Managing a Multi-tier Data Warehousing Environment with the SAS/Warehouse Administrator™

Darrell Barton, SAS Institute Inc., Cary, NC

ABSTRACT

This paper describes a multi-tier computing environment and demonstrates the use of the SAS/Warehouse Administrator software to manage the process of data extraction and data movement using different middleware products. The legacy data that is the source used to populate a data warehouse may reside in many formats and on many platforms. There are several middleware products available that can be used to extract and move that data from the source systems to the data warehouse. Some of these products include ODBC drivers, database middleware such as Oracle® Corporation's client software, and SAS/CONNECT® software. SAS® software and the SAS/Warehouse Administrator software can be used to build and manage the data warehousing process which exploits the existing hardware platforms, data storage formats, network topology, and middleware products while controlling the load on the network.

INTRODUCTION

The architecture for a data warehouse rarely involves a single computing environment. You may have a mainframe handling legacy applications and their associated data, a newer Unix environment that handles other day-to-day operations, and a PC based system for building and managing the data warehouse processing. A multi-tier computing environment is one such as this where there are different systems that must be utilized for their data or computing resources.

One of the first problems encountered in the warehousing process is gaining access to all the enterprise's data wherever it may be located. The heterogeneous nature of the systems found in most organizations' IT department is one of the main obstacles in building a successful data warehouse.

The SAS/Warehouse Administrator software helps you manage the extraction of the data from legacy and OLTP systems, apply business rules to that data to transform it into a useful form for the warehouse, and then load the transformed data into the data warehouse. This process is commonly called ETL for extraction, transformation, and load. ETL can be easily handled with the components of the SAS System that have been available for many years, SAS/ACCESS® or base SAS software for the extraction, base SAS for the transformations, and SAS/ACCESS or base for the loading of the data warehouse. The SAS/Warehouse Administrator provides a metadata driven facility to manage the process and help create self-documenting, repeatable processes for warehouse creation and management across the entire enterprise.

What is a multi-tier environment?

Most organizations have "pockets" of information scattered throughout their operations. For example, the sales department has sales and billing information, the marketing department has advertising and promotional information, and the customer service department has details of contact with the customer after the sale has been made. All of this information is segregated and is typically available only to the group that uses a particular system.

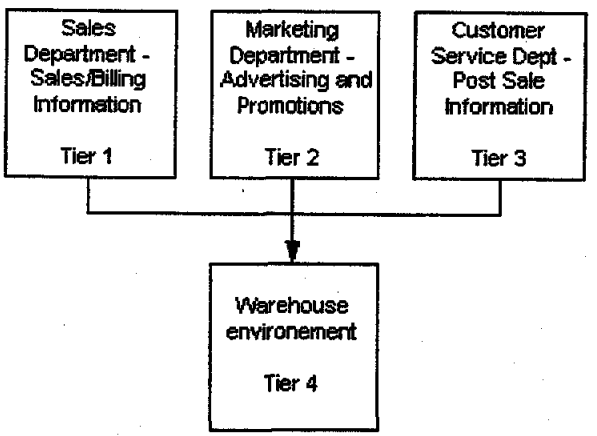

Figure 1: Common Multi-tier environment

Each group has designed their system to meet their unique needs and has purchased the pieces of their solution that best fit those needs. For the organization to get a clear picture of their operations, they need the data from each of these systems included in the data warehouse. Each department's system in Figure 1 is a tier in the data warehouse process so to build a successful data warehouse usually involves accessing multi-tiers.

SAS/Warehouse Administrator Integration with SAS/CONNECT Software

As a part of the SAS System, the SAS/Warehouse Administrator utilizes all the components available with the SAS software. SAS/CONNECT software is the middleware module that enables the communication and data transfer between SAS sessions on different systems.

Using SAS/CONNECT you are able to access data on remote systems for manipulation on the local host. It also provides remote compute services so that large volumes of data can be processed and reduced on the host where it resides. Otherwise, large volumes of data would be passed across the network for transformation processing on a separate system.

With the SAS/Warehouse Administrator software, you are provided with a template, Figure 1, to define your various hosts and the connection protocol used to communicate with them.

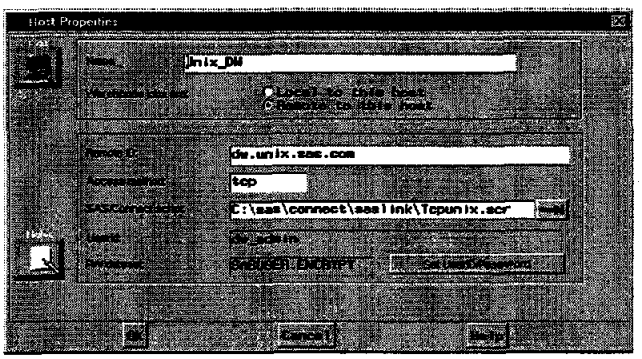

Figure 2: Host Properties Window

Entering the host information into this window also adds the information to the SAS/Warehouse Administrator's metadata repository. When the host is used later in the process, the host name is selected and then the information is reused to make the connection to that host. Figure 3 shows using the host defined in Figure 2 as the data server for an Operational Data Definition (ODD). There are many other areas within the SAS/Warehouse Administrator interface where the host information is reused.

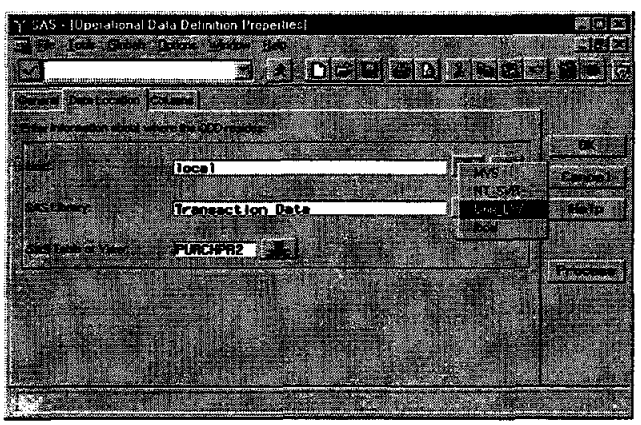

Figure 3: Operational data definition using defined host

Using database middleware

Another way to incorporate a platform in the data warehousing process is to utilize the database middleware provided by the database vendor. The SAS/ACCESS software links with the database vendors' client software to provide a native access solution to the database system. Using this third party middleware mechanism, you can access a remote database from the local host. Since the SAS/Warehouse Administrator integrates with the rest of the SAS software, it is able to incorporate the database system in to the warehouse for both extraction and for loading.

For example, in Figure 4 SAS/ACCESS Interface to Oracle goes through post installation processing to link with the Oracle client software. The result of this linking process is an executable image containing both SAS and Oracle code that enables the SAS System to access the Oracle database system. This image makes calls to the Oracle client software that are necessary to communicate with the database server. In this example the database server resides on a remote system but the same linking process enables the access to a local database server as well.

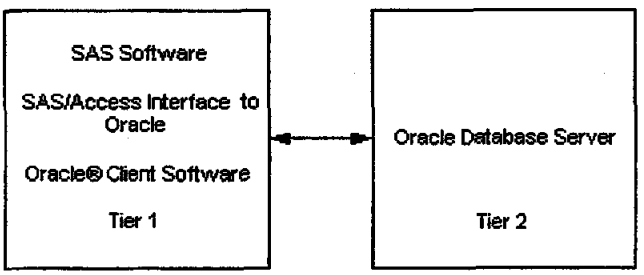

Figure 4: Example configuration with database middleware

Once the post installation process is complete the database tables can be treated as SAS data sets and incorporated into the warehouse. If the database server resides on another platform from the SAS software, the database middleware handles the transportation of the database tables to the SAS software for processing.

Table 1 below provides a list of the major database vendors' middleware products that are utilized by SAS/ACCESS software to communicate with the associated database.

Table 1: Database middleware products

Database name	Middleware Software
Oracle	Oracle8™ Client/SQL*NET®
DB2®/UDB®	DDCS®/DB2 Connect®
Sybase	Sybase Open Client
Informix	Informix-Net/Informix-Star
Ingres	Ingres/Net

Using ODBC Drivers

Similar to using the middleware provided by database vendor, ODBC drivers often contain data transportation

elements that will move data from a remote server platform to the local host requesting the data. ODBC stands for Open Database Connectivity. It is an interface standard that provides a common application programming interface (API) for accessing databases. Many software products that run in the Windows operating environment adhere to this standard, giving users access to data that was created with other software.

SAS/ACCESS Interface to ODBC software utilizes the ODBC API to communicate and access databases supporting the ODBC standard. The components necessary for SAS software to communicate with a database server using ODBC are shown in Figure 5.

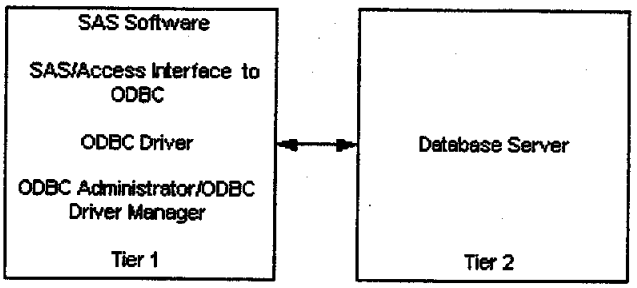

Figure 5: Example configuration using ODBC

There are three components on Tier one in this configuration that are not supplied by SAS Institute Inc. They are:

- The ODBC Driver
- ODBC Driver Manager
- ODBC Administrator

The ODBC architecture and the relationship of the ODBC components are shown in Figure 6. In this diagram, the SAS System using the SAS/ACCESS Interface to ODBC is the client application

The ODBC Driver Manager is the component that the client applications link with. It loads the appropriate driver requested by the client application for a database connection and does some system level management, but primarily passes all calls through to the ODBC driver.

The ODBC driver is the piece that actually provides the interface to a specific data source. Drivers are typically provided by the database vendor or by third party vendors such as MERANT.

A multi-tiered ODBC driver is one that communicates with a database server on another machine. A multi-tiered driver does not directly manipulate database data. It generally makes calls to a DBMS-provided API that allows the driver to communicate directly with the DBMS server as a client application.

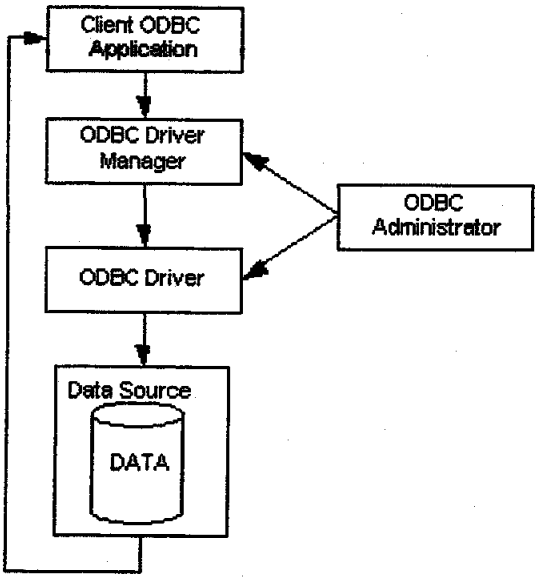

Figure 6: Relationship of ODBC Components

The ODBC Administrator is used to install drivers and create/configure data sources. The ODBC Administrator is not involved in the run time call sequence of a client application, but is used for configuring ODBC Data Source Names prior to their use in client applications.

To summarize the flow of Figure 6, SAS/ACCESS Interface to ODBC makes a call to an ODBC data source via the ODBC Driver Manager. The call is passed to the appropriate ODBC Driver which would convert that ODBC call into a DBMS query. The query results are then passed back to SAS for processing.

SAS/Warehouse Administrator Integration with Third Party Middleware

The SAS/Warehouse Administrator utilizes the SAS/ACCESS software to communicate with the databases. The middleware used depends upon the database being accessed and the SAS/ACCESS product used.

The first step in integrating the database into the SAS/Warehouse Administrator is to define the SAS/ACCESS views to the database tables that are to be loaded into the data warehouse. The views can be SAS/ACCESS view descriptors or SQL pass through views. Once the views are defined, they can be used as a data source to define an ODD.

The views can be defined locally using SAS/ACCESS software on the same system where the SAS/Warehouse Administrator is installed or they can be defined remotely and accessed using Remote Library Services (RLS) of SAS/CONNECT. Defining a local database table is shown in the Operational Data Definition window in

Figure 7 below. The GOODCUST database view is selected from the Transaction Data library.

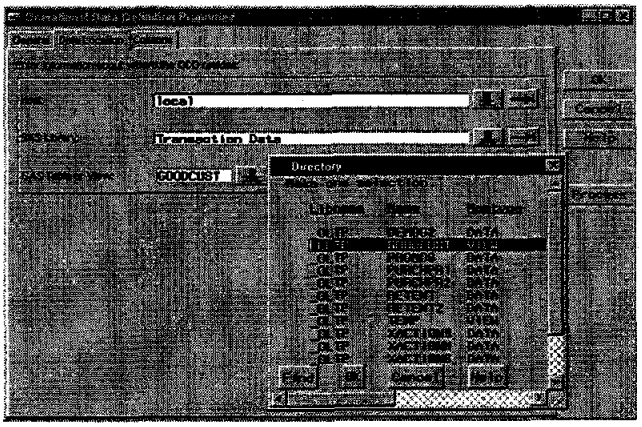

Figure 7: Using a local SAS/ACCESS view as an ODD

To access another system where the SAS/ACCESS view is located you simply change the "Host" and "SAS Library" fields in the ODD properties window in Figure 7. When the "SAS Table or View" is selected, a SAS/CONNECT session is established which allows you to use RLS to select the remote data set or view.

It should be noted that selecting a SAS/ACCESS view to a database table on another platform does not mean the database server is on that same platform. Using the database middleware and the SAS/ACCESS software you could quickly add another tier to your warehousing environment which hosts the database server.

Multi-tier Warehousing Environment Example

Figure 8 shows an example of a five tier data warehouse architecture. The five tiers are:

1. Windows platform with the SAS/Warehouse Administrator
2. MVS platform with legacy data in DB2 and VSAM to be loaded into the data warehouse
3. Unix platform with transactional data in Oracle to be loaded in the data warehouse
4. Window NT platform with database information in SQL Server to be loaded in the data warehouse
5. Unix Platform with Oracle database for warehouse storage

This section will describe how the data will be extracted, transformed, and loaded from the various platforms to the data warehouse using the SAS/Warehouse Administrator. The lines in Figure 8 are numbered. This section will progress through the warehousing process going from number one to number six describing the processing that takes place in that step. The steps in the diagram are provided for reference only. It is not implied that the steps must be defined or processed in this order. The steps should be considered totally separate entities.

Previously in this paper, the methods for defining hosts (Figure 2) and using RLS to access remote views (Figure 7) have been discussed. It is assumed that those steps have already been completed in the SAS/Warehouse Administrator and will not be covered in this example.

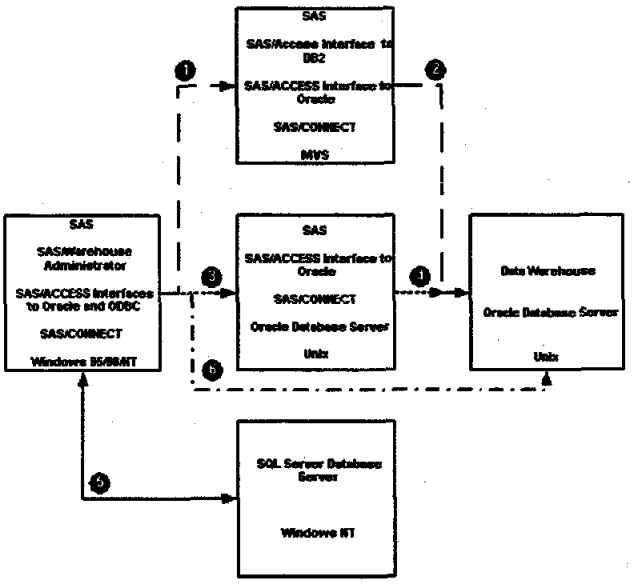

Figure 8: Multi-tier Example

Step 1: Extracting the DB2 and VSAM Data

In step 1 the SAS/Warehouse Administrator creates a process that will run on the mainframe to extract the data from the DB2 database and VSAM files and transform that data by applying the business rules for the warehouse. With the Multi-vendor Architecture of the SAS software, the job generated on the Windows systems by the SAS/Warehouse Administrator can easily be scheduled and run using the SAS software on the MVS mainframe system.

To access the data on the mainframe and utilize it within the SAS/Warehouse Administrator, the views to the data must be created on the mainframe. The SAS/ACCESS Interface to DB2 would be used to create the views for the DB2 data. For the VSAM files, SAS data step views would be created. The views can then be used as ODD's in the warehouse process. Defining ODD's is done in the ODD Properties window as shown in Figure 7.

After the ODD's have been defined, the process of extracting the data, transforming the data, and loading the data is defined in the SAS/Warehouse Administrator's Process Editor. The Process Editor begins as an empty palette that is used to define the steps needed to transform the legacy or operational data from its original form into

the cleaned warehouse data. An example of the process editor is shown in Figure 9.

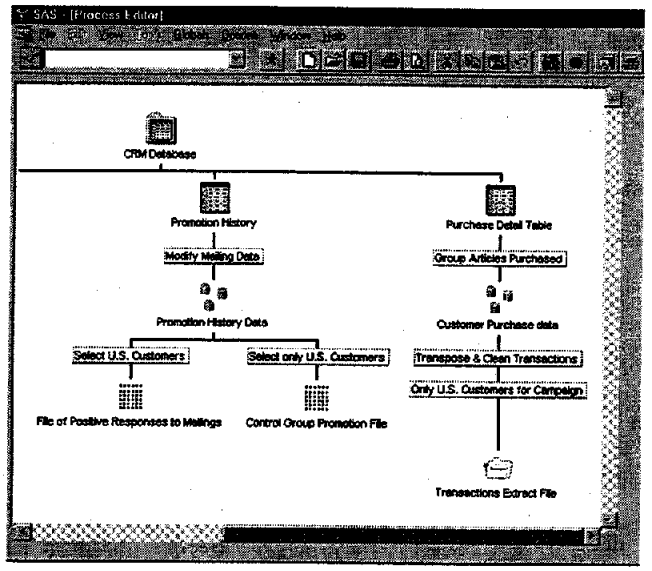

Figure 9: SAS/Warehouse Administrator Process Editor

Step 2: Loading the Oracle Warehouse

Once the data has been cleansed, it is ready to be loaded into the data warehouse. With the SAS/ACCESS Interface to Oracle Software installed on the mainframe system along with the Oracle Client software, the data can be moved and loaded at the same time into the Oracle data warehouse on the Unix system. The database connection is defined in the SAS/Warehouse Administrator metadata DBMS Connection Properties screen in Figure 10.

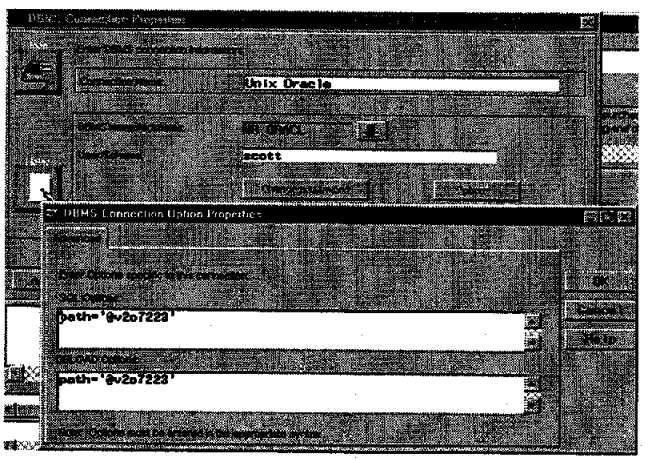

Figure 10: DBMS Connection Properties

The key fields in the screen are the *User/Schema, Password* and the *DBLOAD Options* fields. The *User/Schema* field is used to identify the user ID for logging into the Oracle database server. Another field, the *Password* field (not shown) allows for the entry of the Oracle password. The *DBLOAD Options* field is used to identify options that must be used for loading the data into the database. In Figure 10 it is used to define the Oracle path for connecting to the Oracle database server on Unix. The information entered is added to the SAS/Warehouse Administrator metadata and used to build the ETL process.

The load step is defined by selecting a table in the Process Editor, pressing the right mouse button, and selecting the Edit Load Step menu item as shown in Figure 11.

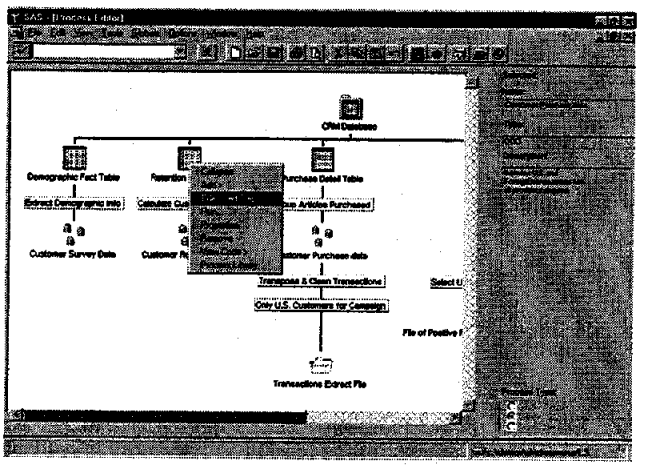

Figure 11: Defining the load step

After selecting Edit Load step from the menu, the Load Table Attributes window will appear. There are 4 tabs on the window.

- Source Code
- Execution
- Load Options
- Post Processing

The Execution tab (Figure 12) is the one of most significance for this example. In the execution tab you define where the process is going to be executed.

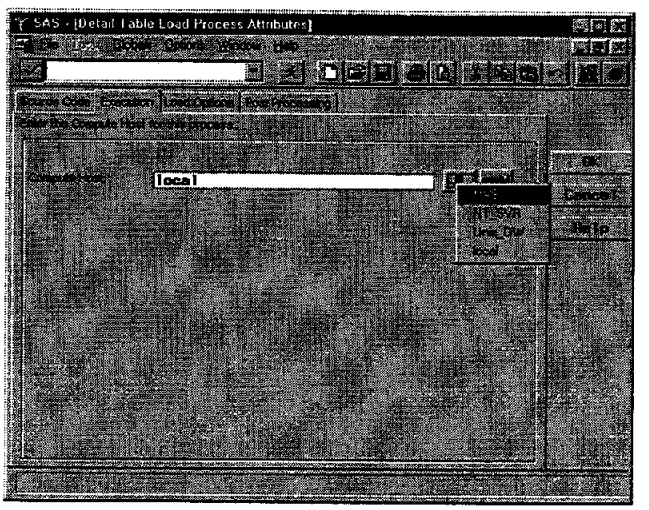

Figure 12: Load Attributes Window

In this example, we want the load step to execute on the MVS mainframe system.

The ETL of the data on the mainframe system is now complete. The SAS/Warehouse Administrator has developed the process on the Windows system to be executed on the mainframe. It will extract the data on the mainframe and move it to the data warehouse in Oracle on the Unix system.

Steps 3 and 4: Extracting/Loading Oracle on Unix

Steps 3 and 4 are conceptually the same as steps 1 and 2. Instead of extracting and cleansing the data out of DB2 tables and VSAM files on a mainframe system, the data is being extracted from an Oracle database and transformed on one Unix system. The data is then loaded into the Oracle data warehouse on a separate Unix system. Again the SAS/Warehouse Administrator manages the process to be executed on the Unix system from Windows.

It should be noted that the ETL processing could take place entirely on the SAS/Warehouse Administrator platform. The SAS/ACCESS Interface to Oracle software on the Windows system in conjunction with the Oracle Client software is capable of accessing the data on the transactional Unix system and loading the data into the Oracle data warehouse on the second Unix system in the example environment.

Some factors to consider when determining which configuration to use are:

- Is there network bandwidth to handle large volumes of data being moved between systems
- Is there CPU capacity to handle the transformation processing

If there is available network bandwidth to handle the large amount of data being transferred, then using the Oracle middleware solution is suitable. If there are available CPU cycles to perform the transformations, then it is possible to reduce the amount of data being transformed and loaded into the data warehouse and ultimately reduce the network traffic.

Step 5: Extracting SQL Server Data

Step 5 is different from the previous extraction steps. There is no SAS software installed on the Windows NT machine where the SQL Server database is installed. To access the data the SAS/ACCESS Interface to ODBC is installed on the Windows machine with the SAS/Warehouse Administrator. The SQL Server ODBC driver must also be set up to access the remote database server.

The SAS/ACCESS Interface to ODBC is used to create SQL Pass Through views to the SQL Server database. These views are then used as ODD's in the SAS/Warehouse Administrator as data sources for the warehousing process.

Since there are no SAS processing capabilities available on the SQL Server database server, the extraction and transformation process executes on the SAS/Warehouse Administrator platform. The data is moved between the two systems via the ODBC driver middleware.

Step 6: Loading the Oracle Data Warehouse

After the SQL Server data has been extracted and cleaned, the data is ready for loading into the Oracle data warehouse. This is accomplished using the SAS/ACCESS Interface to Oracle and the Oracle Client middleware.

The load step processing is the same as the previous two loading examples in Steps 2 and 4. The only difference is that the processing of the load step is performed on the SAS/Warehouse Administrator system. The Oracle Client Software moves the data across the network and loads it into the data warehouse.

CONCLUSION

The SAS/Warehouse Administrator coordinates the middleware products that are available to efficiently build and manage a multi-tier data warehouse. The flexibility of the SAS System enables it to access many different database formats as well incorporate virtually any computing platform into a data warehousing project.

REFERENCES

Boozer, Forrest (1995), Configuring and Using ODBC with SAS/ACCESS® Software, *Proceedings of the Twenty-First Annual SAS® Users Group Conference,* 21, 558-567.

SAS Institute Inc. (1997), SAS/Warehouse Administrator User's Guide, Release 1.1, First Edition.

The author may be contacted at:
Darrell Barton
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27511
Phone: (919) 677-8000
Fax: (919) 677-4444
Email: Darrell.Barton@sas.com

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. ® indicates a USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.