS/400 system.
5. Click **OK**.

Attention: If you use Client Access/400 to access the trace file, you must define the file extension `.VWPRCPY` to Client Access/400. Defining this extension allows Client Access/400 to translate the contents of files with this extension from EBCDIC to ASCII.

To define a file extension to Client Access/400:

1. From Windows NT, select **Start → Programs → IBM AS400 Client Access → Client Access Properties**.

   The Client Access notebook opens.
2. Click the **Network Drives** tab.
3. Type `.VWPRCPY` in the **File extension:** field.
4. Click **Add**.
5. Click **Apply**.
6. Click **OK**.

You should now be able to load the file into an ASCII text editor or word processor.

AS/400 exceptions

If there was a failure of any of the system commands issued by VWPRCPY, then there will be an exception code recorded in the VWPRCPY trace file. To get an explanation for the exception:

1. At an AS/400 command line, enter DSPMSGD RANGE(xxxxxxx), where xxxxxxx is the exception code. For example, you might enter DSPMSGD RANGE(CPF2817) at the AS/400 command line.

   The Display Formatted Message Text panel is displayed.

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

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

The second line in the VWPRCPY trace file contains the information you need to issue the WRKJOB command.

To view the spool file you can cut and paste the name of the message file onto an AS/400 command line after the WRKJOB command and press Enter. View the spool file for the job to get additional information about any errors that you may have encountered.

VWPFTP

The VWPFTP transfers files from one system to another.

To transfer files from one system to another, the program uses a user-supplied command file that contains FTP batch commands to redirect the standard input to the command file and to invoke FTP in batch mode. The program also redirects the FTP output messages from standard output to a file.

The source physical file that holds the output messages is QIWH/FTPCMD.

Members that have the form MSxxxxxxxx contain the output messages from FTP.

The source code for this program can be found in QIWH/VWPCSRC.VWPFTP.

FTP command file

The FTP command file contains all of the commands that you FTP to be issued during batch execution. The following restrictions apply to the FTP command file:

- The FTP command file must be a member in a source physical file.
- Each command must be on its own line in the member.
- The only mandatory line is the first one, which contains the user ID and password separated by a space.

The last command in the file should be `quit`.

To create a command file:

1. Create a library from the AS/400 command line. For example, you might enter:
   ```
   CRTLIB FTPLIB
   ```
2. Create a source physical file from the AS/400 command line. For example:
   ```
   CRTSRCPF FTPLIB/FTPFILE
   ```
3. Use the STRSEU editor from the AS/400 command line to create the command file. For example:
   ```
   STRSEU SRCFILE(FTPLIB/FTPFILE) SRCMBR(FTPCOMMAND)
   ```
4. From STRSEU editor, type in your ftp commands. For example:
   ```
   remoteuid remotepwd
   namefmt 1
   type ascii
   get /qsys.lib/tmp.lib/data.file/bubba.mbr /usr/data/bubba.txt (replace
   quit
   ```

Attention: There are two different file naming conventions on the AS/400:

- UNIX-like file names (NAMEFMT 1)
- AS/400-QSYS-style file names (NAMEFMT 0)

If you are only moving files or members between one AS/400 system and another, you might consider using NAMEFMT 0. In all other cases, use NAMEFMT 1.

Here is an example of a NAMEFMT 0 FTP command file:

```
remoteuid
remotepwd
namefmt 0
type ebcdic
get tmp/data.bubba bubba/data.mystuff (replace
quit
```

For more information about using FTP on the AS/400 see *OS/400 TCP/IP Configuration and Reference V4R2*, SC41-5420.

Security

The user-supplied command file used by VWPFTP is a security exposure because it must contain a user ID and password. As a result, you should restrict access to the library that contains the command file.

Performance considerations

If you are moving text data between an ASCII-based machine and an EBCDIC-based machine, use ASCII as the file transfer type.

If you are moving data between two EBCDIC-based machines (for example, AS/400 to AS/400, or MVS to AS/400), use the FTP command `type ebcdic` to change the file transfer type to EBCDIC. This will save you the overhead of converting from EBCDIC to ASCII to EBCDIC.

For binary file transfers, use the FTP command `type binary`.

File naming conventions

Most file systems, except MVS and AS/400 QSYS, allow files to be retrieved using POSIX path and file name conventions.

To retrieve MVS files or AS/400 QSYS files using FTP/400, you must enter the file names following the conventions described below:

AS/400

/QSYS.LIB/yourlib.LIB/yourfile.FILE/yourdata.MBR

For example:

```
/QSYS.LIB/QGPL.LIB/QCLSRC.FILE/BUBBA.MBR
```

MVS

- mid-level-qualifier.low-level-qualifier(member-name)

  For example:

  ```
  CLIST.CLIST(DEFAULT)
  ```

  gets the file CLIST.CLIST(DEFAULT)
- 'hlq.mlq.llq(member-name)'

  where:

  hlq = high-level-qualifier

  mlq = mid-level-qualifier

  llq = low-level-qualifier

Attention: The single quotes are required on MVS when a fully qualified dataset name is used.

For example:

```
'BUBBA.CLIST.CLIST(DEFAULT)'
```

gets the file BUBBA.CLIST.CLIST(DEFAULT)

Attention:

- When you code your FTP command file, you must put MVS file names in double quotes. You must do this because MVS file names do not follow the POSIX file naming conventions. Using the two examples above, the entries in the FTP command file, using MVS file name, would be:

  ```
  get "CLIST.CLIST(DEFAULT)" /QSYS.LIB/QGPL.LIB/QCLSRC.FILE/BUBBA.MBR
  get "'BUBBA.CLIST.CLIST(DEFAULT)'" /QSYS.LIB/QGPL.LIB/QCLSRC.FILE/BUBBA.MBR
  ```

- When using the get command to transfer data to a file on the AS/400, it is recommended that you add the (REPLACE option to the get command. This will overwrite the previous version of the target file. For example:

  ```
  get d:\pcfile.txt /QSYS.LIB/QGPL.LIB/QCLSRC.FILE/BUBBA.MBR (REPLACE
  ```

Testing VWPFTP from the command line

The call syntax on the AS/400 for VWPFTP is:

```
CALL PGM(QIWH/VWPFTP) PARM('host' 'cmdfilename' )
```

where:

host is the fully qualified TCP/IP host name.

cmdfilename
is the fully qualified member name containing the ftp commands in the following format:

```
library/file.member
```

For example:

```
CALL PGM(QIWH/VWPFTP) PARM('bubba.ibm.com' 'ftplib/ftpfile.commands' )
```

Tip: When testing VWPFTP from the AS/400 command line, submit the job as a background job to get a spool file. If you run the job in the foreground, you will not get a spool file.

The command to run a job in the background is:

```
SBMJOB CMD(CALL PGM(QIWH/VWPFTP) PARM(. . .))
```

When the Visual Warehouse program is run by the Visual Warehouse agent, it automatically runs in the background, so the spool file is always created.

Traces and diagnostic information

VWPFTP provides out three kinds of diagnostic information:

- The return code, as documented in the VW online help
- The FTP message log
- The VWPFTP trace

FTP message logs are members in the QIWH/FTPCMD source physical file. They follow the naming convention MSxxxxxxxx, where xxxxxxxx is the process ID of VWPFTP run that produced the file.

To view the FTP message logs, enter the following command at an AS/400 command line:

```
WRKMBRPDM QIWH/FTPCMD
```

Attention: Successful completion of this program does not guarantee that the data was transferred correctly. It only indicates that the FTP command ran. To ensure that the data was transferred correctly, you must check the FTP message logs for warnings and non-fatal errors.

For example, if the record length of your target physical file is too small, some records may get truncated. This information will show up in the FTP message log for that run.

Reading the VWPFTP trace file

The VWPFTP trace file can be found in the Integrated File System in the /QIBM/UserData/IWH directory.

The VWPFTP trace file has the following name format:

```
VWxxxxxxxx.VWPFTP
```

where xxxxxxxx is the process ID of the VWPFTP run that produced the file.

To view trace files from a workstation:

1. Use Client Access/400 to map your AS/400 root file system to a logical drive, or use FTP to copy the file to the workstation.

   For information about using Client Access/400 to map your AS/400 root file system to a logical drive, see “Viewing the VWPFTP trace file via Client Access/400” on page 93.

2. Open the trace file with a text editor.

Viewing the VWPFTP trace file via Client Access/400

To use Client Access/400 to map an AS/400 system to a logical drive on an NT workstation:

1. Set up a Client Access/400 connection to your AS/400 system over TCP/IP.
2. Open the Windows NT File Explorer.
3. From the Explorer menu, select **Tools → Map Network Drive**.
4. Type the pathname:

   ```
   \\hostname\.
   ```

   where `hostname` is the fully qualified TCP/IP host name of your AS/400 system.
5. Click **OK**.

Attention: If you use Client Access/400 to access the trace file, you must define the file extension `.VWPFTP` to Client Access/400. Defining this extension allows Client Access/400 to translate the contents of files with this extension from EBCDIC to ASCII.

To define a file extension to Client Access/400:

1. From Windows NT, select **Start → Programs → IBM AS400 Client Access → Client Access Properties**.

   The Client Access notebook opens.
2. Click the **Network Drives** tab.
3. Type `.VWPFTP` in the **File extension:** field.
4. Click **Add**.
5. Click **Apply**.
6. Click **OK**.

You should now be able to load the file into your favorite ASCII text editor or word processor.

AS/400 exceptions

If there was a failure of any of the system commands issued by VWPFTP, there will be an exception code recorded in the VWPFTP trace file. To get an explanation for the exception:

1. At an AS/400 command line, enter `DSPMSGD RANGE(xxxxxxx)`, where `xxxxxxx` is the exception code. For example, you might enter `DSPMSGD RANGE(CPF2817)` at the AS/400 command line.

   The Display Formatted Message Text panel is displayed.

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

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

The second line in the VWPFTP trace file contains the information you need to issue the WRKJOB command.

To view the spool file you can cut and paste the name of the message file onto an AS/400 command line after the WRKJOB command and press Enter. View the spool file for the job to get additional information about any errors that you may have encountered.

VWPFTP error codes

The error coeds provided here supersede those given in the online documentation for VWPFTP.

PARAMETER ERROR = 8
The wrong number of parameters or an invalid parameter were passed on the call to VWPFTP.

FTP ERROR = 12
An error occurred during the setup for or batch execution of the FTP command. Check the VWxxxxxxxx.VWPFTP trace file and the FTP batch output messages, QIWH/FTPCMD.MSxxxxxxxx, for more information.

INTERNAL ERROR = 16
An unexpected problem occurred during processing. Check the VWxxxxxxxx.VWPFTP trace file for more information.

There is no error code 128 as documented in the online help.

Appendix B. Examples of sample Visual Warehouse program definitions

Before you can use a sample Visual Warehouse program, you must define it to Visual Warehouse. You use the Program notebook to define a program.

This appendix provides examples that you can use to help you define the sample programs to Visual Warehouse. These examples describe the information that you must enter on the Program page and the Parameters page of the Program notebook for the following sample programs:

- VWPLOADR
- VWPRCPY (GET)
- VWPRCPY (PUT)

Attention: This appendix only provides information for the Program and Parameters pages of the Program notebook. To define a program successfully, you must also supply information about the agent site where the program is stored. For general information about defining programs to Visual Warehouse, including information about adding an agent site to a program definition, see "Chapter 6. Defining a Visual Warehouse program that runs on OS/400" on page 33.

To open the Visual Warehouse program notebook:

1. From the Visual Warehouse desktop, select **Definitions → Visual Warehouse Programs**.

   The Programs window opens.
2. Select **File → New**.

Defining VWPLOADR

You use this program to load data from a flat file into a DB2 table.

Providing information about the program

When you define the Program page for VWPLOADR, ensure that you specify the following information:

- The name of the Program Group is **VW Sample Program Group**.

  To add a new program group, select a field in the **Program Group** list. Then type the name of the new group in the field. To change the name of an existing group, type the new name of the group over the old one.

- The Program Executable Type is **Executable**.
- The Fully Qualified Program Name is **QIWH/VWPLOADR**.

Figure 25 shows an example of the completed Program page:

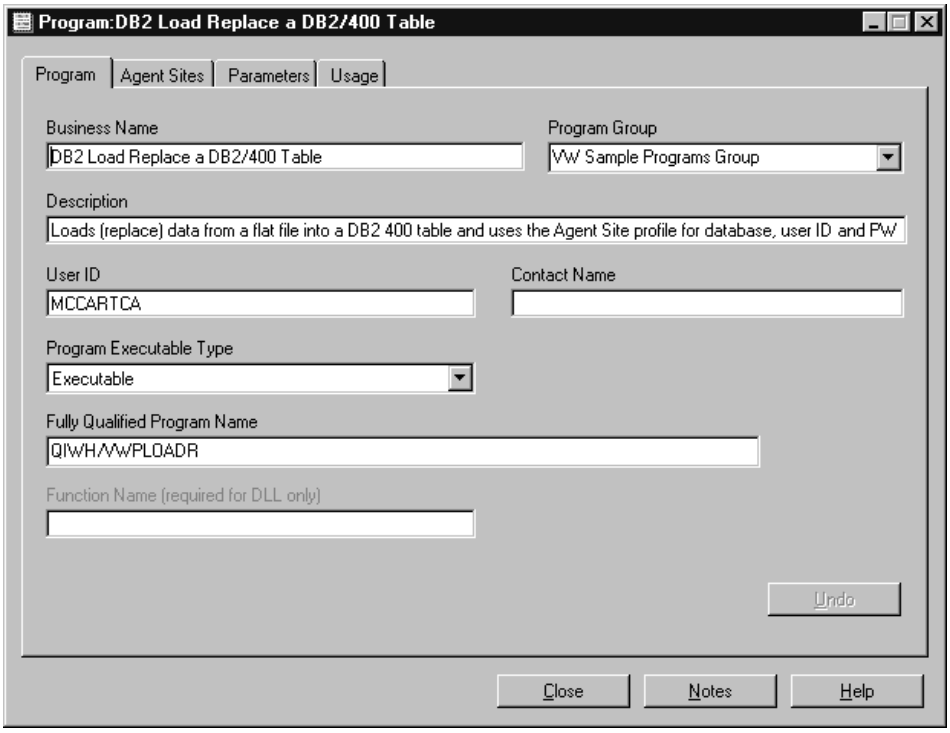


Figure 25. An example of a completed Program page for VWPLOADR

Setting default parameters

You use the Parameters page in the Program notebook to set the default parameters for a program.

To set the default parameters for VWPLOADR, add the following parameters to the **Parameter List**:

- Source File Name
- Target Table Name

Figure 26 on page 97 shows an example of a completed Parameters page:

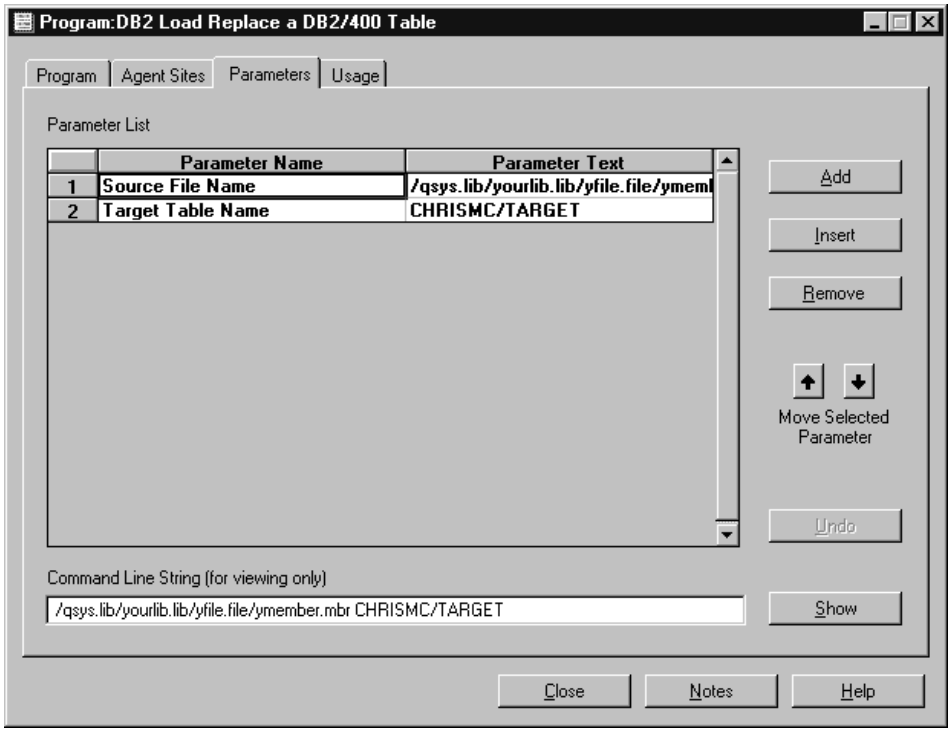


Figure 26. An example of a completed Parameters page for VWPLOADR

For information about adding parameters, see "Setting parameters" on page 36.

Defining VWPRCPY (GET)

You use this program to transfer a file to your local system.

Providing information about the program

When you define the Program page for VWPRCPY (GET), ensure that you specify the following information:

- The name of the Program Group is **VW Sample Program Group**.

  To add a new program group, select a field in the **Program Group** list. Then type the name of the new group in the field. To change the name of an existing group, type the new name of the group over the old one.
- The Program Executable Type is **Executable**.
- The Fully Qualified Program Name is **QIWH/VWPRCPY**.

Figure 27 shows an example of the completed Program page:

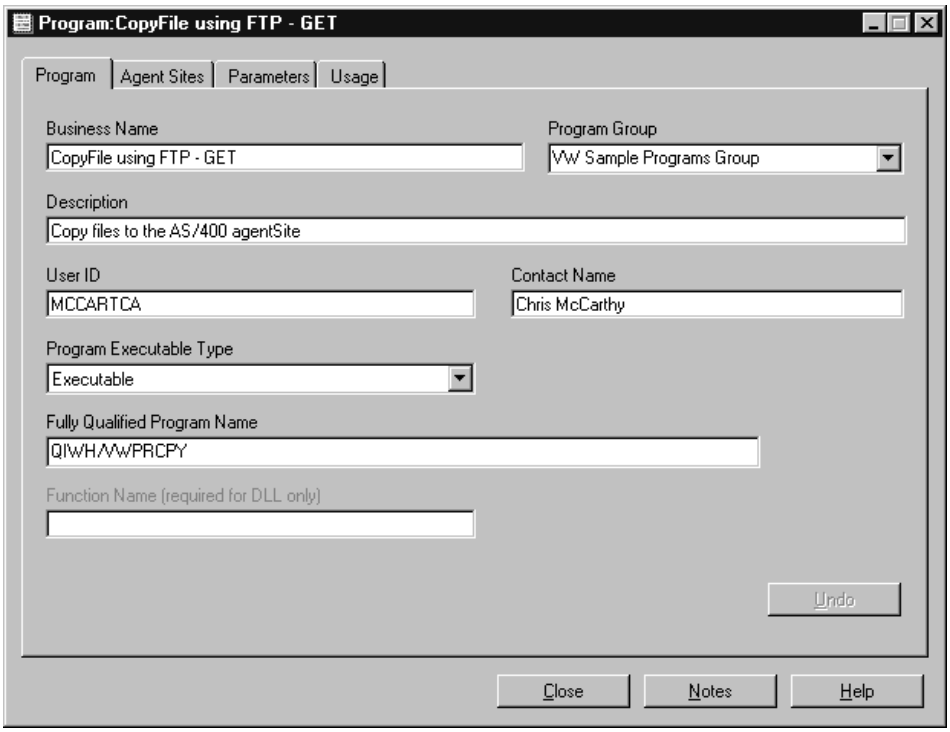


Figure 27. An example of a completed Program page for VWPRCPY (GET)

Setting default parameters

You use the Parameters page in the Program notebook to set the default parameters for a program.

To set the default parameters for VWPRCPY (GET), add the following parameters to the **Parameter List**:

- Remote Hostname
- GET or PUT keyword (define the value as GET)
- Remote UserID
- Remote password
- Remote file full path name (Source)
- Local file full path name (Target)

Figure 28 on page 99 shows an example of a completed Parameters page:

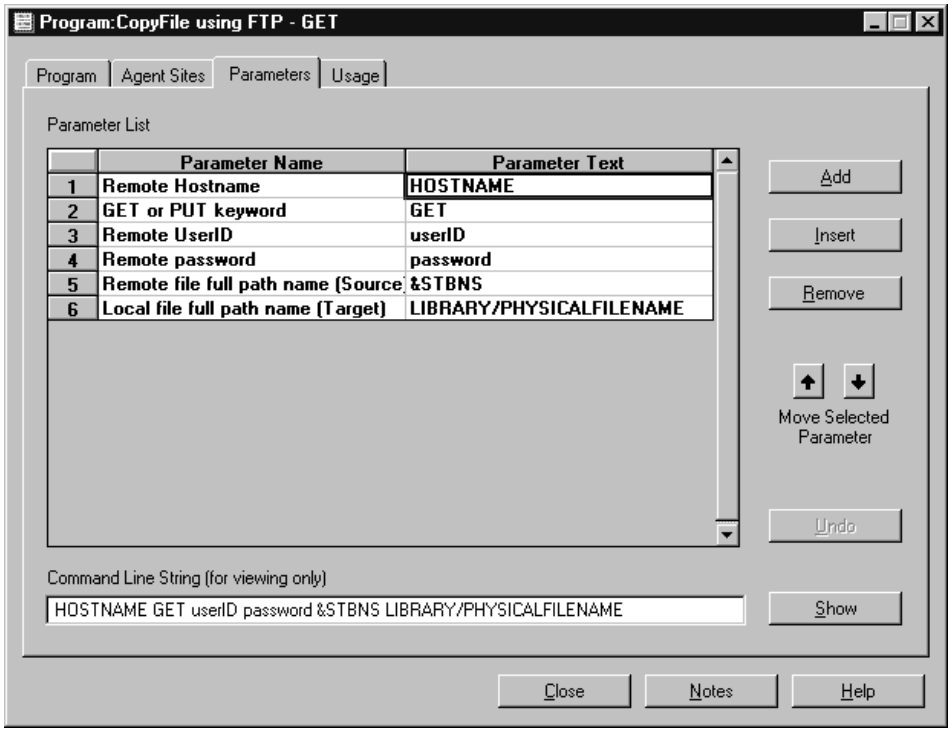


Figure 28. An example of a completed Parameters page for VWPRCPY (GET)

For information about adding parameters, see “Setting parameters” on page 36.

Defining VWPRCPY (PUT)

You use this program to transfer a file to a remote system.

Providing information about the program

When you define the Program page for VWPRCPY (PUT), ensure that you specify the following information:

- The name of the Program Group is **VW Sample Program Group**.

  To add a new program group, select a field in the **Program Group** list. Then type the name of the new group in the field. To change the name of an existing group, type the new name of the group over the old one.
- The Program Executable Type is **Executable**.
- The Fully Qualified Program Name is **QIWH/VWPRCPY**.

Figure 29 shows an example of the completed Program page:

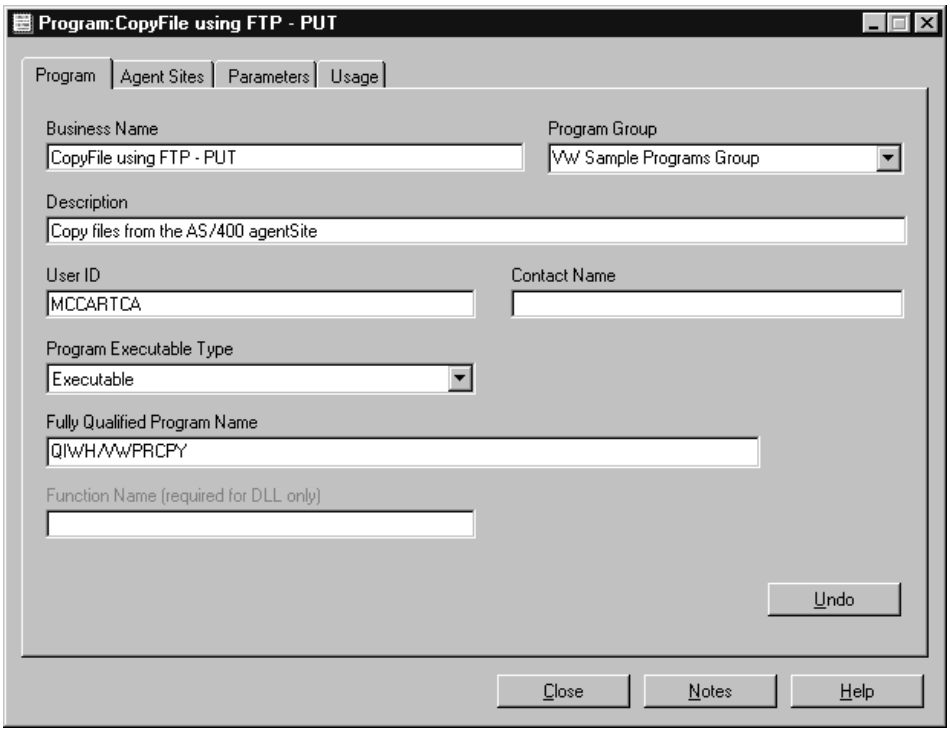


Figure 29. An example of a completed Program page for VWPRCPY (PUT)

Setting default parameters

You use the Parameters page in the Program notebook to set the default parameters for a program.

To set the default parameters for VWPRCPY (PUT), add the following parameters to the **Parameter List**:

- Remote Hostname
- GET or PUT keyword (define the value as PUT)
- Remote UserID
- Remote password
- Remote file full path name (Target)
- Local file full path name (Source)

Figure 30 on page 101 shows an example of a completed Parameters page:

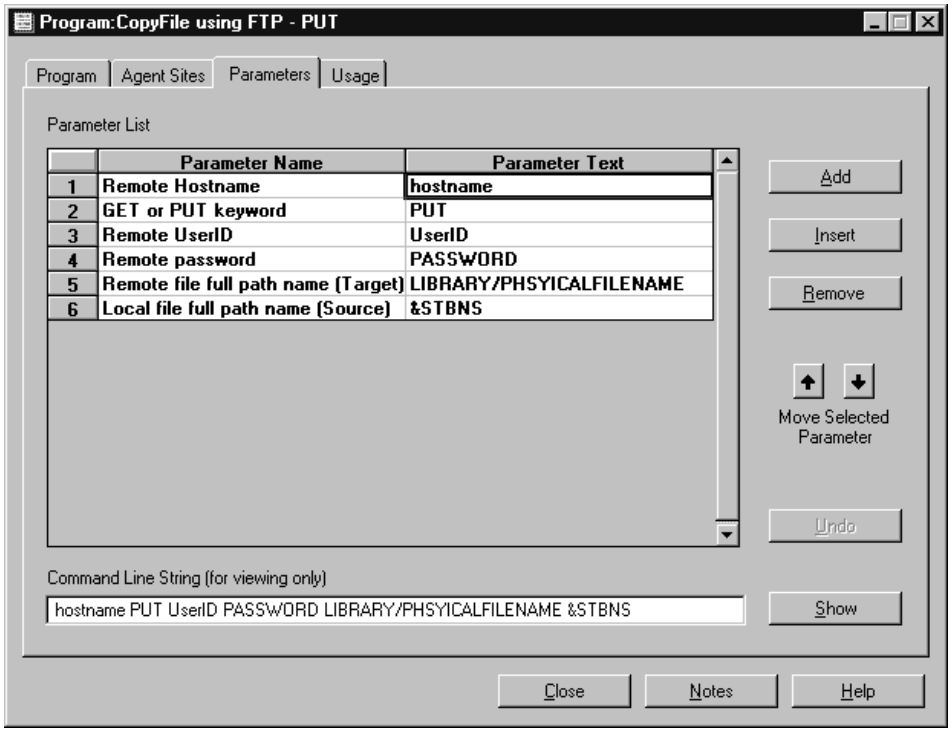


Figure 30. An example of a completed Parameters page for VWPRCPY (PUT)

For information about adding parameters, see “Setting parameters” on page 36.

Appendix C. Changing the status of a business view

You can use Visual Warehouse to manage the development of your business view by classifying business views in one of three statuses: development, test, or production. Table 2 on page 103 describes each status level and the changes that occur to your business view when you promote or demote your business view to one of these levels.

Table 2. Business view status levels

Business view status	**When you will see it**	**What you can do with it**	**What you cannot do with it**	**What happens when you promote or demote to this status**
Development	• When you create a business view • When you demote a business view from test	Change any part of the business view definition	Run the business view either manually or according to a schedule	When you demote a business view to development, Visual Warehouse drops the warehouse table for the business view, and the data in the table is lost. A target table cannot exist for a business view in development status.

Table 2. Business view status levels (continued)

Business view status	When you will see it	What you can do with it	What you cannot do with it	What happens when you promote or demote to this status
Test	• When you promote a business view from development • When you demote a business view from production	• Add items to the business view. For example, you can add a column definition to your business view. • Test the business view by running it manually. • Populate the business view table by running the business view manually. • Change the values of the information fields in the Business View notebook, such as the **Description** field.	• Remove items from a business view. For example, you cannot remove a column definition from your business view. • Use a schedule to populate the business view table.	• When you promote a business view to test status, a warehouse table is created for the business view (if you request that Visual Warehouse create tables). • When you demote a business view to test status, any schedule that is attached to the business view will no longer apply to it. • Whether you promote or demote to this status, Visual Warehouse creates an entry for the business view in the Run Business View window.

Table 2. Business view status levels (continued)

Business view status	When you will see it	What you can do with it	What you cannot do with it	What happens when you promote or demote to this status
Production	When you promote a business view from test	• Run the business view according to a schedule that you attach to it. • Populate the business view table that was created when you promoted the business view to Test. • Change the values of the information fields, such as the **Description** field.	Make certain changes to the business view definition. For example, you cannot modify the SQL.	When you promote a business view to production, the schedule (if any) that is attached to the business view is activated.

After you define a business view, you can change its status.

To change the business view from development to test status:

1. Select the business view name in the Business View List window.
2. From the menu bar, select **Status → Promote to Test**.

   A DB2 table is created for the business view, and an entry is created for the business view in the Run Business View window.

To promote the business view from test to production status:

1. Select the business view name in the Business View List window.
2. From the menu bar, select **Status → Promote to Production**.

   The business view table will be populated when the business view runs.

If you want to change a production-level business view, you can return your business view to test status, make changes, and promote the business view to

production again. If you have Business View Maintenance privilege, you can return the business view to test status even if the business view has dependent business views in production status. If you do not have Business View Maintenance privilege, you must also return any dependent business views to test status.

To demote your business view to test status:

1. Select the business view name in the Business View List window.
2. From the menu bar, select **Status → Demote to Test**.

When the status of a business view is test, you cannot make certain changes to the definition. To make these changes, demote the business view to development status. When you demote a business view to development status, Visual Warehouse drops the DB2 table for the business view, and the data in the table is lost.

To demote your business view to development status:

1. Select the business view name in the Business View List window.
2. From the menu bar, select **Status → Demote to Development**.

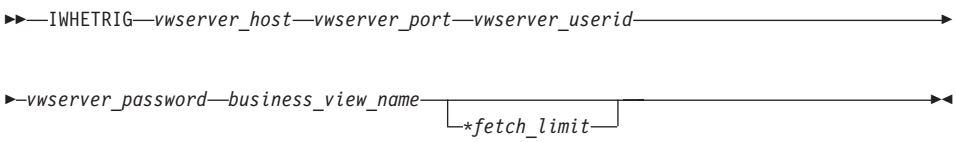
For more information about business view status, including a comprehensive list of tasks that you can complete at each level, see the online help.

Appendix D. Using the external trigger program

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

The external trigger program is included with the AS/400 agent code. The program is called IWHETRIG.

IWHETRIG syntax



vwserver_host	Either the host name of the Visual Warehouse server or its TCP/IP address.
vwserver_port	The TCP/IP port assigned to the Visual Warehouse server.
vwserver_userid	A user ID with Visual Warehouse Operations function privileges.
vwserver_password	The password for the user ID.
business_view_name	The name of the business view to start. This name is case sensitive.
**fetch_limit*	The maximum number of rows to fetch. This parameter is optional and is valid only when the business view is in test status.

To use the external trigger program, you call it from an AS/400 command line as you would any other application program. There are two ways to do call the external trigger program:

- Enter CALL QIWH/IWHETRIG. When the prompt screen for the CALL command is displayed, enter the parameters in the spaces provided.
- Use the full CALL syntax from the AS/400 command line. For example:

  ```
  CALL PGM(QIWH/IWHETRIG) PARM('VWSERVER' '11000' 'UID' 'PWD' 'YOURBV')
  ```

  where:

- VWSERVER is the vwserver_host
- 11000 is the vwserver_port
- UID is the vwserver_userid
- PWD is the vwser_password
- YOURBV is the business_view_name

Because *fetch_limit is optional, it is not listed in this example.

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

IWHETRIG 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.

The business view might take hours or even days to run, depending on the size of the data and the complexity of the query. Therefore, IWHETRIG 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 window or the Log Viewer windows.

Appendix E. Determining the status of a conditional cascade

After a conditional cascade program runs, Visual Warehouse verifies its return code. If the return code equals 0, Visual Warehouse starts the next business view. If the return code does not equal 0, Visual Warehouse will not start the next business view.

When a business view that has a conditional cascade program attached to it runs successfully, the successful icon is displayed in the Operations Work In Progress window. This icon refers only to the business view. It does not apply to the status of the conditional cascade program. For example, if a business view runs successfully, and its conditional cascade program fails, the successful icon is still displayed.

To determine the status of a conditional cascade program, verify that the business view it starts is running. If the conditional cascade program ran successfully, the edition for the second business view is displayed in the Operations Work in Progress window. If the conditional cascade program failed, no edition is displayed for the second business view.

To access the Operations Work in Progress window, select **Operations → Work in Progress** from the Visual Warehouse desktop.

To view the value returned by a failed cascade program, open the Log Viewer Details window for the business view that has the conditional cascade program attached to it. The return code for the conditional cascade program is displayed in the **Error RC2** field.

To access the Log View Details window:

1. Click **Log** in the Operations Work in Progress window.
   The Log Viewer window opens.
2. Select the name of the business view. Then, click **Details**.
   The Log Viewer Details window opens.

Conditional cascade status

Appendix F. Troubleshooting

This appendix describes possible solutions to problems you might encounter as you work with Visual Warehouse and the AS/400 agent. For general troubleshooting questions, see *Managing Visual Warehouse*.

Visual Warehouse and the AS/400 agent are unable to communicate

To communicate with each other, Visual Warehouse and the AS/400 agent must use the same TCP/IP port entries. If you change a port number in the Windows NT Services file, but do not change it on OS/400 (or vice-versa), communication between Visual Warehouse and the AS/400 agent will fail.

Visual Warehouse refers to ports by name rather than by number. During installation, Visual Warehouse assigns names to port numbers. On Windows NT, Visual Warehouse writes the port assignments to the Services file. On OS/400, Visual Warehouse uses the WRKSRVTBLE command to store the port assignments. By default, Visual Warehouse uses these ports:

- 11000 (vwkernal)
- 11001 (vwd)
- 11002 (vwlogger)

Only change a port number if that port number is already in use by another application. If you change a port number on either Windows NT or OS/400, ensure that you change the corresponding number on the other operating system.

The Visual Warehouse Logger stops while Visual Warehouse is running

If the Visual Warehouse Logger stops while Visual Warehouse is running, you will hear five beeps. The following message might be displayed:

```
There are no log records to display
```

If this message is displayed, the DB2 Log for the control database is probably full. To fix this problem, you must increase the number of primary or secondary files for the Visual Warehouse control database.

If this message is not displayed, contact IBM Technical Support.

FTP log displays ″NAMEFMT not a valid command″

This message is displayed if the FTP server system does not support the NAMEFMT command. The NAMEFMT command is an AS/400 specific FTP subcommand.

It is normal for this message to be displayed. You should ignore it.

Data types are misinterpreted by Visual Warehouse or OS/400

There are some datatypes that require special handling when you are working with the AS/400 agent. The sections that follow describe problems that will occur, and suggest solutions that you can implement to resolve them.

Adding FOR BIT DATA to columns that contain binary data

If you have columns that contain binary data, you must edit the DDL to add the FOR BIT DATA keyword. If you do not add this keyword, the binary data will be considered normal character data. The data will go through page translation on INSERT into the target column.

To edit the DDL:

1. Open the Business View notebook for your business view.
2. Click the **Information** tab.
3. Click **Create DDL**.
   The Create DDL window opens.
4. Edit the DDL.
5. When you are finished, click **OK**.

Columns defined as BINARY WITH PRECISION are interpreted as INTEGER

If your source table is an AS/400 file that contains fields that are defined as BINARY WITH PRECISION, the datatypes of these fields will be returned as INTEGER by DB2 for AS/400. For example, when you view a column that is defined as BINARY WITH PRECISION in the Add Columns window, it is displayed with an INTEGER data type.

This is a permanent AS/400 restriction. This restriction causes the fractional portion of the BINARY WITH PRECISION data to be truncated.

Visual Warehouse displays return code 7356

This section describes some of the problems that can cause Visual Warehouse to display return code 7356.

Importing source tables from a remote, earlier release of an AS/400 system

When you attempt to import table definitions from a remote AS/400 system prior to Version 4 Release 2 to a new information resource, you might receive the following messages:

- From Visual Warehouse:

```
**Table list error**

Return Code = 7356
Message = An agent's processing of a command of type
selectAndInsertEdition failed for edition x of
business view y.
```

  where x is the business view edition number and y is the business view name.

  A secondary return code is also displayed:

```
Method = VWabsCommand::ExecuteOnce
Secondary Code = 8515
System Message = Section number 500 not valid. Current
high section number is 200.
```

- From the OS/400 operating system:

```
 Message . . . . :   Section number 500 not valid. Current high
     section number is 200. Reason 4.
```

This error occurs when the SQL package for the SQL Call Level Interface is older than the current version. The SQL package is located in the QGPL library on the system that is remote from the system where the import was requested.

To correct this problem:

1. Delete the SQL package from the QGPL library on the system that is remote from the system where the import was requested. To do this, enter the following command at an AS/400 command line:

   ```
   DLTSQLPKG SQLPKG(QGPL/QSQCLI*)
   ```

2. Import your tables again.

   A new, current SQL package for the Call Level Interface is created when you import.

Preserving CCSIDs in a CREATE TABLE statement

The character coded set identifier (CCSID) is an identifier for a collection of one or more code pages and character sets, and the encoding scheme for the character set.

When you promote a business view to test status, the character and graphic columns are created with the CCSID of the job that creates the business view table during the promote operation. When you run the business view, the actual CCSIDs are retrieved from the source. If your business view extracts character or graphic data, an error will occur if the default CCSIDs that were created during the promotion of the business view do not match the actual CCSIDs. If this error occurs, you will receive the following messages:

- From Visual Warehouse, you receive return code 7356.
- From the OS/400 operating system:

```
Message ID . . . . . :   SQL0330    Severity . . . . . :   30
Message type . . . . . :   Diagnostic
Date sent  . . . . . :   aa/aa/aa   Time sent  . . . . . :   bb:bb:bb

Message . . . . :   Character conversion cannot be performed.
```

where:

aa/aa/aa is the date the message is sent

bb:bb:bb is the time the message is sent

To preserve the AS/400 CCSID for the individual columns in character and graphic type data, you must either create the table outside of the promote or add the CCSID xxx clause to the CREATE TABLE statement for each column that requires a specific CCSID.

To add a CCSID xxx clause to a CREATE TABLE statement:

1. Open the Business View notebook for the business view that you want to work with.
2. Select the **Information** tab.
3. Click **Create DDL**.
4. Edit the CREATE TABLE statement to include the CCSID clause for each column that requires it.

   For example, consider the following CREATE TABLE statement:

```
CREATE TABLE IWH.BUSINESS_VIEW_TEST_3

  (KEYFLD NUMERIC(3,0)  NOT NULL,
   CHAR1 CHAR(10),
```

```
CHAR2 CHAR(15)  NOT NULL,
CHAR3 CHAR(1),
CHAR4 CHAR(3)  NOT NULL,
OPEN4 CHAR(22),
ONLY4 CHAR(8)  NOT NULL,
DATE1 DATE,
DATE2 DATE NOT NULL)
```

In this example, the OPEN4 and ONLY4 fields need to retain their CCSID values when the business view is promoted to test status.

To edit this CREATE TABLE statement to include the proper CCSIDs:

a. At an AS/400 command line, enter DSPFFD against the information source table. The CCSIDs associated with any CHAR, VARCHAR, GRAPHIC, and VARGRAPHIC data types are displayed.
b. Add the correct CCSID information to the CREATE TABLE statement to the fields that must retain their CCSIDs.

   For the example CREATE TABLE statement, the correct CCSID information is CCSID 937:

```
CREATE TABLE IWH.BUSINESS_VIEW_TEST_3

  (KEYFLD NUMERIC(3,0)  NOT NULL,
   CHAR1 CHAR(10),
   CHAR2 CHAR(15)  NOT NULL,
   CHAR3 CHAR(1),
   CHAR4 CHAR(3)  NOT NULL,
   OPEN4 CHAR(22) CCSID 937,
   ONLY4 CHAR(8)  NOT NULL CCSID 937,
   DATE1 DATE,
   DATE2 DATE NOT NULL)
```

5. Click **OK** to save your changes and close the Create DDL window.
6. Click **OK** to save your changes and close the Business View notebook.

Target table is not being journaled

Return code 7356 is also displayed when the target table is not being journaled. To verify that this is the problem, check the AS/400 agent joblog.

You should see the following message:

```
CPF4328 Member nnnnnnnnnn not journaled to journal *N
```

where nnnnnnnnnn is the name of the file/table.

To resolve this problem, ensure that the target table is being journaled.

Using DB2 for MVS as a source or a target

The concept of a database is different for DB2 for MVS than it is for the rest of the DB2 family.

For example, a DB2 for UDB database is a collection of information, such as tables or views, that can be connected to and accessed from an application. In other words, for UDB, the database you connect to will be the one that contains the information you want to work with.

In DB2 for MVS, the entity that you connect to and the database that contains your data are different. The entity you connect to is an entire DB2 subsystem. Each subsystem may contain numerous databases. From the DB2 for MVS perspective, a database is just a logical collection of information, such as tables, views, and other relational objects.

This difference can impact users using DB2 for MVS as a source or a target database. If a database name is not explicitly specified, DB2 for MVS assumes you wants to create or access data that resides in the default DB2 database defined for the given subsystem. This is most likely to impact users who want to use DB2 for MVS as a Visual Warehouse target.

Creating non-system default target tables on DB2 for MVS

If you want Visual Warehouse to create target tables in an MVS database other than the system default, you must specify the database name and the table space name in the business view definition:

1. From the Business View list, select the business view that you want to work with.
2. Double-click the business view.

   The Business View notebook opens.
3. Select the Information tab.
4. In the **DB2 Table Space** field, type:

   `databasename.tablespacename`

   where:

   `databasename` is the name of the database and `tablespacename` is the name of the table space.
5. To save your changes and close the Business View notebook, click **OK**.

   The database name and the table space name are now added to the CREATE TABLE statement.

To specify the database name within the default table space, in the **DB2 Table Space** field, type:

```
DATABASE name
```

where name is the name of the database.

Promoting a business view that uses DB2 for MVS Version 4 Release 1 as the source

During the promotion of a business view with DB2 for MVS Version 4 Release 1 as the source, Visual Warehouse might seem to hang. This problem is caused by the IWH2AGNT job going into an ICFW wait state indefinitely.

To verify that the IWH2AGNT job is the cause of the problem:

1. At an AS/400 command line, enter WRKACTJOB.
2. Verify that IWH2AGNT is in an ICFW state.

To resolve this problem:

1. Apply PTF UQ02568 (APAR PQ01375) to DB2 for MVS Version 4 Release 1.
2. Apply PTFs SF47985 and SF47986 to your AS/400 Version 4 Release 2 system.

No code changes to Visual Warehouse required.

Promoting a business view that uses DB2 for MVS as the target warehouse

When you promote a business view to test using DB2 for MVS as your target warehouse, the promotion might fail with the following messages:

- An 8525 return code
- The System Message, Resource Limit exceeded

This problem occurs when Visual Warehouse tries to create the target table in the DB2 for MVS default database and tablespace, and the default tablespace is already full or restricted to your user ID.

You can resolve this problem in two ways:

- If you want your target tables to go to your default database, you can increase the amount of space in your default database.
- If you don't want your target tables to go to your default database, you can create another database and tablespaces for use by Visual Warehouse.

  After you create the database and tablespaces, complete the following tasks:

  1. Define a warehouse for the database.

2. When you create a business view for your new warehouse, manually edit the Visual Warehouse-generated CREATE statement, as described in "Using DB2 for MVS as a source or a target" on page 116.
3. To save your changes and close the Business View notebook, click **OK**.

Problems with sample Visual Warehouse programs

This section describes problems that can occur when you use the sample Visual Warehouse programs, and suggests solutions you can implement to resolve them.

Business view running VWPRCPY fails with RC2 = 12

Verify that the VWxxxxxxxx.VWPRCPY trace file contains a message that says the FTP command file couldn't be created, OS/400 errno = 3021.

If this is the case, the user profile under which the agent process was running doesn't have the authority to create members in the QIWH/FTPCMD source physical file.

To fix this problem, either use a different user profile to run the agent or grant the agent user profile authority to access the QIWH/FTPCMD file.

VWPFTP trace says that job completed successfully, but data wasn't transferred and the FTP message log is missing.

The library or physical file name you gave as containing the FTP command file member does not exist or is invalid. Because of this, FTP executes successfully, but the stream of commands it gets is empty. Therefore, FTP does not do anything. As a result, the FTP message log doesn't get generated.

To resolve this problem, ensure that the library, physical file, and command member actually exist. If they do, check for typographical errors in the command file parameter value passed to VWPFTP. The VWPFTP trace lists the parameter values that were passed to it.

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:

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	DRDA
AS/400	IBM
DataGuide	IMS
DataJoiner	MVS
DataPropagator	OS/2
DB2	OS/390
DB2 Olap Server	OS/400
DB2 Universal Database	

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Other company, product, and service names may be trademarks or service marks of others.

Bibliography

For other questions about how Visual Warehouse works, see the online help and other hardcopy documentation.

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

Managing Visual Warehouse, GC26-8822

Planning and Installing Visual Warehouse and DataGuide, SC26-3496

IBM Data Guide: Managing DataGuide, SC26-3362

IBM DB2 OLAP Server: Using DB2 OLAP Server, SC26-9235

IBM Visual Warehouse Host Adapters: MVS Planning and Installation Guide, GC26-8821

IBM Visual Warehouse Host Adapters: Non-Relational Data Mapping Utility User's Guide, GC26-8932

IBM Visual Warehouse Host Adapters: ODBC Driver Installation and User's Guide,, GC26-8931

Publications related to connectivity

Distributed Relational Database Cross Platform Connectivity and Application, SG24-4311

Distributed Relational Database Architecture Connectivity Guide, SC26-4783

IBM DDCS for Windows NT Installation and Configuration Guide, S33H-0311

IBM DDCS User's Guide for Common Servers, S20H-4793

AS/400 Publications

OS/400 DDS Reference Version 4 Release 2 (SC41-5712-01).

Bibliography

Glossary

This glossary defines terms that are used in this book and throughout the Visual Warehouse Version 3.1 library.

adapter. A Visual Warehouse component that is located on remote operating systems and provides direct access to various kinds of operational databases and operating systems. Adapters are designed to extract data on request and return it to Visual Warehouse.

administration client. A workstation 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.

agent. A Visual Warehouse component that handles the flow of data between data sources and target tables in a data warehouse.

agent process. The transfer of data by an agent between a source and a target database.

agent site. The machine on which an agent runs.

business metadata. Metadata used primarily by end users to locate information in the datamart or data warehouse.

business view. A step in the transformation process from the data's source format to its target format.

business view edition. A snapshot of the data in a business view at a particular time.

conditional cascade program. A Visual Warehouse program that determines whether a condition that you specified was met after a business view runs. If the condition was met, the program returns a 0 return code. The 0 return code initiates a cascade that starts the next business view.

datamart. A data warehouse that contains informational data optimized for end-user decision making and data analysis.

data model. A description of the organization of data in a manner that reflects the information structure of an enterprise.

data warehouse. Stores of *informational data.*

Distributed Relational Database Architecture (DRDA). An IBM-published architecture for providing access to relational data.

extract. To select and remove from a group of items those items that meet specific criteria.

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. A database or a flat file that contains source data.

local agent. An agent that is installed on the same machine as the Visual Warehouse server.

metadata. Data that describes the characteristics of stored data; descriptive data.

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.

post processing program. A Visual Warehouse program that performs an action that you want to occur after a business view runs. This is the default Visual Warehouse program type.

program group. A logical grouping of programs.

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.

remote agent. An agent that is installed on a different machine than the Visual Warehouse server.

star schema. A type of relational database schema made up 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.

status report. A report that you build on a regular basis that helps you manage a business process over time.

Structured Query Language (SQL). A command language used with relational databases. The language consists of statements to insert, update, delete, query, and protect data.

subject. A collection of business views that, by default, do not result in tables in a target warehouse. For example, a subject can consist of a group of business views that contain intermediate programs that cleanse, load, and unload data.

target table. The database table that contains the results of the Visual Warehouse business views after they are run.

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.

Visual Warehouse program. An optional aggregation program, provided with Visual Warehouse or written by a user, that augments the aggregation of a business view.

warehouse. A DB2 database that contains business views that are of interest to a particular group of end users.

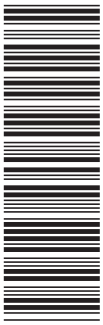
Index

Index

Part Number: CT64AIE
Program Number: 5697-VW3

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

CT64AIE



SC26-9468-00