Paper 114-25

New Features in SAS/Warehouse Administrator™

Ken Wright, SAS Institute Inc., Cary, NC

ABSTRACT

SAS/Warehouse Administrator™ 2.0 introduces several powerful new features to assist in your data warehousing efforts. The Process Editor has been greatly enhanced to support the creation of multiple tables from a single step, the creation of multiple tables in a single job, the scoping of code generation through the concept of a job, and to maintain metadata about dependency relationships between jobs in your warehouse. Support for OLAP reporting needs has been integrated and greatly enhanced through the addition of new OLAP summary data stores.

INTRODUCTION

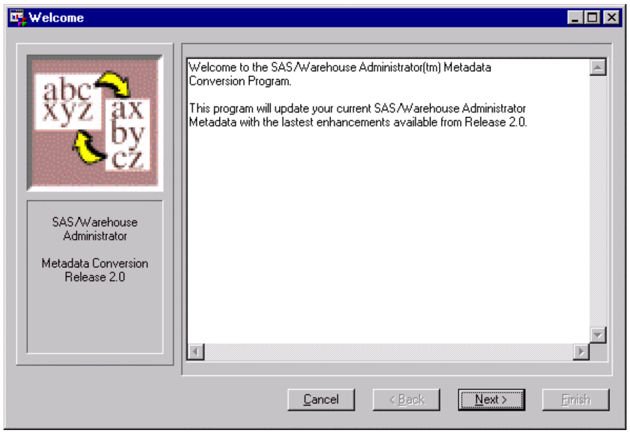
This document introduces the new features and changes in SAS/Warehouse Administrator that will better enable you to manage your warehouse. Several of these features existed in some form in Release 1.3, but are now more tightly integrated, easier to use, and incorporate a more comprehensive and accessible metadata model. For more information, see the SAS/Warehouse Administrator User's Guide or the online help.

Familiarity with Release 1.3 of SAS/Warehouse Administrator is assumed. In addition, basic knowledge of Online Analytical Processing (OLAP) techniques is also required for some features addressed.

METADATA CONVERSION WIZARD

If you have an existing warehouse built with a prior release of SAS/Warehouse Administrator, your first step with Release 2.0 will be to convert your metadata. This is a simple matter with the metadata conversion wizard.

When you first run Release 2.0 of SAS/Warehouse Administrator, you will not have your old warehouses already defined to the WA desktop, unless you converted your sasuser from your prior SAS release. Add a Data Warehouse Environment to the desktop, and define the path for this new environment to the location of your Release 1.3 Data Warehouse Environment. SAS/Warehouse Administrator will determine that your metadata needs to be converted, and the metadata conversion wizard will display (see figure 1).



Welcome to the SAS/Warehouse Administrator(tm) Metadata Conversion Program.

This program will update your current SAS/Warehouse Administrator Metadata with the lastest enhancements available from Release 2.0.

Figure 1: The Metadata Conversion Wizard

This wizard will step you through the process of defining your new Release 2.0 metadata repository locations, and upon completion will copy your old metadata into the specified repositories, modify the metadata as necessary, and add the necessary new metadata to support Release 2.0 of SAS/Warehouse Administrator. Be sure to specify new locations for your Release 2.0 metadata repositories. When the wizard is finished, you will have your new Data Warehouse Environment on your desktop. Open it and explore the new features in SAS/Warehouse Administrator!

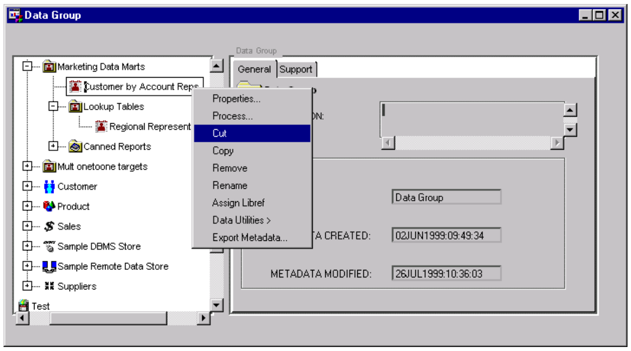
METADATA COPY WIZARD

There are many reasons you may want to make a copy of your entire Data Warehouse Environment. You may need a copy of the environment for remote access (a spoke environment), to make a snapshot of your warehouse, or as a way of moving metadata changes in your warehouse through a development/QA/production delivery cycle. In prior releases, you had to use OS utilities to copy your environment and warehouse, and your repositories had to be defined with relative paths in order to be accessible after relocation. With the new metadata copy wizard, you can copy your environment and warehouse into fully qualified repository locations, and rename them in one easy process.

To copy your environment, click its icon on the WA desktop with the right mouse button and select "Copy". The metadata copy wizard will display and step you through the process.

CUT AND PASTE

In the SAS/Warehouse Administrator explorer, you can now move a warehouse item to a new location in the explorer hierarchy via right mouse button (RMB) **Cut** and **Paste** actions. You must paste the cut item to another item in the same warehouse or environment, and the cut item must be a valid member type for that paste location. For instance, you can cut the "Customer by Account" data table in your warehouse, as in figure 2:



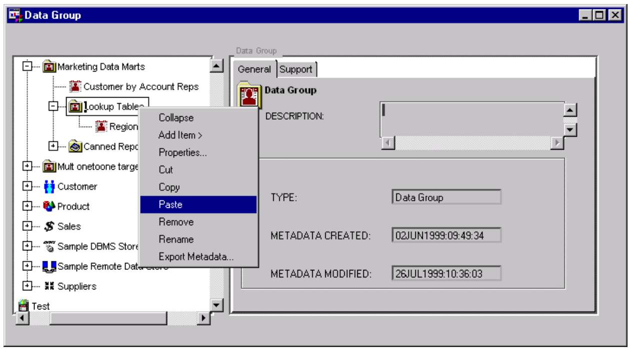
Data Group

TA CREATED: 02JUN1999:09:49:34

METADATA MODIFIED: 26JUL1999:10:36:03

Figure 2: **Cut** Customer by Account

And then paste it onto the "Lookup Tables" data group, as in figure 3:



DESCRIPTION:

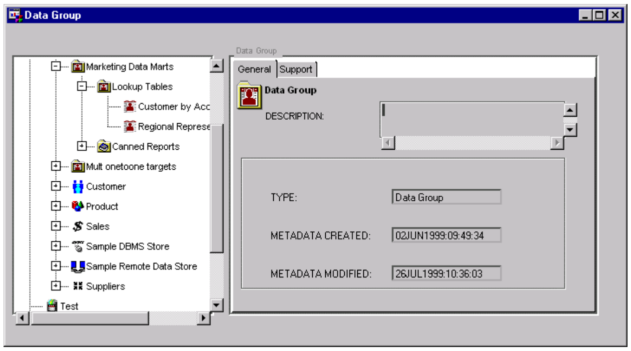
TYPE: Data Group

METADATA CREATED: 02JUN1999:09:49:34

METADATA MODIFIED: 26JUL1999:10:36:03

Figure 3: **Paste** onto Lookup Tables data group

The result will move the “Customer by Account” table to be a member of the ”Lookup Tables” data group, it will no longer be a member of the “Marketing Data Marts” group:



DESCRIPTION:

TYPE: Data Group

METADATA CREATED: 02JUN1999:09:49:34

METADATA MODIFIED: 26JUL1999:10:36:03

Figure 4: Result of cut/paste action

Attempting to paste the data table to an invalid parent, such as a subject, will produce an error. Moving an item that contains other items (such as a data group containing other data groups or data tables) will move the data group and all of its members. The cut and paste feature is also available in the tree view of the process editor.

COPY AND PASTE

The RMB **Copy** and **Paste** actions in the SAS/Warehouse Administrator explorer are used in the same way as the **Cut** and **Paste** actions described above. Select the desired item in the explorer with the RMB **Copy** action, and then **Paste** onto the desired parent item.

The results of a copy and paste action can be confusing at first, because the new object is not necessarily the same type of warehouse item as the copied object was. If a copied item is a valid member type for the paste location, the new object will be the same type of object as the item you copied (i.e. copying a data table and pasting it onto a data group will produce a new data table).

The potentially confusing behavior will occur when you paste onto an item that cannot contain that member type. Rather than not allowing a paste in this situation, SAS/Warehouse Administrator will create a new object that is a valid member of that group, and will copy the applicable properties from the initial object into the new object. For instance, a data table can be pasted onto a detail logical table to create a new detail table with the same properties (columns, table name, description, extended attributes, etc.) as the initial data table. This interpretation of a copy and paste action can seem confusing at first, but is actually an easy way to copy relevant metadata from one object type to another.

PROCESS EDITOR OVERVIEW

The Process Editor underwent a major overhaul for Release 2.0, and now truly enables you to see and navigate through all of the processes used to load the tables in your warehouse, all from one central location! No more switching back and forth between the process editor and the explorer to edit the processes for a different table. Access the Process Editor either via the Tools, Process Editor… pmenu, or by a RMB Process… selection on a table in your warehouse. The new Process Editor look incorporates a tree view on the left hand and a process flow diagram on the right (see figure 5).

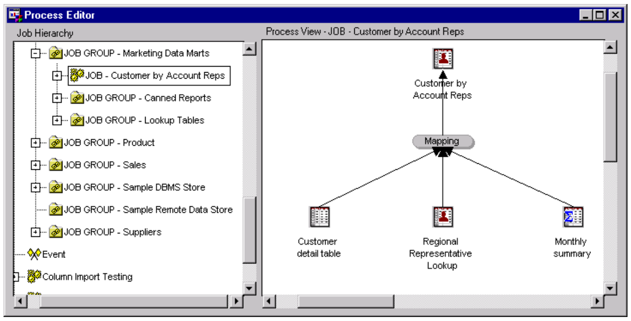

Process Editor
Job Hierarchy
JOB GROUP - Marketing Data Marts
JOB - Customer by Account Reps
JOB GROUP - Canned Reports
JOB GROUP - Lookup Tables
JOB GROUP - Product
JOB GROUP - Sales
JOB GROUP - Sample DBMS Store
JOB GROUP - Sample Remote Data Store
JOB GROUP - Suppliers
Event
Column Import Testing
Process View - JOB - Customer by Account Reps
Customer by Account Reps
Mapping
Customer detail table
Regional Representative Lookup
Monthly summary

Figure 5: The Process Editor

The tree view on the left is essentially an explorer for process related objects. The process objects in the tree view are the **Job Group**, **Job**, and **Event**, all of which are new objects in Release 2.0 of SAS/Warehouse Administrator. A Job Group is like a folder; it can contain other job groups, jobs, and events, and is used to organize your process editor hierarchy as you see fit. An Event is a warehouse object that can be used to describe real world conditions that must be satisfied prior to running a certain job in your warehouse, such as a return code value from a prior job or the existence of a certain file. A Job is a warehouse object that represents all of the processes that create any number of tables in your warehouse.

The process flow diagram on the right hand side of the Process Editor displays two views into your warehouse processes. They are the **Process View** and the **Job View**, and you can toggle between them via the View pmenu.

PROCESS VIEW

You will use the Process View to accomplish the same tasks you used the Process Editor for in earlier releases of SAS/Warehouse Administrator. The process flow diagram in the right panel of the new Process Editor will display the various processes used to extract, transform, and load the tables in your warehouse. While the basic activity of defining the processes is the same, there are several significant advances in the new version.

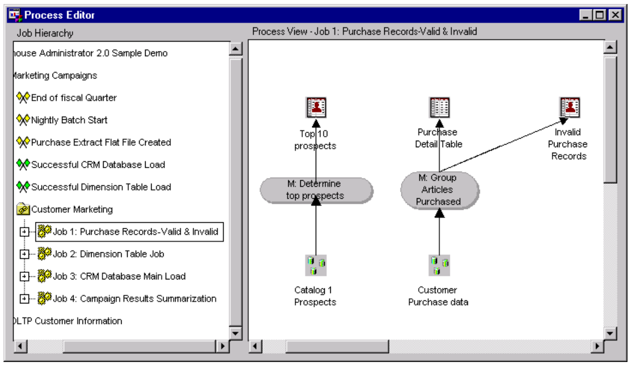

Process Editor
Job Hierarchy
ouse Administrator 2.0 Sample Demo
arketing Campaigns
End of fiscal Quarter
Nightly Batch Start
Purchase Extract Flat File Created
Successful CRM Database Load
Successful Dimension Table Load
Customer Marketing
Job 1: Purchase Records-Valid & Invalid
Job 2: Dimension Table Job
Job 3: CRM Database Main Load
Job 4: Campaign Results Summarization
LTP Customer Information
Process View - Job 1: Purchase Records-Valid & Invalid
Top 10 prospects
Purchase Detail Table
Invalid Purchase Records
M: Determine top prospects
M: Group Articles Purchased
Catalog 1 Prospects
Customer Purchase data

Figure 6: Purchase Records Job

The first major difference is the introduction of the concept of the Job, which allows you to group the processes for several tables together in order to generate and run their code together, as a single job. Figure 6 displays the process flow for the “Purchase Records – Valid & Invalid” job (notice it is selected in the Job

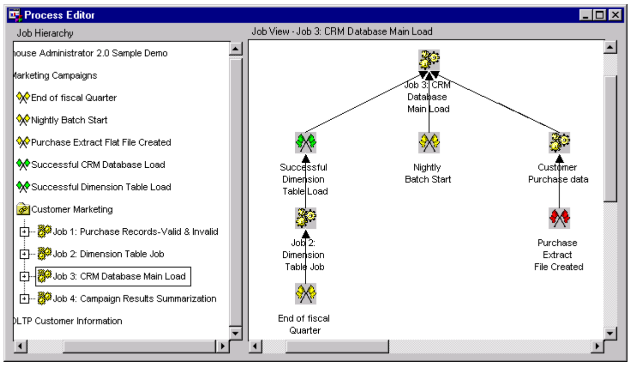
Hierarchy panel on the left side of the Process Editor). This job consists of all the processes involved in the creation and loading of three warehouse tables. The Purchase Detail table and the Invalid Purchase Records tables have a common process, but notice that the Top 10 Prospects table is not connected via the process flow to the other two. Code will be correctly generated in the right order for the various process steps involved in this job, regardless of whether the processes for the tables being created are related or not. The concept of a "job" in SAS/Warehouse Administrator enables the administrator of the warehouse to scope the processes and tables created in a single SAS program to include as few or as many tables as desired.

Another key feature evidenced in Figure 6 is the enhancement of the mapping process to allow for the creation of multiple output tables in a single process step. The mapping step "M: Group Articles Purchased" creates tables that feed into two output tables. This process will be examined closer below.

Design your process flow as you wish with RMB actions on nodes, arrows, or the background in the process flow diagram. In addition to the old functions such as adding, removing, and inserting processes, you can also zoom in and out, drill down into the job that creates an input table in the current job, and add new tables or merge existing jobs into the current one.

JOB VIEW

The Job View allows you to define the relationships between the jobs and events in your warehouse. Rather than looking **into** the processes inside the job selected in the left panel, the Job View looks **out** from the job to display its relationship to the other jobs and events in the warehouse. You can use the events and arrows in the job flow diagram to document the jobs and events that must occur prior to running the selected job.



ouse Administrator 2.0 Sample Demo
larketing Campaigns
End of fiscal Quarter
Nightly Batch Start
Purchase Extract Flat File Created
Successful CRM Database Load
Successful Dimension Table Load
Customer Marketing
Job 1: Purchase Records-Valid & Invalid
Job 2: Dimension Table Job
Job 3: CRM Database Main Load
Job 4: Campaign Results Summarization
LTP Customer Information

Job 3: CRM Database Main Load
Successful Dimension Table Load
Nightly Batch Start
Customer Purchase data
Job 2: Dimension Table Job
Purchase Extract File Created
End of fiscal Quarter

Figure 7: Job View – CRM database dependencies

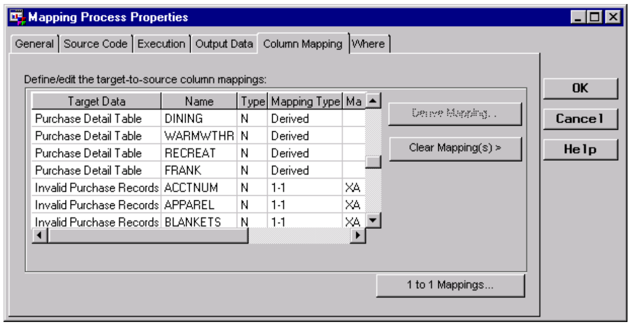
Figure 7 shows the Job View perspective of the "Job 3: CRM Database Main Load" job. This job flow shows that before the CRM database load is run, the Dimension Table load should be successful, the Nightly Batch window should have started, and the Customer Purchase data job should have run.

The information you document in the Job View to describe the dependency relationships surrounding your jobs can be used to create a report that you use to schedule the job execution with your scheduling facility. Alternately, you could use the Metadata API to read the Job Flow information and generate the necessary conditional processing code to drive the execution of the jobs in your warehouse.

MAPPING SUPPORT FOR MULTIPLE OUTPUT TABLES

Figure 6 shows the use of the mapping process to create multiple tables that feed into the load processes for the "Purchase Detail Table" and the "Invalid Purchase Records" tables. Adding an additional output to a mapping is as easy as 1, 2, 3. One: select RMB Add> on the mapping node. Two: select Outputs… from the ensuing popup. Three: select the desired target tables from the selector window.

Additional Mapping Properties have been added to support the creation of multiple output tables in the mapping step. Once you have selected the output tables for that mapping, edit the properties of the mapping as usual. The Column Mapping tab will now show all of the Columns in all of your target tables (see Figure 8). Define the necessary mappings to the appropriate columns in your target tables.

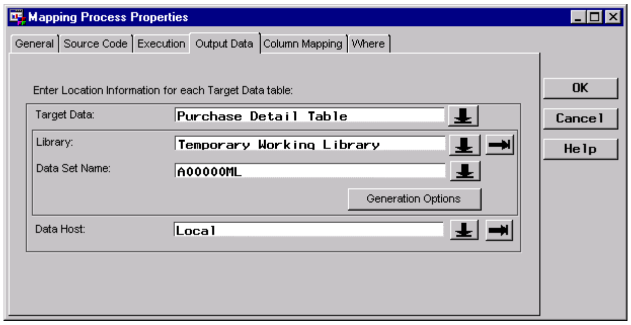


Define/edit the target-to-source column mappings:

Target Data	Name	Type	Mapping Type	Ma
Purchase Detail Table	DINING	N	Derived	
Purchase Detail Table	WARMWTHR	N	Derived	
Purchase Detail Table	RECREAT	N	Derived	
Purchase Detail Table	FRANK	N	Derived	
Invalid Purchase Records	ACCTNUM	N	1-1	XA
Invalid Purchase Records	APPAREL	N	1-1	XA
Invalid Purchase Records	BLANKETS	N	1-1	XA

Figure 8: Column Mappings for multiple target tables

Notice the Table Options tab is no longer on the Mapping Process Properties window. Since the mapping can now have multiple targets, the Output Data tab has been modified to provide target table-specific access to table options and to enable you to define the row selection criteria for that target table (see figure 9).



Enter Location Information for each Target Data table:

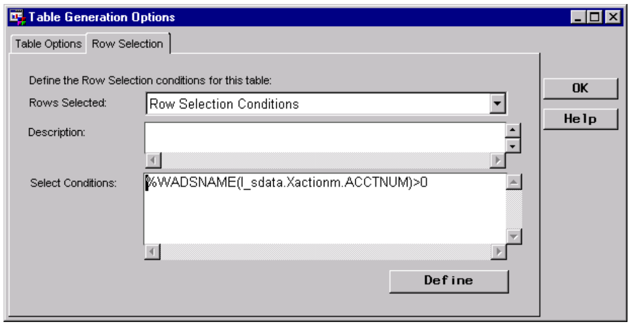
Target Data: Purchase Detail Table
Library: Temporary Working Library
Data Set Name: A00000ML
Data Host: Local

Figure 9: Output Data tab

A mapping with multiple outputs will create several working tables via a SAS data step (mappings with one output usually produce SQL views). Each working table will be used as input to the next process in the creation of one of the target tables. There is a one to one correlation between the number of working tables created by the mapping and the number of target tables (such as Data or Detail tables in your warehouse) the mapping feeds. The Output Data tab allows you to edit the basic properties about one of those working tables at a time. The Target Data field gives you a reference point that tells you which target table is associated with the working table whose properties you are editing, and allows you to toggle between the different target table and work table pairs.

In figure 9, the Target Data field shows Purchase Detail Table (the other valid selection via the down arrow would be Invalid Purchase Records). So the other fields on this tab, as well as the Table Generation Options window accessed via the Generation Options button, apply to the table created by this process that will be input in the load of the Purchase Detail Table. If you toggle the Target Data value to Invalid Purchase Records, the Library and Data Set Name values would change to reflect the other table this mapping creates. The Data Host field is offset from the other working table specific fields, even though it certainly is a property of the working table, because the host value specified here is common to all of the working tables created by this process.

So now you need to subset the data that feeds into these two target tables, one to hold valid records and one for invalid records. Select the Generation Options button to modify the options for the A00000ML (valid records) working table.



Define the Row Selection conditions for this table:

Rows Selected: Row Selection Conditions

Description:

Select Conditions: %WADSNAME(I_sdata.Xactionm.ACCTNUM)>0

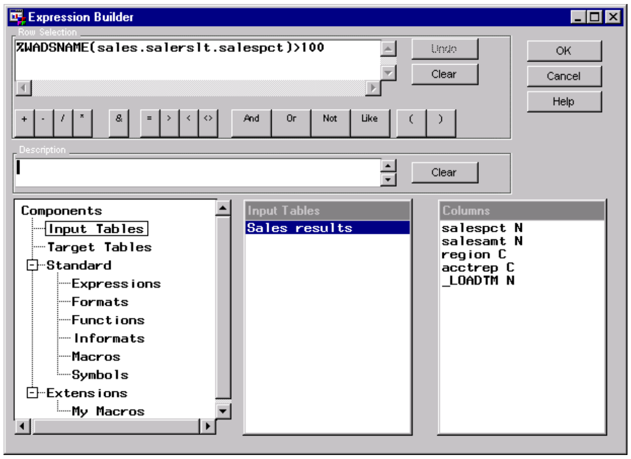
Figure 10: Row Selection criteria

The Table Options tab has been moved from the Mapping Properties window to the Table Generation Options window, but the real news is the Row Selection tab. This tab allows you to specify the selection criteria for rows for this working table. The default value in the Rows Selected field is All Rows, meaning that all rows input to the data step will be output to this tables. You have two options to subset the data, either choose Row Selection Conditions or User Defined Statements. The difference is subtle, but important.

The subset criteria you define here will be applied in a SAS data step, so the code generated (or entered) needs to be valid data step code. If you select Row Selection Conditions, all that you need to enter is the condition for selection. You can use the Define button to access the Expression Builder for a point and click interface to create the condition. This is the easier option, because you do not need to write your own data step syntax to subset your data, you just specify the condition and SAS/Warehouse Administrator will generate the proper data step code to apply it. However, if necessary, you can leverage the full power of the SAS data step by selecting User Defined Statements and defining your own SAS data step statements. When the mapping code is generated your code will be included into the data step as is, so you can subset data for multiple tables using if-then-else logic, etc. If you select this option, be sure to include your own output statements.

EXPRESSION BUILDER CHANGES

The expression builder is now accessible from the Row Selection tab as well as the Where tab of the Mapping Properties window and the Record Selection tab in the Record Selection properties window. You can now add your own customized expression components to the standard list of components provided in the Expression Builder.   The standard expression components in this window come from the SAS data set SASHELP.WAXFORM. You can create your own customized list of components by creating an _SASWA.WAXFORM table with the same format, and filling in the appropriate values and code for your expression component. These components will appear under the Extensions label in the Components list. In figure 11 you see the customized category of My Macros.



```
%WADSNAME(sales.salerslt.salespct)>100
```

Sales results

```
salespct N
salesamt N
region C
acctrep C
_LOADTM N
```

Figure 11: Customized Expression Builder

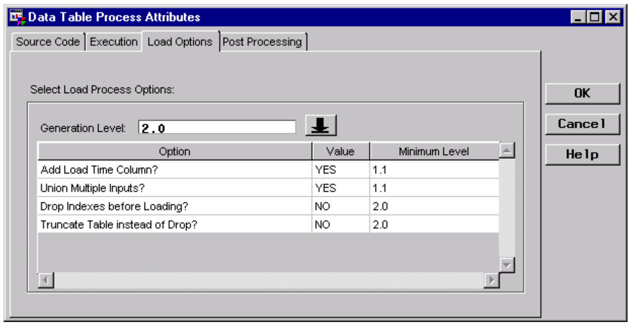
In addition to enabling you to customize your own expression components, the Expression Builder has two other changes that are seen only when accessed via the Row Selection tab in the Table Generation Options window. Under the list of components in Figure 11, the first two headings are Input Tables and Target Tables, which can be used to select the column names that you want to insert into your expression. The Input Tables list is available at all times in the Expression Builder, as in prior releases. The Target Tables heading is only available when accessed via the Row Selection tab.

When defining your subset criteria, you may only need to use variables that are available in the tables that are input to the mapping process. If so, use the Input Tables heading to show that list of columns and to select the desired columns for your transformation. However, your subset criteria may be more complex; you may want to subset your data using columns that are being transformed in this mapping. Selecting Target Tables will give you access to the transformed columns being created in this mapping, so you can subset with your transformed values.

The final note about the Expression Builder is really a note about any code entered or generated for the Row Selection tab. Notice in figure 11 that the code in the Row Selection uses a macro in the form of %WADSNAME(libname.tablename.column). The SAS data step does not handle multiple variables with the same name like SQL, and SAS/Warehouse Administrator will use the data step to create the multiple output tables. To be able to handle multiple input or target tables with possible column name conflicts, this macro is used to identify the library and table and column, and the generated code will resolve the macro to the appropriate column name in the data step. If you have a possible name conflict in your input or target tables, and are writing the code yourself rather than using the point and click method of selecting columns, you will need to use the %WADSNAME() macro.

DATA LOADING ENHANCEMENTS

There are several new options available to improve the performance of loading a warehouse table. Edit the load step of a table in your warehouse, and select the Load Options tab, it will now look something like this:



Select Load Process Options:

Generation Level: 2.0

Option	Value	Minimum Level
Add Load Time Column?	YES	1.1
Union Multiple Inputs?	YES	1.1
Drop Indexes before Loading?	NO	2.0
Truncate Table instead of Drop?	NO	2.0

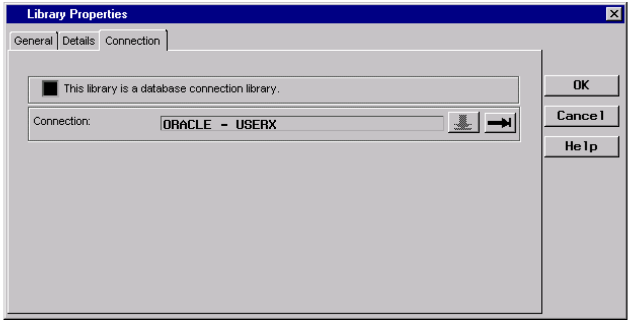
Figure 12: Load Options

The Code Generation Level is used to determine if you want to take advantage of some of the newer features in SAS and SAS/Warehouse Administrator when generating the code for this table. There are currently 2 valid values for the generation level, 1.1 and 2.0. A value of 1.1 will limit the options available, and it will generate load code just like prior releases of SAS/Warehouse Administrator, to be run with version 6 of SAS. This value should be used if you are still using version 6 of SAS to load this warehouse table. Selecting a value of 2.0 will allow you to take advantage of the new code generation options described below, as well as some of the newer features in SAS such as the DBMS libname engine support. Use this value only if you are loading your tables with SAS version 7 or higher.

The "Drop Indexes Before Loading?" option specifies that any existing indexes on the table to be loaded should be dropped before loading the data into the table. The indexes will be recreated after loading the data. This option can provide a significant performance boost when updating the indexes while loading is too slow.

The "Truncate Table Instead of Drop?" option specifies that when refreshing the data in a table, the table should be truncated (all the rows are removed but the table is not deleted) instead of completely dropping the table and recreating it. This option is useful when the table has many options, privileges, and other characteristics defined in the database.

SAS/Warehouse Administrator Release 2.0 also supports the use of the SAS/ACCESS® engines to access your DBMS via the libname statement. The DBMS libname engines in V7/V8 of SAS can greatly simplify access to your DBMS tables. The new Connection tab on the Library Properties window (see figure 13) enables you to specify that this library definition is a database connection library. You can then associate the library to a Connection you have defined to SAS/Warehouse Administrator (or create a new one). When a table associated to this library is accessed or code is generated for that table, SAS/Warehouse Administrator will generate the correct libname syntax using the options you have defined for the associated Connection.



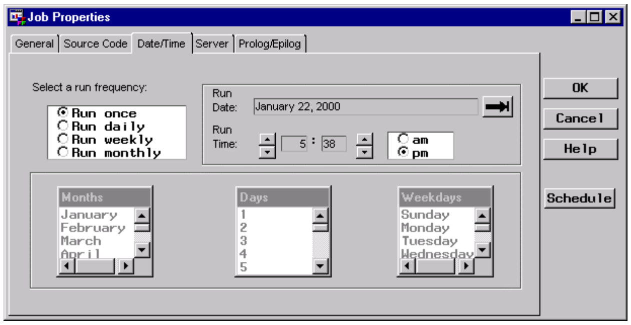
■ This library is a database connection library.

Connection: ORACLE - USERX

Figure 13: Database Connection library definition

INTEGRATED JOB SCHEDULER

Edit the properties of a job with a RMB Properties selection on a Job icon in the left panel of the new Process Editor. Properties you have come to expect in warehouse objects, such as name and description and whether the code is User Supplied or generated by SAS/Warehouse Administrator, can be defined here for Jobs as well. In addition, the Job Scheduler, available in earlier releases as a separate tool, has been enhanced and incorporated into the Job Properties:



Select a run frequency:

● Run once
○ Run daily
○ Run weekly
○ Run monthly

Run Date: January 22, 2000

Run Time: 5 : 38 ○ am ● pm

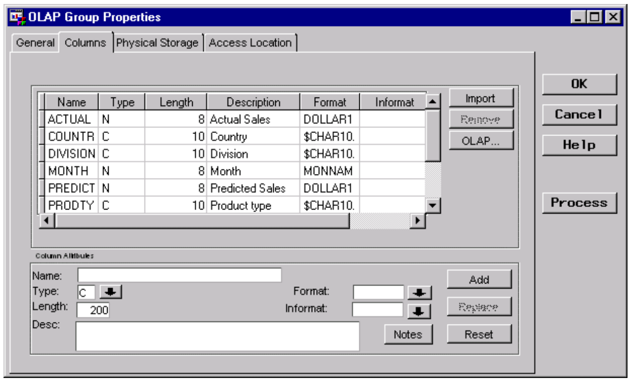
Figure 14: Job Properties Date/Time tab

The Date/Time tab displayed in figure 14 can be used to specify the date and time the job should be scheduled, as well as the recurrence criteria. The Server tab allows you to choose one of three types of scheduling servers supported for direct scheduling from SAS/Warehouse Administrator, the Cron, AT, and Null schedulers. The Cron and AT servers were available in the release 1.3 Job Scheduler, but the Null server is new. The Null server does not actually schedule the job to be run, but is actually an interface to save the generated code to a specified location and file name, and to create a command file as well with the appropriate commands to run the job. These files can be placed in a specified location on a remote host to be used by your standard scheduling system.

INTEGRATED OLAP SUPPORT

In Release 2.0 of SAS/Warehouse Administrator OLAP Groups, OLAP Tables, and OLAP MDDBs are new summary data stores that replace Summary Groups, Summary Tables, and Summary MDDBs. You cannot create new Summary Groups in this release, you will need to create an OLAP Group instead. Existing Summary Groups will still be supported. These new OLAP stores are more flexible and designed to support different types of OLAP reporting models such as ROLAP (relational OLAP), MOLAP (multidimensional OLAP), and HOLAP (hybrid OLAP). OLAP Groups can be added to Subjects in the SAS/Warehouse Administrator explorer, and OLAP Tables and MDDBs can be added to OLAP Groups.

OLAP Groups have a type attribute that you can specify as HOLAP, ROLAP, MOLAP, or MIXED. The MIXED type indicates that the group can contain both OLAP Tables and OLAP MDDBs that are not intended to be used together. A HOLAP type OLAP Group is more than just a grouping mechanism for summary data stores, it is actually a summary data store itself and serves as a proxy MDDB when used by the proper reporting tools. When the OLAP Group metadata is exported to an EIS metabase, the export process will register the necessary information in the metabase so the OLAP reporting tools will take advantage of the HOLAP structure.

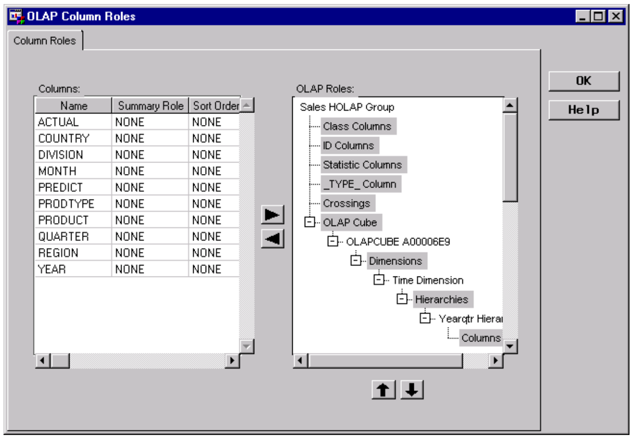


Name	Type	Length	Description	Format	Informat
ACTUAL	N	8	Actual Sales	DOLLAR1	
COUNTR	C	10	Country	$CHAR10.	
DIVISION	C	10	Division	$CHAR10.	
MONTH	N	8	Month	MONNAM	
PREDICT	N	8	Predicted Sales	DOLLAR1	
PRODTY	C	10	Product type	$CHAR10.	

Name:
Type: C
Length: 200
Desc:
Format:
Informat:

Figure 15: Columns tab for OLAP data stores

All of these new summary data stores have the same column information that is registered for other data stores such as detail tables, but there are additional OLAP specific characteristics that can be defined for them by selecting the OLAP… button on the Columns tab, shown in figure 15. This will display the OLAP Column Roles window:



Name	Summary Role	Sort Order
ACTUAL	NONE	NONE
COUNTRY	NONE	NONE
DIVISION	NONE	NONE
MONTH	NONE	NONE
PREDICT	NONE	NONE
PRODTYPE	NONE	NONE
PRODUCT	NONE	NONE
QUARTER	NONE	NONE
REGION	NONE	NONE
YEAR	NONE	NONE

Figure 16: OLAP Column Roles

From this OLAP Column Roles window you can define the various OLAP characteristics of your individual OLAP Table or OLAP MDDB, or in the case of an OLAP Group of type HOLAP, you can define the entire set of those characteristics for all tables and MDDBs in the group. In the right panel of figure 16 you see the various types of OLAP characteristics that can be defined for your columns (Class, ID, Statistic, and _TYPE_) and the OLAP structures that are defined to group the columns and other structures (Crossings, Cubes, Dimensions, and Hierarchies). Defining the OLAP characteristics for columns and building the OLAP structures is a simple matter of selecting the desired columns from the left panel, selecting the characteristic or structure on the right, and pressing the right arrow control in between.

Other features can also be used from this window to help you build your various OLAP crossing definitions based on the Class columns, Dimensions, and Hierarchies you have defined. Once the OLAP definitions are in place for the group, you can Edit the Column Roles for tables and MDDBs in that group, and import the subset of OLAP definitions that applies to that table or MDDB, you will not need to repeat the process.

These new summary data stores enable you to register all of the OLAP specific metadata for any of these types of OLAP models. In the Process Editor you can define the processes used to create the tables or MDDBs, including transformations for columns being created. SAS/Warehouse Administrator will generate the code to create and load your tables and MDDBs (though as always you can register your own code). And finally, you can export the OLAP Group metadata to your SAS/EIS metabase with a RMB Export Metadata action on that group in the explorer.

CONCLUSION

The advances made in the Process Editor allow complete control over the extraction, transformation, and loading processes that build your data warehouse. The integrated OLAP support allows you to more completely describe and build summary data stores to support the reporting processes that motivate the creation of the warehouse. The extended data loading and code generation features reduce the human effort involved in writing the code to build the warehouse. The combination of these new features in SAS/Warehouse Administrator, in concert with the rest of the Version 8 SAS® System, make building, maintaining, and exploiting your warehouse easier than ever!

REFERENCES

SAS Institute Inc., SAS/Warehouse Administrator User’s Guide, Release 2.0, First Edition, Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

I would like to thank the following people for their assistance with this paper:

Frank Nauta
Kevin Myers
Dave Russo

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Ken Wright
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Work Phone: 919-677-8000
Fax: 919-677-4444
Email: Ken.Wright@sas.com