

Paper 093-2009

## What's New in SAS® Data Integration Studio 4.2

Eric Hunley and Nancy Rausch, SAS Institute Inc., Cary, NC

### ABSTRACT

SAS® Data Integration Studio 4.2 provides many new enhancements to help both data warehouse developers and data integration specialists carry out the data integration process more efficiently and with greater control and flexibility. A major focus of this release is to deliver new visualization features and integrated debugging support. Major improvements have been made to nearly every transform node. Performance analysis and optimization capabilities have been enhanced through the use of integration with ARM facilities. Increased integration across the platform is another focus area that provides a better integrated user experience across the platform suite of products. Enhancements coming in version 4.21 will also be discussed. Customers will find many reasons to upgrade to SAS Data Integration Studio 4.2.

### INTRODUCTION

SAS Data Integration Studio is a powerful visual-design tool that is used to build, execute, and maintain data integration projects, from building an enterprise data warehouse to migrating data from applications like SAP. SAS Data Integration Studio simplifies and speeds project development with an easy-to-use interface, extensive built-in transformations, and powerful productivity capabilities, while providing a single point of control for managing complex enterprise data-integration processes. SAS Data Integration Studio is easy to learn, collaborative, and lets you build reusable processes to speed data integration development both now and in the future.

SAS Data Integration Studio is the successor to SAS® ETL Studio and SAS/Warehouse Administrator®. Before SAS/Warehouse Administrator, many SAS users solved data-integration challenges by using Base SAS® software to create their own applications. Over time, the demands and expectations changed and the need for tools that increase productivity and capture related metadata became a requirement throughout the industry.

History illustrates that data management and data integration capabilities grew up from a do-it-yourself, fundamental approach to integrated tools-based solutions. Today, organizations expect tools and solutions that anticipate their needs and provide capabilities or services that can complete the same tasks as the past, but with more elegance, ease, and precision. Today's world brings challenges or regulations that might not have existed 15, 10, or even 5 years ago. There is more data than ever (some say doubling every 12 to 18 months), service-level agreements that must be met, requirements to maintain compliance from both the business and IT perspectives, and the ever-present cliché of doing more with less. Organizations are expected to leverage existing investments and provide tighter integration between third-party components. It is becoming an expectation especially with having more control or pushing more logic into the databases for processing. Minimizing data movement and using the parallel processing capabilities of the Database Management Systems and Data Appliances has tremendous benefits in areas of performance, scalability and securing data assets.

All these challenges continue to place higher expectations on the tools and solutions that are designed to deliver the right data to the right people at the right time. This is true for SAS Data Integration Studio as well. Developers and users require products that enable them to work as efficiently and effectively as possible, in every stage from design through maintenance.

This paper highlights the new features and capabilities in SAS Data Integration Studio 4.2 and 4.21 to meet these challenges. It discusses how it can help you meet the expectations and requirements that we face in the data integration world of today. We will cover three main areas: improved user experience (including debugging capabilities and performance enhancements), exploitation of the SAS® 9.2 Platform, and transformation improvements and additions.

### IMPROVED USER EXPERIENCE

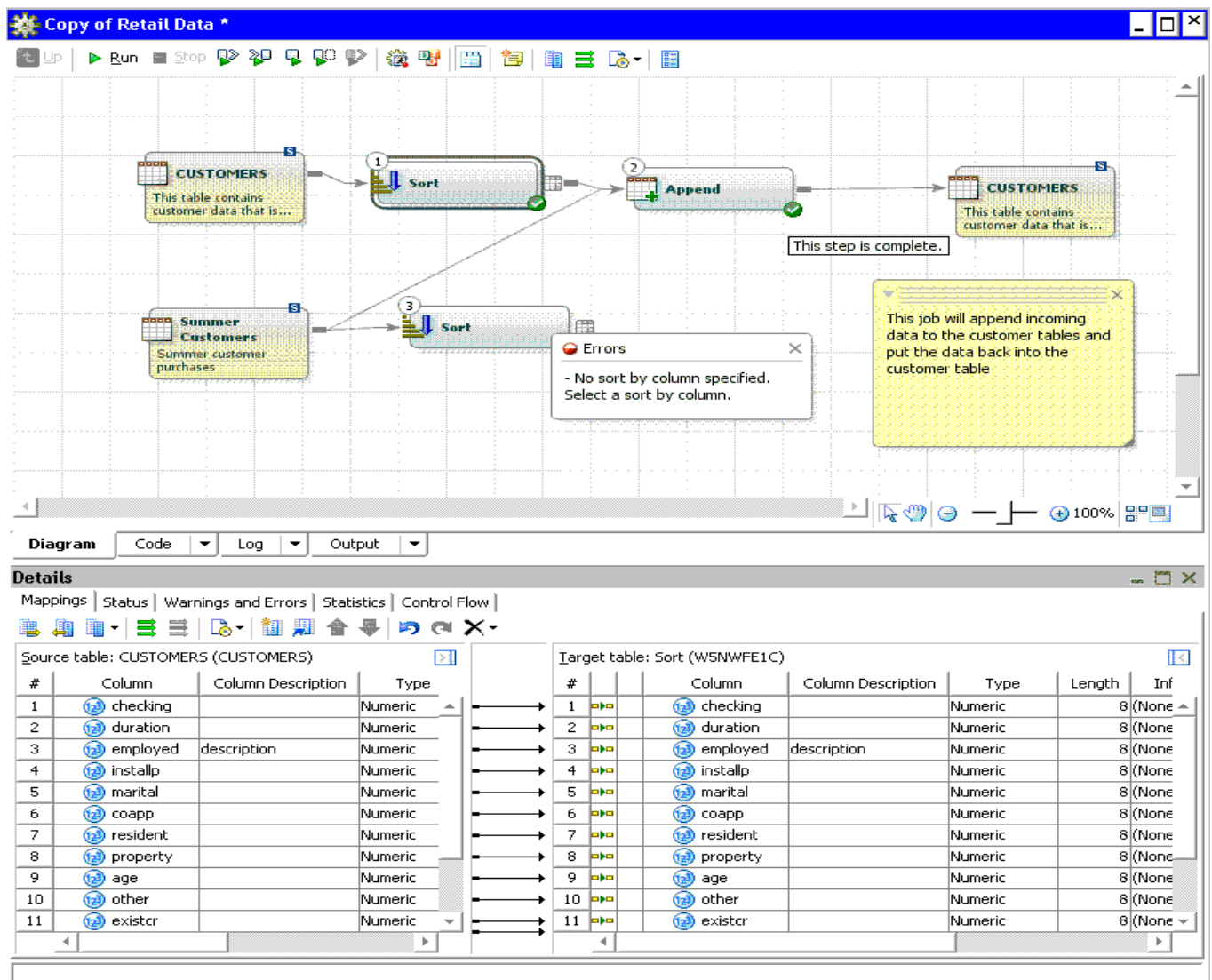
One of the major workflow enhancements that is available in SAS Data integration Studio is an updated job editor. The job editor has been significantly redesigned to make the process of developing high-performing data integration flows easier and faster. The job editor also includes an integrated debugger that can help you identify and resolve design, run-time, and performance issues in your data integration flows as early as possible. Key features in the job editor include:

- visual design for SAS code

- configurable mapping rules
- integrated debugger
- early detection of design errors
- ability to submit individual steps, run even when a job is not complete, and view intermediate results
- run-time progress indicators and status
- detailed performance, warning, and error information
- control over node-execution order
- source/target table type indicators
- transformation push-down indicators that specify when a node will be pushed down to a database
- checkpoint restart support
- enhanced expression builder

You launch the job editor from the **New Job** menu or by selecting **Open** on an existing job. The **Diagram** tab in the upper half of the editor window displays a visual representation of the steps that are contained in the job. You build jobs by dragging and dropping data objects, such as tables and external files, into the diagram area. You can add transformations such as sorts, joins, and loads from a transformation library and draw arrows to connect the objects together. This process is used to produce the results that you need.

Display 1 shows an example of an open job in the job editor.

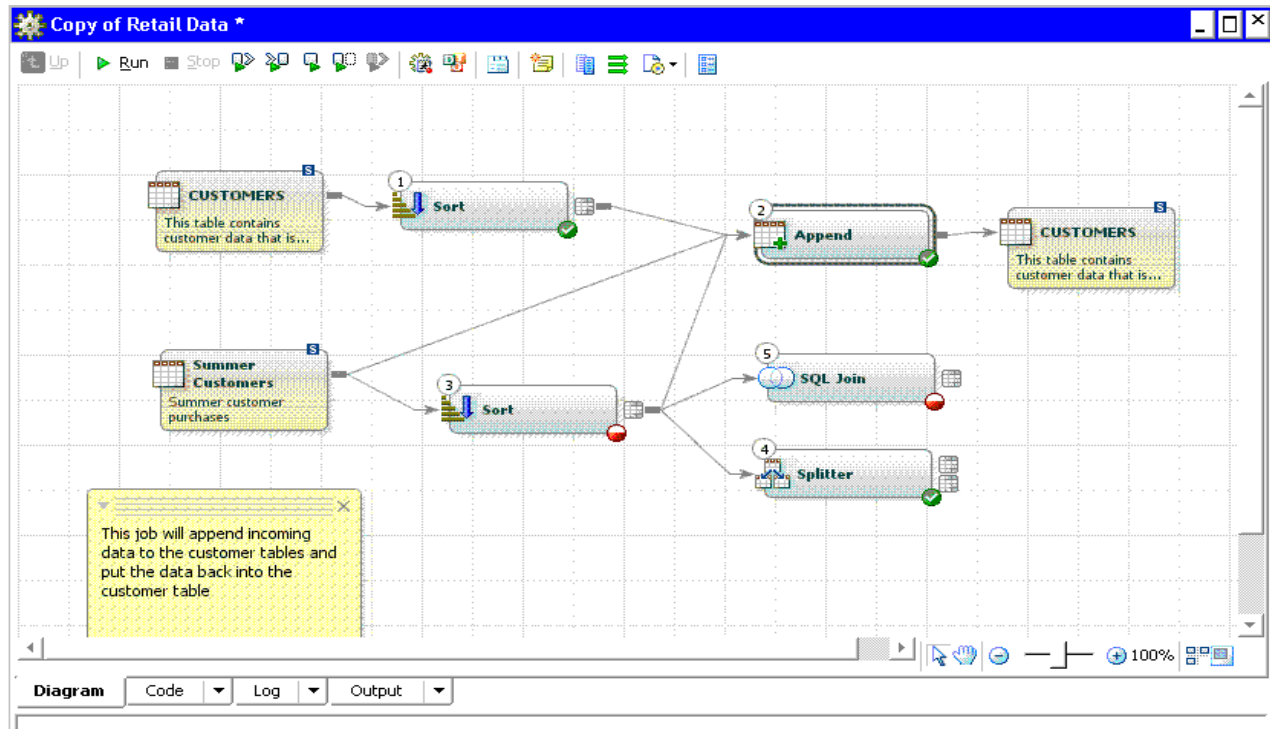


**Display 1. An Example of an Open Job in the Job Editor**

The job editor has several tabs and panels that help you to visualize your data integration flow. The diagram panel includes annotated data and transformation nodes. Further annotated comments can be placed in sticky notes in the

diagram. The status of each node is displayed as well. Display 1 shows that the Sort transformations are incomplete (half circle in lower right of transformation node), while the Append transformation has all the needed information to run (check mark on the node).

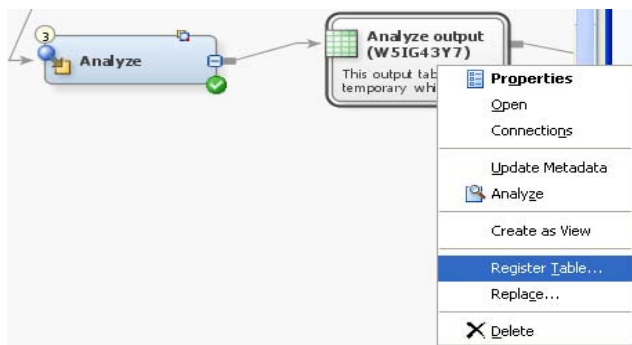
New in SAS Data Integration Studio is the ability to use objects such as tables in a diagram multiple times. For example, a table can be a source to multiple transformations in a job and can be both source and target in the same job. This opens up a new world of table updates, appends, and replacements that might be needed in some data integration scenarios. An icon on each table node indicates the database type of the table and that enables you to see at a glance where your tables are physically located. This helps when you are trying to optimize your flows for performance by showing you where data is being transferred between different database systems. *Intermediate tables*, also known as work tables, are also shown on the **Diagram** panel, and can be connected to multiple nodes in the job as shown in Display 2.



**Display 2. Connecting Tables to Multiple Job Nodes**

In this diagram, the CUSTOMERS table is both source and target of the flow. In addition, output from a Sort transformation feeds into three subsequent steps. The tables are SAS tables, indicated by the type icon on the top right hand side of the table nodes.

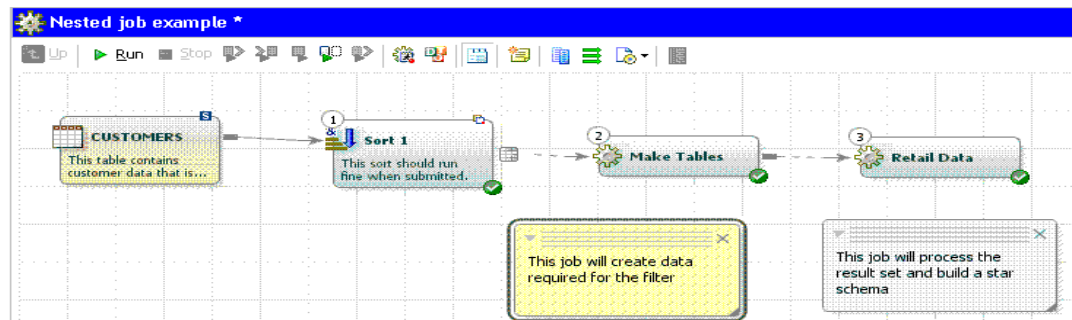
The job editor makes it easy to build complexity as you go along. For example, you might know some information about your source tables, but have not yet designed your target tables. You can work with temporary tables until you obtain the results you want. You can create a permanent table from the temporary table as shown in Display 3, run an update process to read the physical structure of the table and update the metadata about it, or replace a temporary table with an existing physical definition. These update techniques retain any mapping relationships the prior table might have had, saving time in the design process.



### Display 3. Creating a Permanent Table

The Register Table technique is just one of several techniques available that enable you to easily modify table structures as you build your jobs.

Also new in SAS Data Integration Studio, jobs can be embedded or nested more easily inside of other jobs. For example, you can build a job flow, and then test and validate that it is working as you expected by seeing that it produces the results you want. Later, you can embed that job into a second job like a standard transformation, as shown in Display 4.



### Display 4. Embedding a Job into Another Job

The diagram nodes, **MakeTables** and **Retail Data**, are data integration jobs that are part of the larger job called **Nested Job**. Sticky notes in the diagram document the overall flow.

When you are working with objects in a job, it is helpful to see information about an object like a transformation without having to open the property window for it. The Details panel shown in the bottom half of the job editor contains information about a transformation, such as the column mappings present in the selected transformation.

Details

Mappings

Status

Warnings and Errors

Statistics

Control Flow

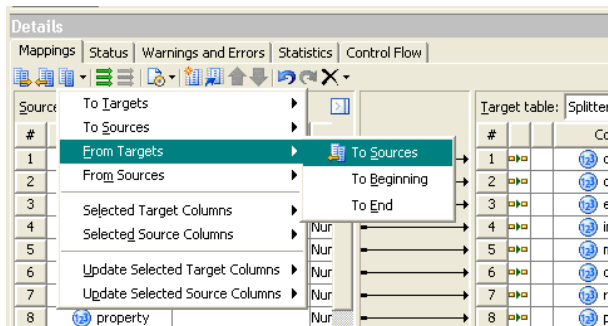
</

Display 5. Mapping details panel showing source to target mappings

The Details panel shows information such as source to target mappings. This lower panel can be viewed at the same time as the upper Diagram panel, simplifying complex mapping design.

## MAPPINGS

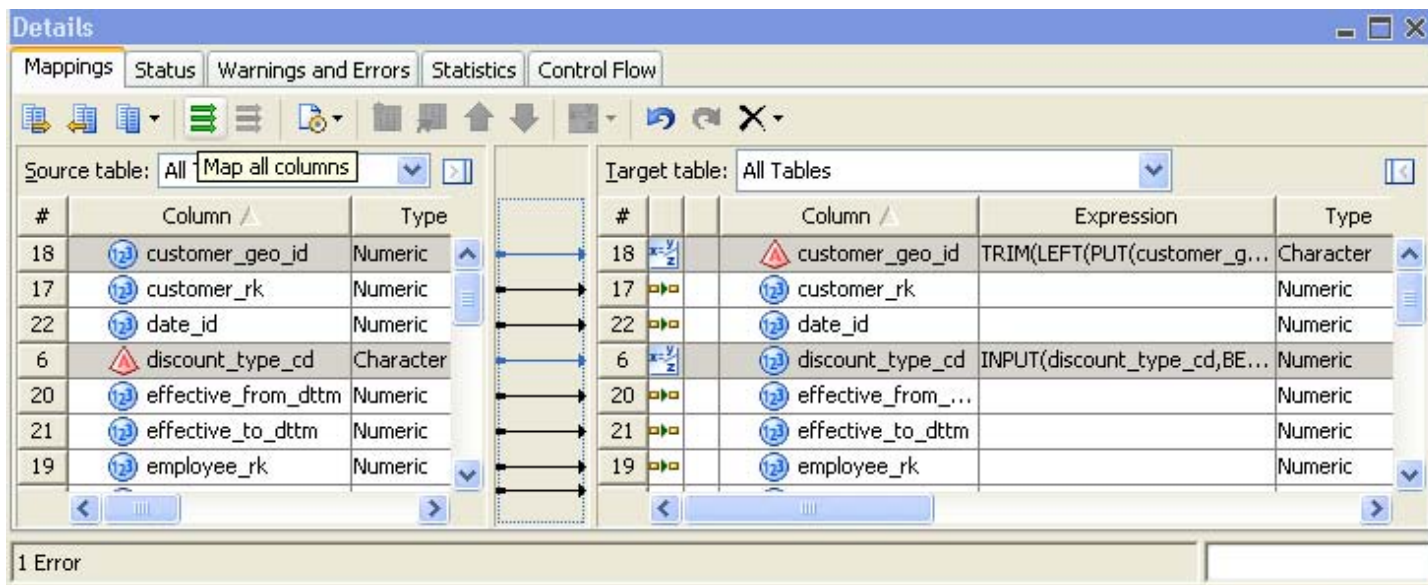
In any transformation, the Mappings panel enables you to define the propagation of information from table to table in your job. You can create propagation mappings that originate before or after the transformation, or they can come from other places such as the next beginning or ending table in a job. For example, if a new column is added to a table that is used at the beginning of a job, it can be easily propagated to other tables in the job. Changes can also be propagated through the columns in the job.



**Display 6. Mappings and Propagations Specified in a Mapping Panel**

The Mappings panel is also used to describe expressions that map source and target columns together. For example, if the measurement unit changes in a job, a calculation should be performed. An expression that converts English units to metric units (Fahrenheit to Celsius) can be used to perform this conversion automatically when columns like TEMP\_F and TEMP\_C are present in tables.

The Mappings panel uses a rules file to determine how to perform column matching and conversion expressions. This rules file is user-configurable so that you can define rules that match columns on patterns, and you can create expressions to map source columns to target columns wherever they occur. The default rules file contains several rules, including an expression to map numeric-to-character columns and character-to-numeric columns.

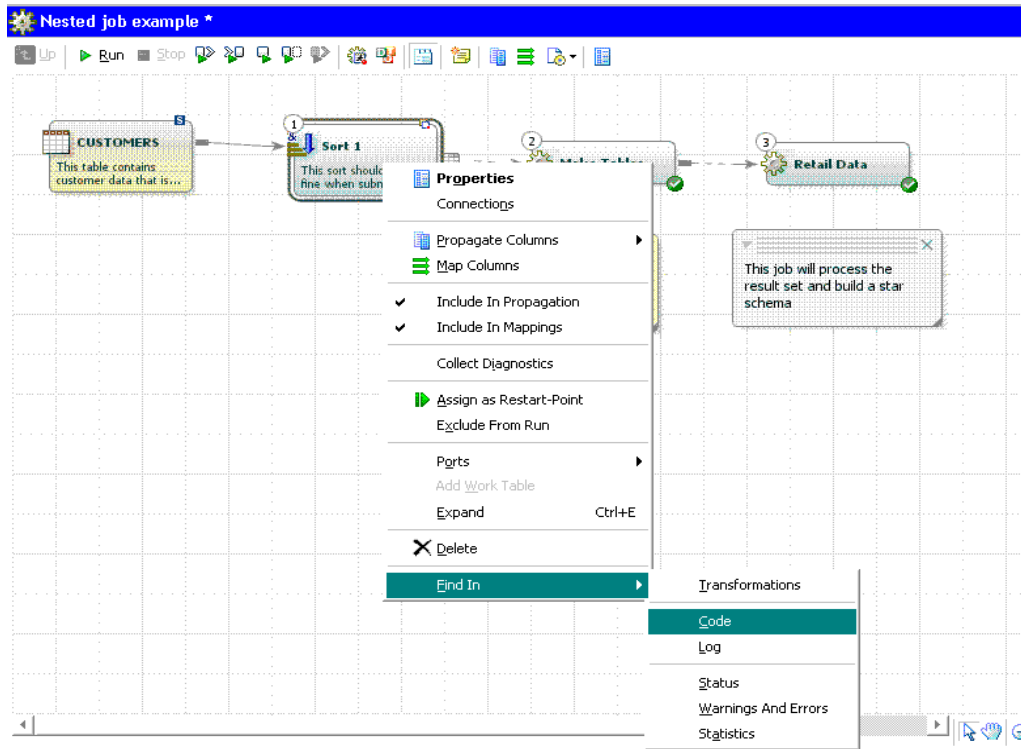


**Display 7. Example of the Default Mapping Rules Being Applied When Mapping Numeric-to-Character Columns and Character-to-Numeric Columns**

Default mapping rules are predefined and can be extended to meet business-specific needs. A number of search capabilities have been added to SAS Data Integration Studio to make it easier to locate items in the application. Display 8 illustrates some of these available search options. Any table can be located in the main folders tree by



using the **Find in** menu on the table. Any transformation can be located in the Transformations tree. The code that the transformation generates can be located in the **Code** tab, and the log results it produces can be located in the **Log** tab. New panels that show warnings, errors, and performance statistics also show linked information based on the run-time behavior of the selected transformation.



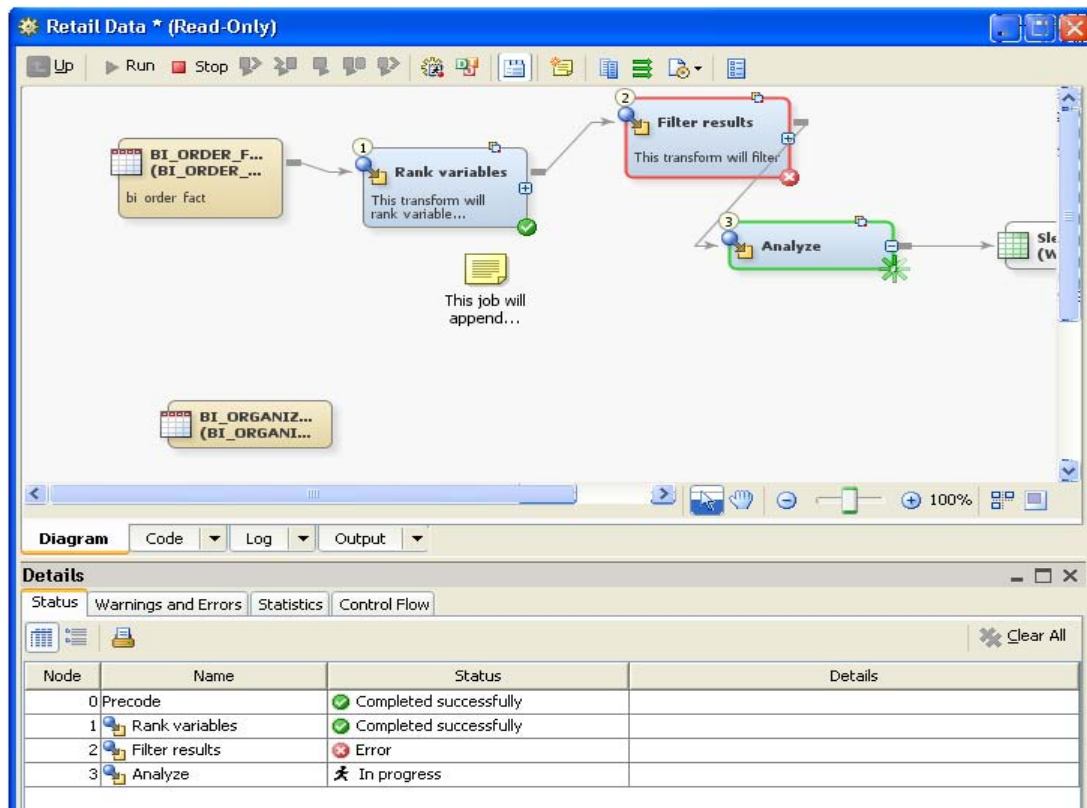
## Display 8. New Search Capabilities in SAS® Data Integration Studio

Find functionality makes it easier to synchronize among various windows and panels. For the **Sort 1** transformation, you can jump to the code that is generated for this transformation or the log file and warnings that result from its execution.

## INTEGRATED DEBUGGER

As you build your job, you might want to run it to make sure that it produces the results that you expect and that it meets design requirements. This takes us to the world in which transformation nodes become execution steps that can be controlled to validate the process that is being designed. SAS Data Integration Studio makes this process easy. The integrated debugger supports a number of helpful features including:

- a status panel that indicates which job step is running and status of other steps
- the ability to run steps all at once or to run individual selected steps
- the ability to stop a running step at any time
- an animated visual indication of which step is currently running
- links to the log where warnings or errors occurred

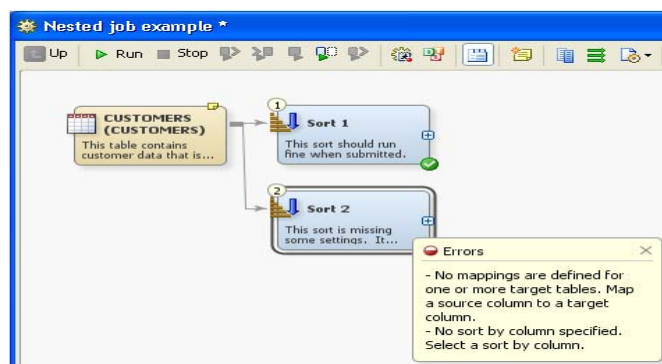


Display 9. Snapshot of a running job

Transformation step **Analyze** is currently running, while the **Rank variables** step has completed successfully. The **Filter results** step failed. While a step is running, progress is tracked by an animated, green indicator. The **Status** tab shows the progress of the job, and the **Log** tab contains the SAS log as the job is running. Once a step completes, its status changes to show whether the node ran successfully or with warnings or errors. You can view the **Status**, **Warnings and Errors**, and **Statistics** tabs to see additional information for each step in the diagram.

## EARLY DETECTION OF DESIGN ERRORS

All transformations are self checking. When a transformation is added as a node into a job, the transformation checks to see if the job has all of the information that it needs to run without errors. Missing information is indicated on the transformation as shown in the following display.

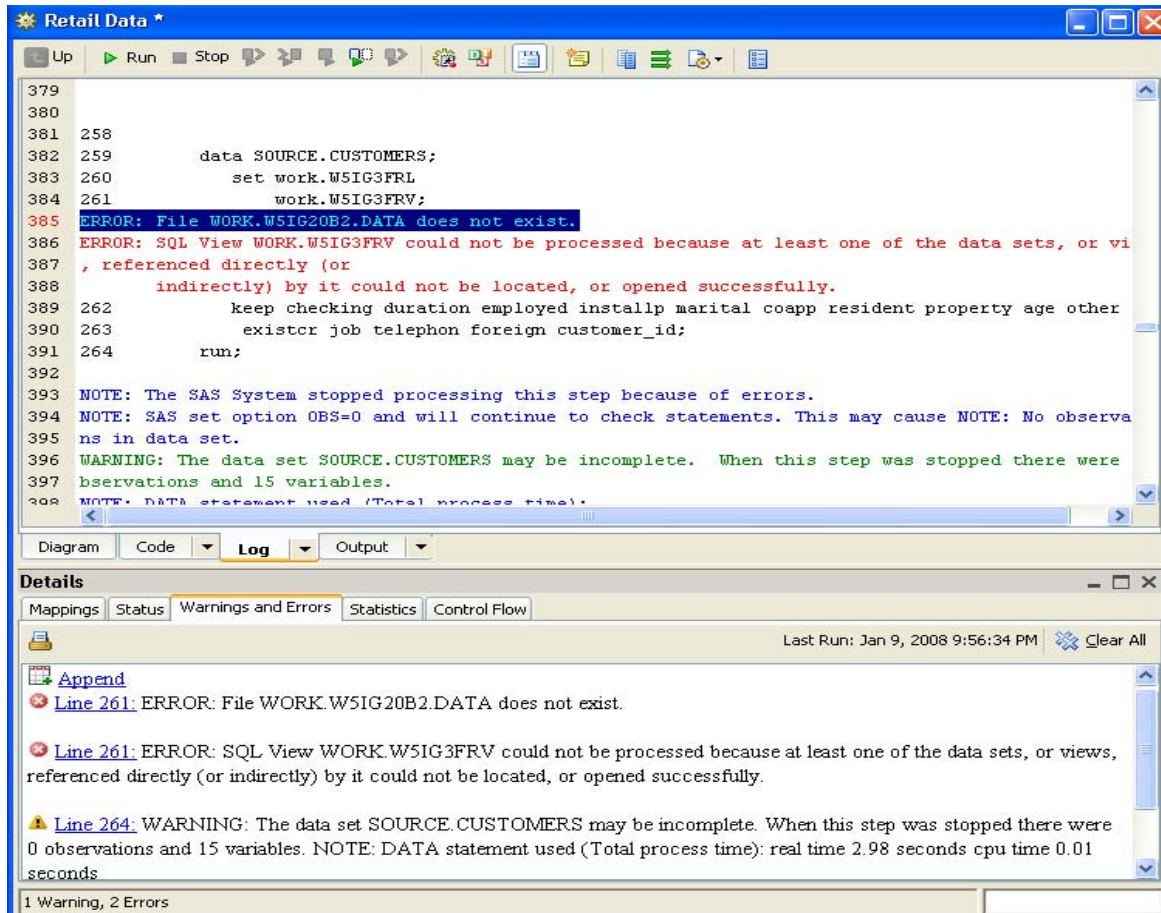


Display 10. Transforms display design errors on the node

Transformation **Sort 1** is complete; **Sort 2** is incomplete, as shown by indicators on the lower right of each node. In Display 10, we see that **Sort 2** is missing some settings. Details are provided about what information is missing and what to do to fix it. Transformations that are not complete are skipped when you submit them. This allows you to

run and test jobs even if they are not fully functional yet. You can also exclude a transformation from a run by choosing a menu option. Such transformations are dimmed in the diagram.

When it is time to run the job that you're designing, you might find that warnings or errors occur. This is common during any development process, and finding these indications quickly can make it easier to complete your work successfully.



### Display 11. Sample Execution Log of a Job

An error occurred while executing the **Append** step. This is highlighted in the general log (upper panel), and items of interest are enumerated in the lower panel.

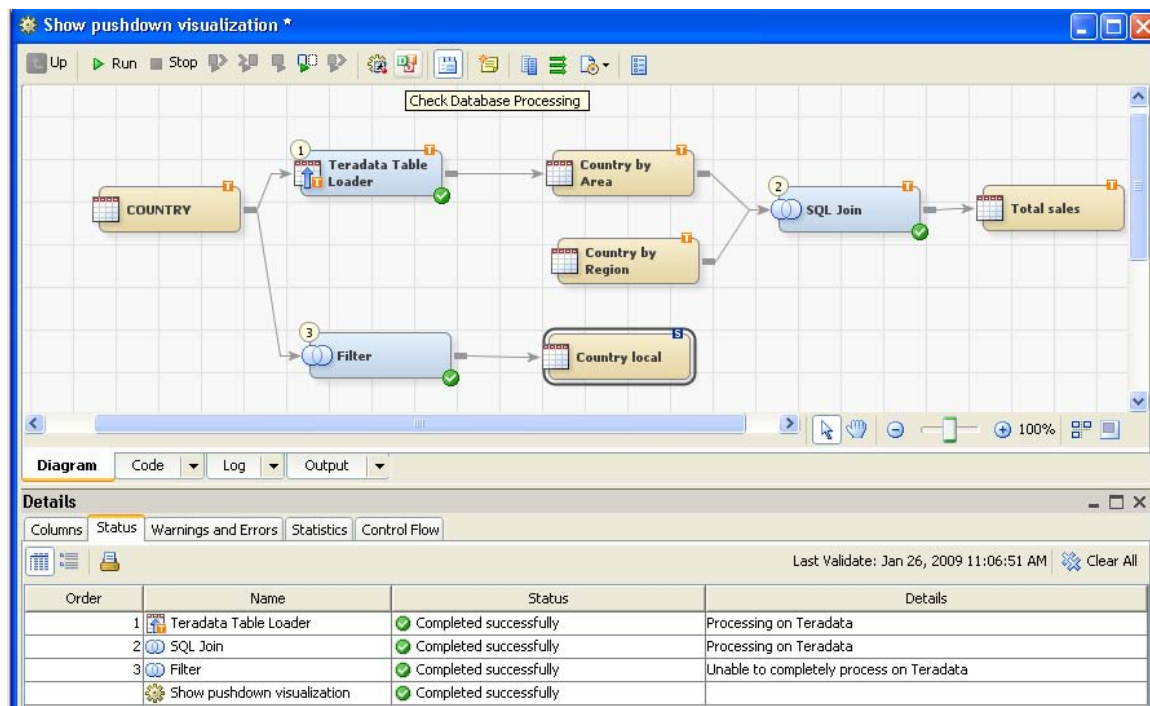
After the job executes, the lower panel contains a listing organized by transformation step of any warnings and errors that occurred in the job. For each warning or error, there is a clickable link to the location of that warning or error in the log. This simplifies the process of locating items in the original, complete execution log.

### CHECK DATABASE PROCESSING THROUGH TRANSFORMATION PUSHDOWN INDICATORS


You always want to ensure that your jobs are running as efficiently as possible. When joining data in a job that comes from a database, it is best to perform the join in the database. For example, suppose you are joining two tables that are coming from a Teradata system. If you can perform the join in the Teradata system and then bring back the result set, your job will run faster. This is because the result of the join is normally smaller than the source tables being joined and also because you don't have to move the data to perform the join.

SAS Data Integration Studio 4.21 provides a feature that will show you where your joins are taking place in your job. Nodes that can perform a join entirely on the database have an indicator and additional status showing where the join will be performed.





## Display 12. The Check Database Processing Feature

The nodes marked with the  indicator completely perform their processing in a Teradata database. The **Filter** node does not show an icon because it cannot push down to the database.

The transformation pushdown indicators are updated after the job runs to give you additional details about where your joins were performed. By looking at where your joins are going to run, you can make informed decisions about how to arrange the transforms in your job so that you can process as much data as possible on the database.

## SUBMIT INDIVIDUAL STEPS AND VIEW INTERMEDIATE TABLES

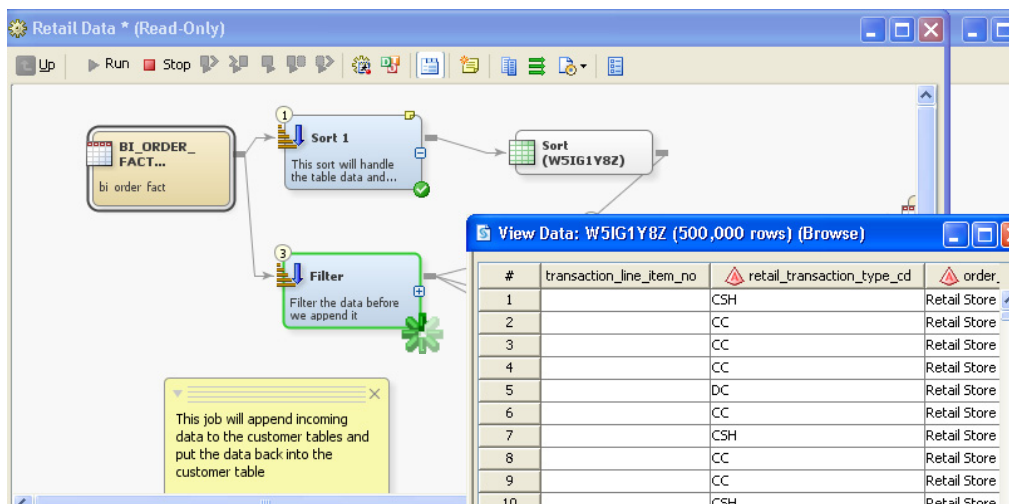
Sometimes you might want to run one or two steps in a job while you are building a job flow to make sure that you are getting the results that you expect. The run options enable you to run all or a portion of any job and stop at any time.



## Display 13. The debugging toolbar in SAS® Data Integration Studio

The run options in order from left to right are: **Run**; **Stop**; **Run from selected transform to the end**; **Run to selected transform from the beginning**; **Run only the selected transform**; **Step**; and **Continue**.

After any step completes, you can immediately see its status and view the results it produced, including any results that are in temporary tables. This makes it easier to check the outcome of any step that can be validated by inspecting its output. This can be performed even while the job is still running.

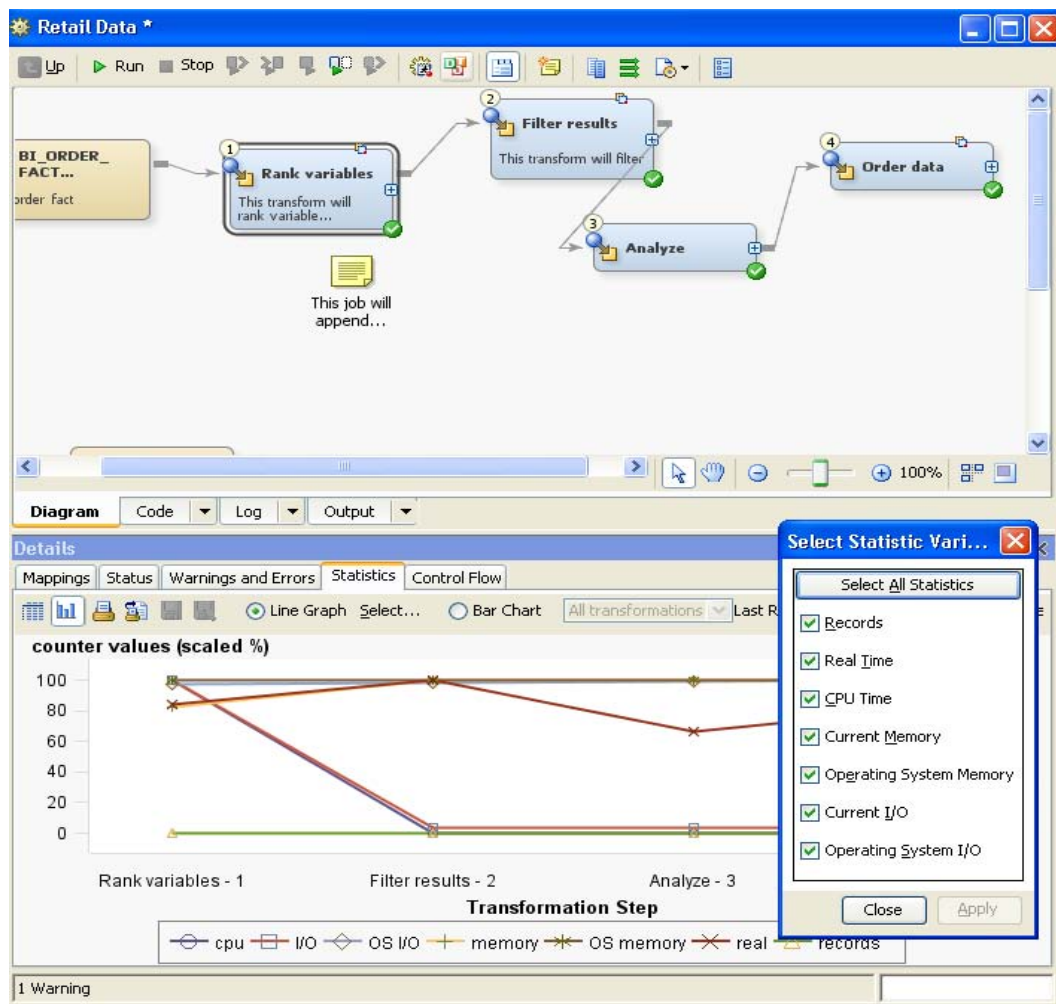


**Display 14. A running job with intermediate data displayed**

During the execution of **Filter** in the running job, output from a previous step is being reviewed.

### RUN-TIME PROGRESS INDICATORS AND STATUS

Often it is not enough to ensure that a job completes successfully. When you are working with large data flows, it is also important to ensure that the job performs well. SAS Data Integration Studio can capture run-time statistics on jobs to help you validate how well your code performs. This is based on Application Resource and Monitoring (ARM) capabilities in SAS 9.2.



**Display 15. A job showing the performance statistics of its last run**

During execution of the **Retail Data** job, statistics have been collected for each step.

Saved statistics are available for each transformation step and can be viewed graphically as shown in Display 15 or in a report. (See Display 16.) In addition, data can be saved to a file for further processing later. Various types of ARM-related information can be selected, including time elapsed, memory used, and I/O usage. These statistics depend on the SAS 9.2 application server that is running with ARM enabled.

Another helpful statistic shows the number of records (or rows) of data processed. This information can be used with any SAS 9.2 application server and is often a key diagnostic value for successful execution of a job step. This is very helpful if values are zero, which suggests that a complete step failure occurred. Another use of this statistic is to ensure that the same number of records is processed in each step, if the job is expected to maintain the same number across all steps.

Details												
Mappings Status Warnings and Errors Statistics Control Flow												
Line Graph Select... Bar Chart All transformations Last Run: Jan 9, 2008 10:30:52 PM Clear All												
Node	Name	Status	Records	Start Time	End Time	Duration	CPU Time	Current Memory	System Memory	Current I/O	System I/O	
0	Sort 1 - 1	Completed ...	500000	01/09/2008 at 10:29:...	01/09/2008 at 10:29:4...	30.671	8.656	5730.304	9670.656	1651273785	1652770853	cypre
1	Filter - 2	Completed ...	500000	01/09/2008 at 10:29:...	01/09/2008 at 10:30:1...	31.281	4.719	0	9670.656	800089136	0	cypre
2	Splitter - 3	Completed ...	500000	01/09/2008 at 10:30:...	01/09/2008 at 10:30:5...	36.797	4.062	0	9408.512	646966366	0	cypre
3	Retail Data	Completed ...		01/09/2008 at 10:29:...	01/09/2008 at 10:30:5...	101.843	17.438	0	9670.656	0	3099982329	SASJ

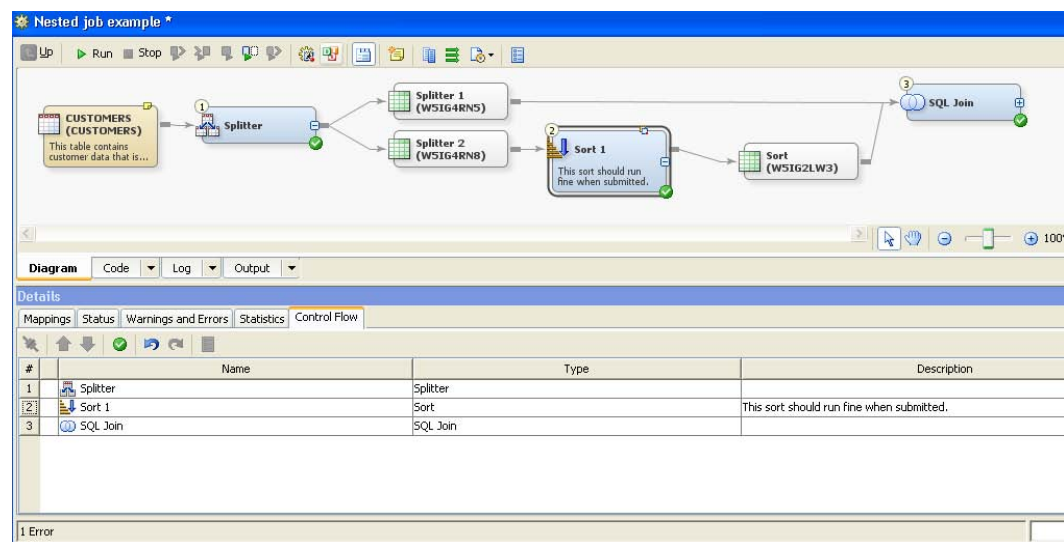
**Display 16. Run-time Statistics in Tabular Form**

This data can be used to drive further analysis or reporting.

## CONTROL OVER NODE-EXECUTION ORDER

Sometimes you need to change the order in which the nodes in a job should run. For example, you might want to run a particular node before others in the job are run because it does some data setup. In this case, you want to order the steps so that the job runs them in the correct sequence. Another reason is to control the order of execution of steps that follow a transformation like the data splitter. For example, if the data you are using is already sorted by a person's gender (F or M), and you are splitting the data on that variable, ensuring that the F gender data path executes first can turn a data consolidation into a simple append rather than a time-consuming sort.

SAS Data Integration Studio enables you to exert full control over run order. The order in which the steps will run is indicated by a number in the top left corner of the node. The **Control Flow** tab allows you to manipulate the order of the steps as shown in Display 17.



### Display 17 Using the Control Flow Tab to Manipulate the Order of Steps

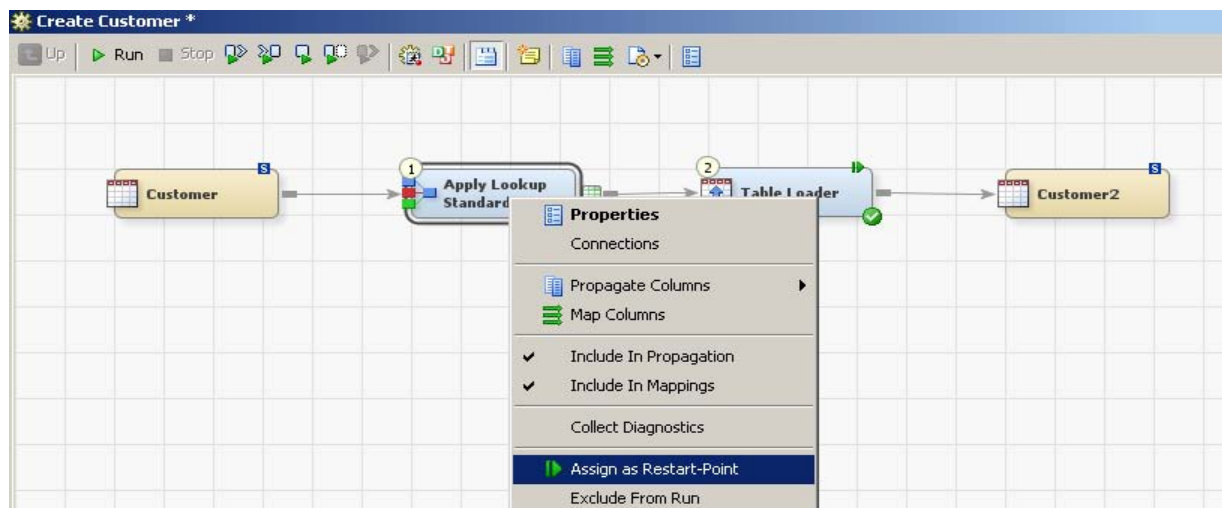
The order of nodes is in the standard form, as shown by values in the upper left of each transformation step.

The lower panel shows this order in a tabular form, where the order can be altered, if needed.

## SUPPORT FOR CHECKPOINT/RESTART

As data volumes continue to grow, the window of opportunity to update existing data is becoming smaller and smaller. In many situations the delivery of updated information is critical for day-to-day operations and decision making. Failure to deliver information because a problem was encountered during the processing of data is not acceptable. At the same time, starting the job process over from the beginning is not an option either because the additional processing time would exceed the window of opportunity. With SAS Data Integration Studio 4.21, there is now the ability to add checkpoints or restart points within a job flow. These restart points provide a means to store the state information of the job flow and the location of temporary tables that will be used to store the information that has been processed up to that point. Job flows that fail during execution can now capture all the necessary information and be restarted at the point of failure.

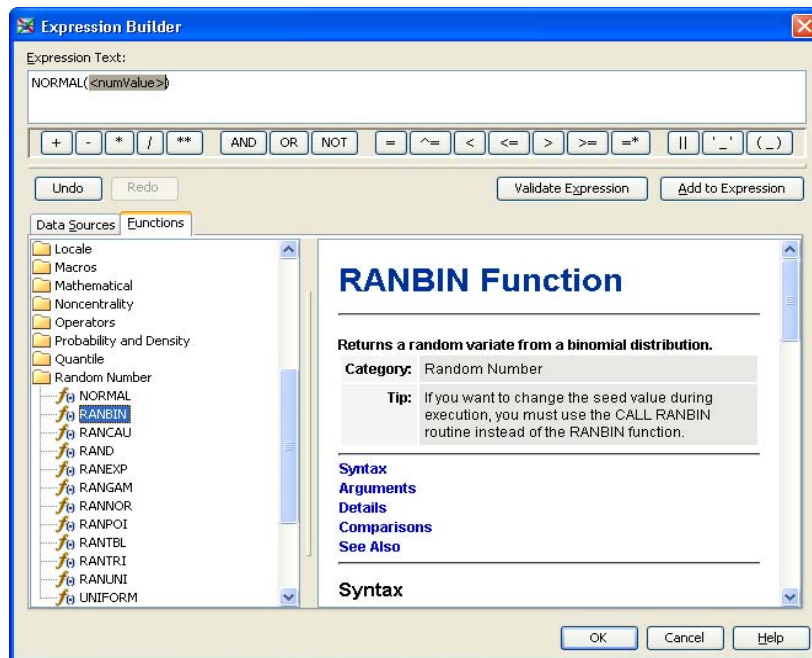
To access the Assign as Restart Point option, right-click on a given transformation node within a job flow or select the **Options** tab in the transformations properties.



Display 18. Setting a Restart-Point on Job Flow

### ENHANCED EXPRESSION BUILDER WITH USER-CUSTOMIZATION SUPPORT

The expression builder gives you the ability to build column-level expressions for manipulating your data. SAS supports hundreds of transformations, and many of them are displayed in a window where you can select them when building your expressions. This editor has been enhanced to support expression validation, to directly access SAS help for the many functions displayed, and display many more functions that became available with SAS 9.2.

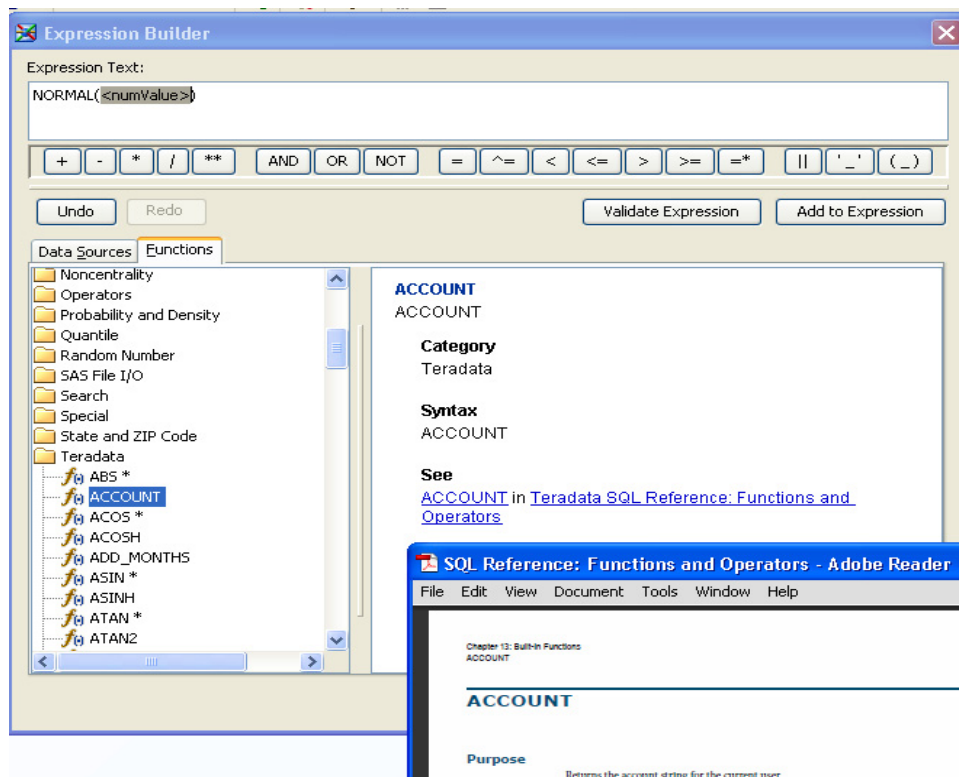


Display 19. Expression editor showing detailed help for the RANBIN function

The enhanced expression builder includes direct access to SAS Help for syntax, examples, and other detailed information. In this display, the RANBIN function has been selected and shows the help information that is available for that function on the right side of the panel.

Also, functions specific to Teradata are now selectable in the list of functions to choose from, with direct links to Teradata help using the Teradata function reference. A star (\*) indicator on the functions in the **Teradata** folder shows which functions will directly pass down to the database when used in SQL procedure statements.





**Display 20. The ACCOUNT Function Selected in the Teradata Folder**

The help information includes a link to the Teradata SQL reference manual. Click the link to see the help information that is available in the manual. You can also add information to this list by using a new interface that enables you to add your own groupings, functions, and optional Help to the window.

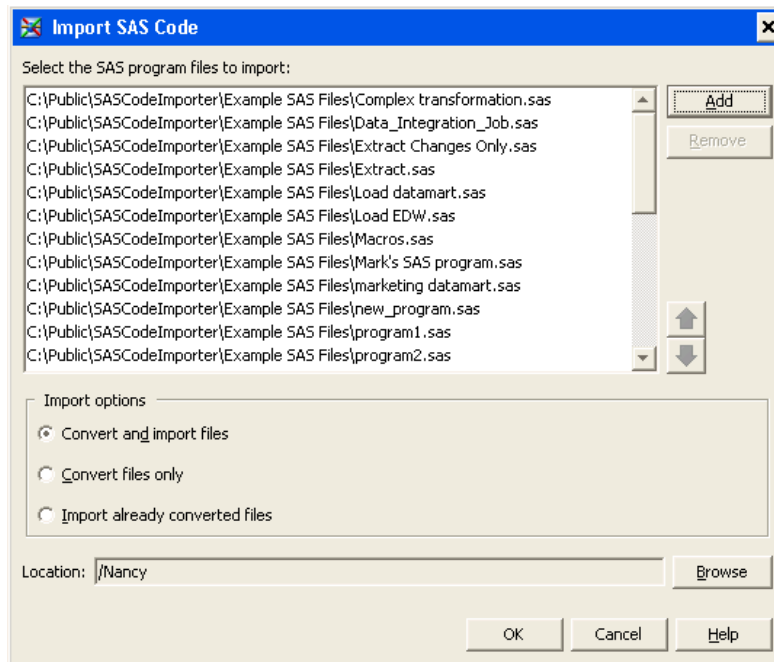
The expression editor enhancements, checkpoint restart, enhanced visualization, debugging capabilities, and the many other features described in this section are examples of the many enhancements available in the updated job editor. The job editor makes the task of building and testing high-performing SAS jobs quick and easy. Using the job editor, you can build, test, document, and support complex SAS jobs that get the best run-time performance. It also supports many useful workflow features to help make you more productive.

## EXPLOITATION OF THE SAS® 9.2 PLATFORM

New functionality in SAS 9.2 also brings a new world of capability for data integration. Enhancements in importing existing SAS programs into SAS Data Integration Studio, metadata management, security, as well as prompts and parameters for generated transformations are beneficial, as will be shown.

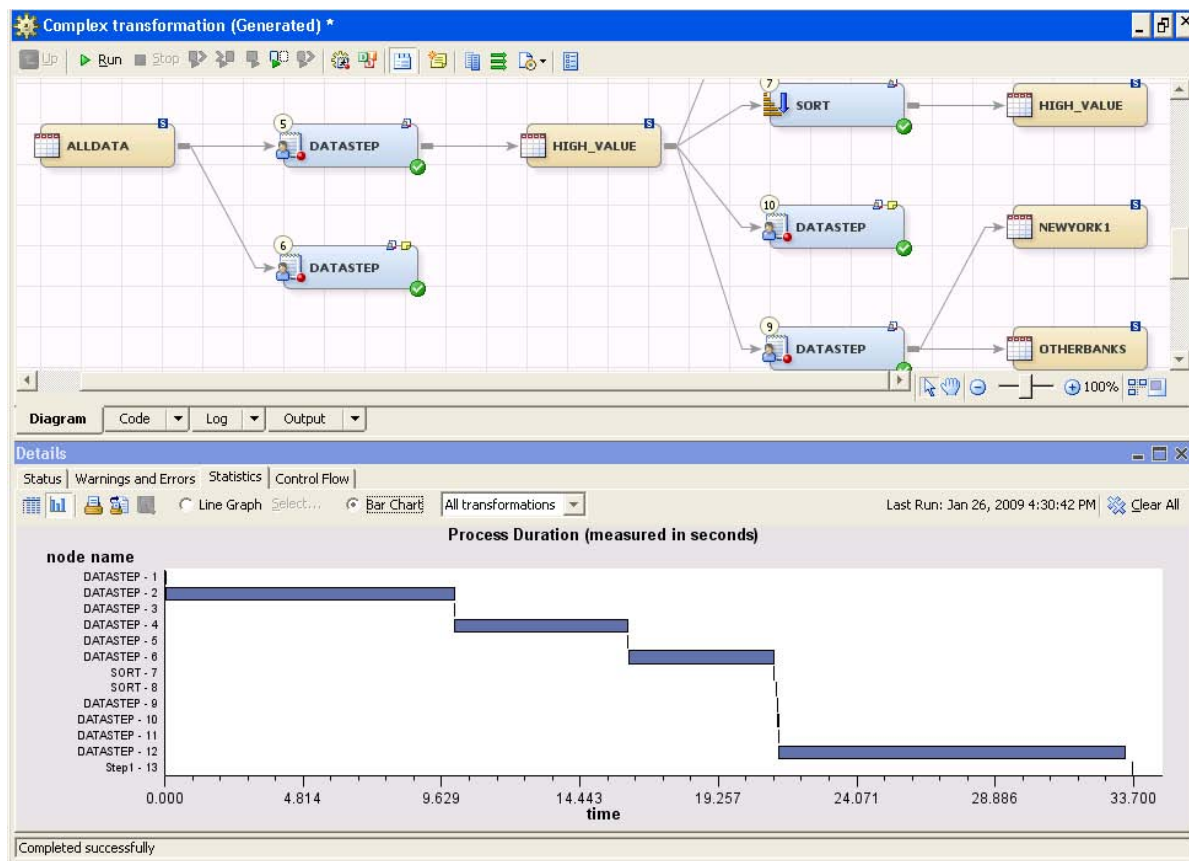
### SAS CODE IMPORTER: IMPORT EXISTING SAS® PROGRAMS INTO SAS® DATA INTEGRATION STUDIO

SAS Data Integration Studio now has the ability to import existing SAS programs and generate jobs that include the steps in the programs. Therefore, you can bring all sorts of existing programs that are currently in unmanaged code into a managed environment. For example, you can import a SAS program, then run that imported program in SAS Data Integration Studio, and view the performance characteristics of the program that shows how each step in the program is performing and where the bottlenecks are. Each step in the program is shown in a visual manner, and the code is enhanced with headers that describe the inputs and outputs of each step, to improve its readability. Any tables and libraries in the programs are registered for you, or connected to if the tables or libraries already exist in metadata. Mappings between steps are also created in the imported job flow.



**Display 21. The SAS Code Importer can import any existing SAS program into SAS® Data Integration Studio**

Simply point to the programs you want to import. The code importer uses a new SAS 9.2 procedure that parses existing SAS programs using the SAS Language processor and pulls out information about each step in the program. SAS Data Integration Studio reads that output, and creates jobs, tables, and libraries in the SAS® Metadata Server from the output. You can also add additional syntax in your SAS program that provides hints to the SAS Data Integration Studio importer that will add notes and naming conventions to the steps in your jobs. Other options available in the importer include the ability to rerun the import and view the output iteratively.



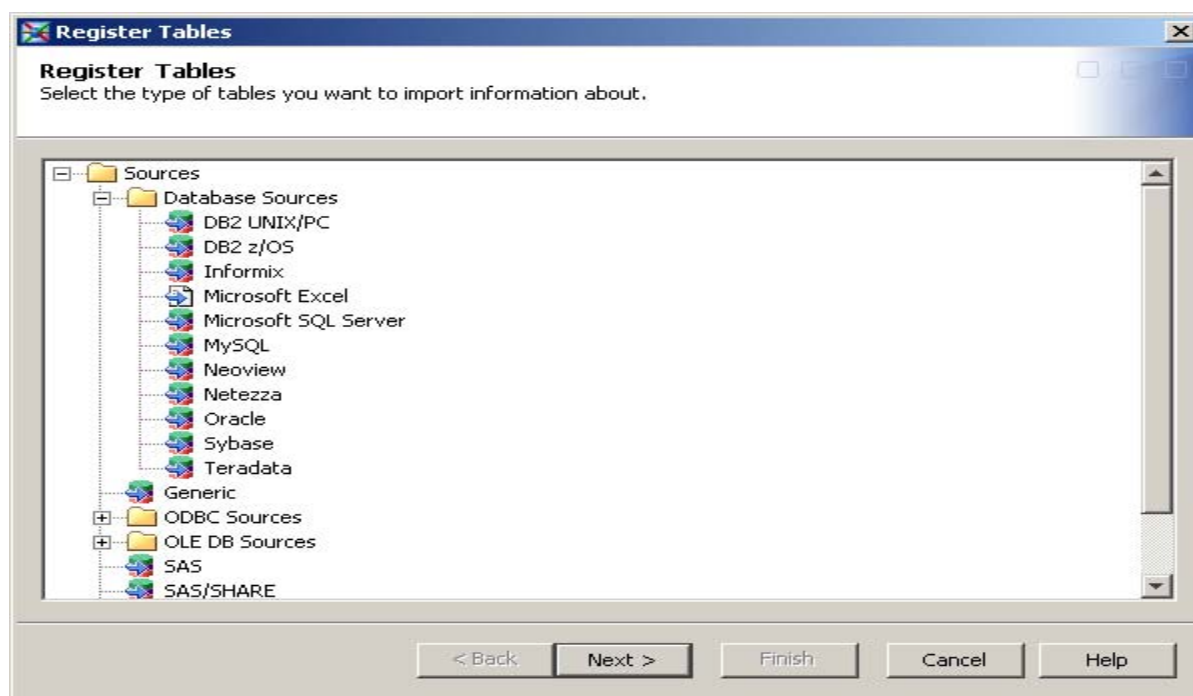
**Display 22. A Job That Was Imported Using the Code Importer**

Each step in the original program is represented in the job. Once the job is imported, you can use all of the SAS Data Integration Studio features, which include debugging and the ability to view performance metrics when the job is run.

## NEOVIEW AND NETEZZA SUPPORT

SAS Data Integration Studio surfaces source and target tables through the Register Table and Add New Table capabilities. In previous releases, the ability to define Netezza and HP Neoview sources and targets was available only through ODBC or the Generic LIBNAME Template. While this enabled you to both read and write data from or to these two data environments, it was not optimal and had limitations. Data Sources available through Register Tables and Add New Table are recognized as first class data objects. Basically this means that there is full support of that data source and its options and settings are supported through the LIBNAME template.

In the display below, both Netezza and Neoview are listed in the Database Sources folder in the Register Table wizard.

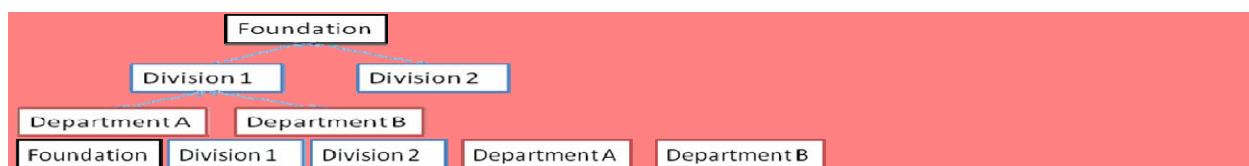


**Display 23. The Register Tables Wizard and the Addition of Netezza and HP Neoview to the List of Support Database Sources**

## METADATA

SAS 9.2 provides configurations of independent metadata repositories to store and organize metadata definitions of data, processing, and other elements of interest. Whereas a single metadata repository is a folder-based collection of these types of metadata objects, a collection of these objects is managed by a single metadata server to provide the right level of access to large bodies of metadata.

Users of SAS® 9.1 software will recognize the hierarchical configuration of metadata as useful for representing metadata along strictly hierarchical organizational units. This is beneficial when sharing, or inheriting, metadata definitions from higher-level organizational units. In this arrangement, global definitions are held at the highest level, and more specific objects exist where they're needed. This arrangement can be thought of as a dependent repository configuration because lower-level repositories depend on higher-level repositories for some types of needed information.

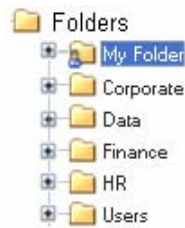


**Figure 1. SAS 9.1 Dependent Repositories (left) and SAS 9.2 Independent Repositories (right)**

In SAS 9.1, this dependent relationship also comprised the sum total of metadata that is available for use by someone who connects to a specific repository. In the preceding example, someone that is connecting to the repository, **Department A**, had visibility only to objects in **Department A**, **Division 1**, and **Foundation**. This is beneficial when cross-organization data sharing is allowed, but can be problematic in other cases. However, someone who connects to the **Foundation** repository sees a different view. In this case, that person only sees metadata present in the **Foundation** repository. Another effect of this configuration is that someone connecting to **Department** can't access any metadata in the repository for **Department B**.

In SAS 9.2, metadata repositories are all independent, and all metadata can be visible to all users. This means that people accessing metadata in **Department A**'s repository can also access metadata from **Department B**, if security settings allow them this access level. This is highly beneficial for report writers who need to access information from across an enterprise. Independence means that full visibility is possible across repositories.

Another effect of this configuration is increased metadata visibility in applications like SAS Data Integration Studio. In SAS 9.1 applications, a user could connect to a specific repository and be able to access metadata in that dependency tree, up to the **Foundation** repository. Although multiple, independent repositories can be used in SAS 9.2 configurations, that same user connects to the metadata server itself, and can then access any metadata repositories that security settings allow.



#### Display 24. Metadata Folders and Repositories in SAS® Data Integration Studio 4.21, based on SAS 9.2

Separate repositories for **Corporate**, **Finance**, and **Finance** are shown as separate folders.

In the preceding display, no **Foundation** repository is shown. The root of **Folders** is shown. In addition, separate divisions like **Finance** and **Finance** can be accessed as folders in a tree structure. This means that all metadata is accessible when needed. SAS 9.2 also introduces the general notion of home folders for each user's metadata. In this example, we see **My Folder**, but metadata belonging to other users can be found in the general **Users** folder. This is a big increase in access and can be managed through the application of security settings to grant or deny access and visibility to certain metadata areas within a repository.

This is also reflected in the view in SAS Data Integration Studio. While previous versions showed a Custom repository view, all SAS 9.2 applications show this common metadata folder view.

### Security

Security plays a role in providing the appropriate access level to metadata across repositories, but also has an effect when connecting to SAS metadata and workspace servers from SAS Data Integration Studio. This can be important in organizations in which the storage of passwords in metadata or in files on a personal computer is a concern, even when they are in an encrypted form..

In an environment in which Windows servers are used with Windows desktop PCs for client applications like SAS Data Integration Studio, Integrated Windows authentication can be used to avoid any password storage. If this type of authentication is used, no passwords need to be entered or stored. Instead, a trusted relationship between the server and desktop PC is used to convey credentials to SAS servers. This is first seen in SAS Data Integration Studio when creating a connection profile to specify the metadata server that is being used.



#### Display 25. Example of Integrated Windows Authentication Being Selected

With Integrated Windows authentication (also known as single sign-on), the user ID used to log on to the desktop PC is used to connect to a metadata server in addition to any workstation servers that are needed to view data, execute



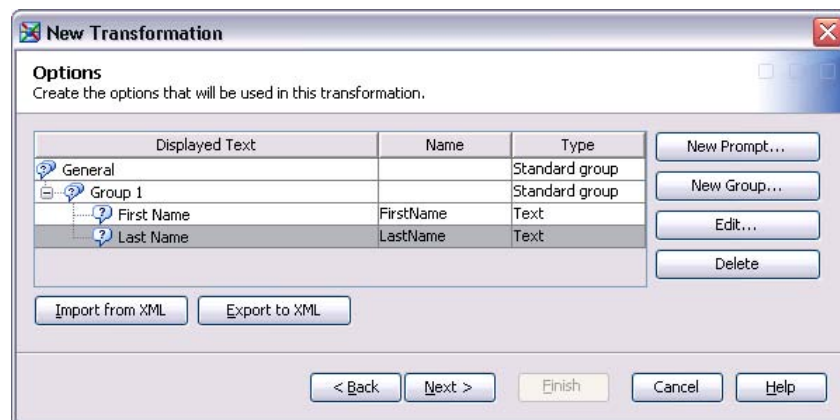
data integration flows, or perform other activities with SAS servers. This is recommended when one user's login ID is used on all systems, and in cases in which password maintenance is costly. Password maintenance can be an issue in enterprises that require passwords to be changed on a regular basis based on site-wide security policy. Because Windows looks up passwords at run time, the information is always current.

## SAS® 9.2 Dynamic Prompts

A major improvement occurring in SAS 9.2 is the ability to use dynamic prompts in reports, filters, and SAS Data Integration Studio transformations. When they are used in transformations, dynamic prompts enable option selections that provide a rich, dynamic user experience. The dynamic aspect can be based on previous selections, and can help provide the right level of information for subsequent questions that need to be answered in order to use the transformation.

### Dynamic Option Selection in Data Integration Transformations

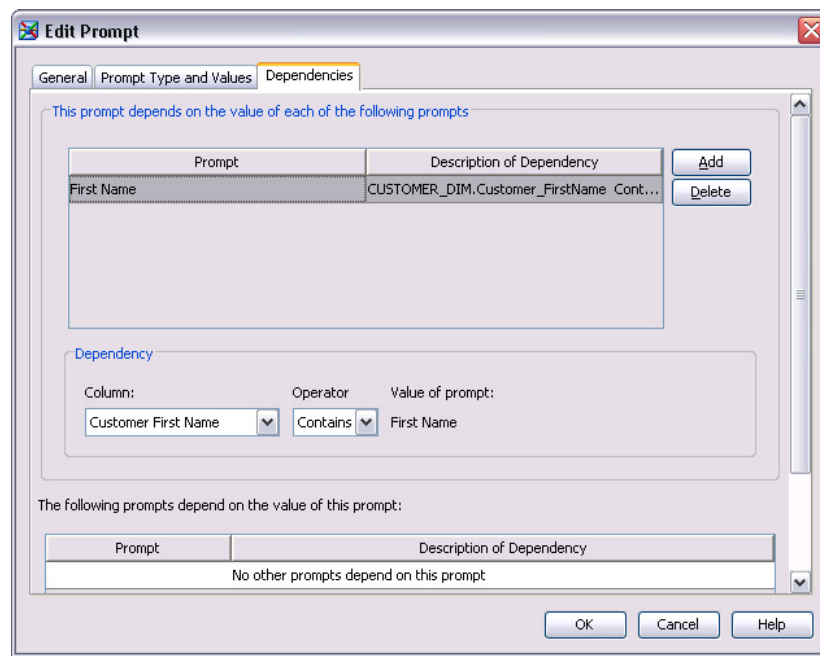
In the new world of cascading prompts, related fields need to be selected, such as matching first and last names. To support this statically, a selection from all last names is made, and then a listing of all possible first names follows. This method can be error-prone and time-consuming to use. To cascade dynamically, the initial choice of a first name causes the second list of last names to be set to only the matching last names. The benefit is clear to users of the transformations, who can more easily make their needed selections. To create a transformation like this, first define a set of prompts in a new transformation.



**Display 26. A SAS Data Integration Studio transformation design that uses two related prompts**

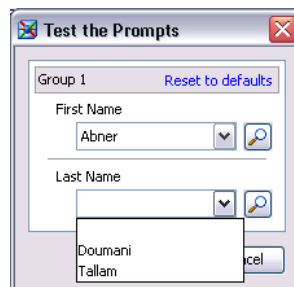
In this case, the transformation used requires you to select a first and a last name.

Next, assign a dependency between the two values that need to be entered. This enables dynamic linking between the first item that is entered (**First Name**) and values that can be selected for the subsequent item (**Last Name**).



**Display 27. Assigning a Dependency between First Name and Last Name in a Customer Table from Which Possible First and Last Name Data Pairs Are Chosen**

When you select these types of dependencies, a linkage is created that enables you to tie the options together. The result is that when testing or using the transformation, you are able to interact with the transformation based on dynamic data access. While designing the related option prompts, you can test the dynamically related prompts.

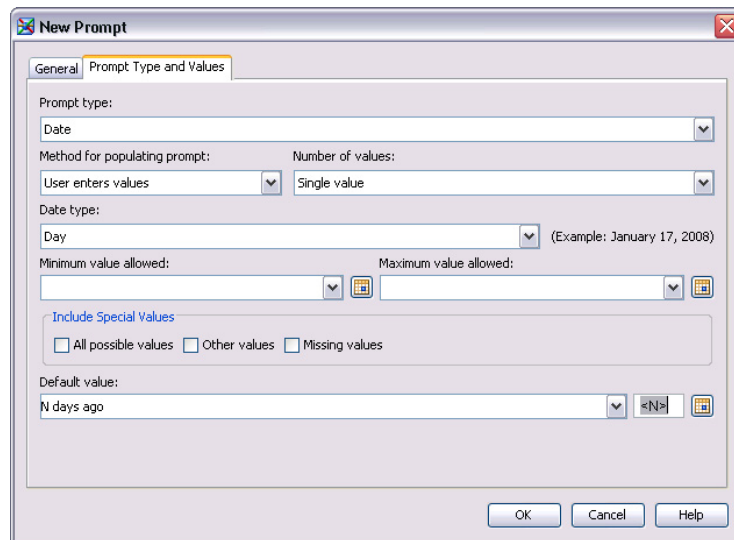


**Display 28. Last Names That Are Available to Match With the First Name, "Abner"**

In this case, there are two last names possible for people named "Abner."

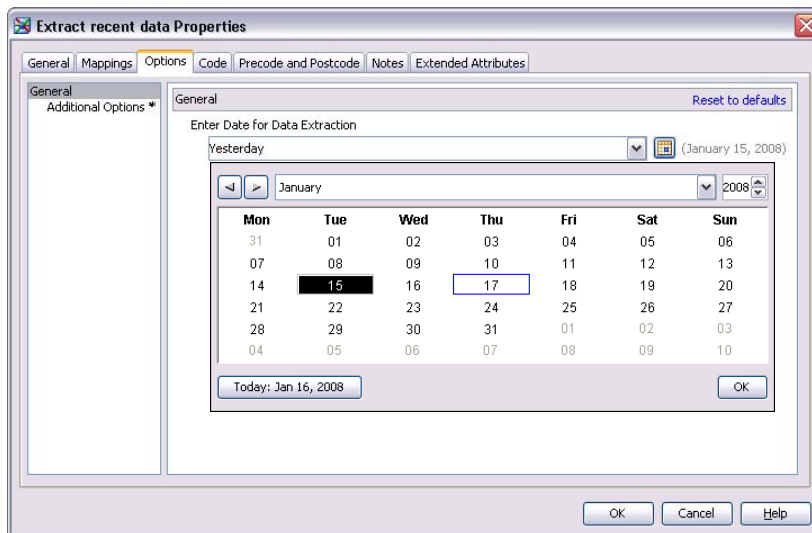
### More Interactive Option Selections

In addition to other enhancements, SAS Data Integration Studio 4.2 provides a richer user interface for options visualization. For example, when defining that a DATE value needs to be entered to use in a transformation, the designer has many more options for defaults, ranges, or other special handling required for that option.



**Display 29. Designing an Option Prompt for a Date Value With Specific Default Values and Other Parameters Set**

When using this transformation, you select a drop-down calendar to simplify your choice and ensure that the right format of data is entered.



**Display 30. Example of a Transformation That Requires a Date For Data Extraction**

When you are using this transformation, you can use a graphical calendar to select a date value for their selection.

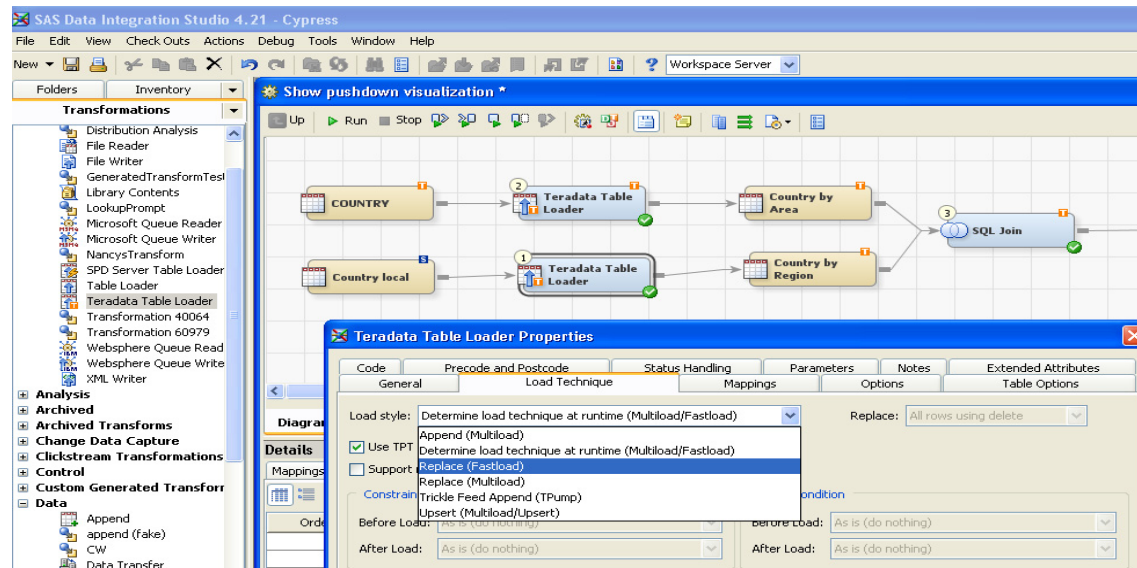
## TRANSFORMATION, CODE GENERATION, AND REPORTING ENHANCEMENTS AND ADDITIONS

Up to this point, you have seen how SAS Data Integration Studio 4.21 leverages new functionality in the SAS 9.2 Platform deliverables and previewed many of the major enhancements to the user interface, which include design capabilities, debugging, and run-time statistics. Another important component of SAS Data Integration Studio is the rich set of table-level and column-level transformations it provides, its integrated impact analysis, and its new reporting capabilities. In this section we will highlight some of these additional features including:

- optimized Teradata loader
- change data capture
- enhanced options handling
- data quality integration
- impact analysis
- metadata reporting

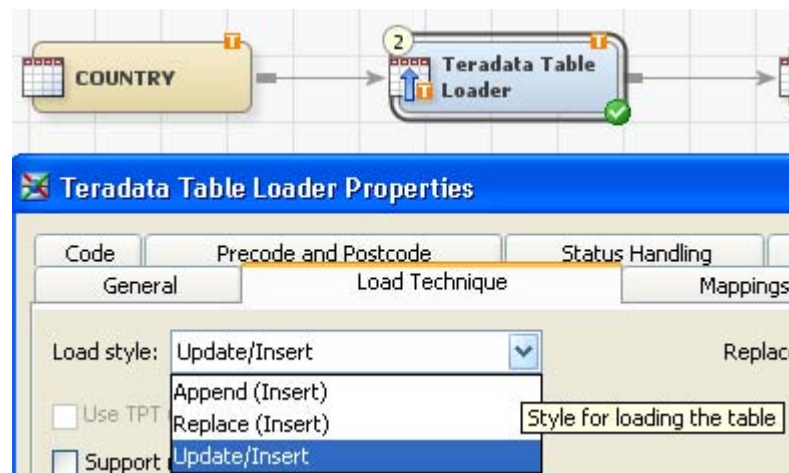
## OPTIMIZED TERADATA LOADER

In order to enhance performance when you are working with data, it is beneficial to take advantage of features of the database wherever possible. A new Teradata Loader transform has been added to enable you to optimize performance when loading data into a Teradata database. The new loader supports many features specific to Teradata, including Fastload, Multiload, TPUMP, and Upsert capabilities. The loader also supports restart ability when loading Teradata tables, using the restart features that are available in later releases of Teradata systems. Support for TPT Utilities is also included.



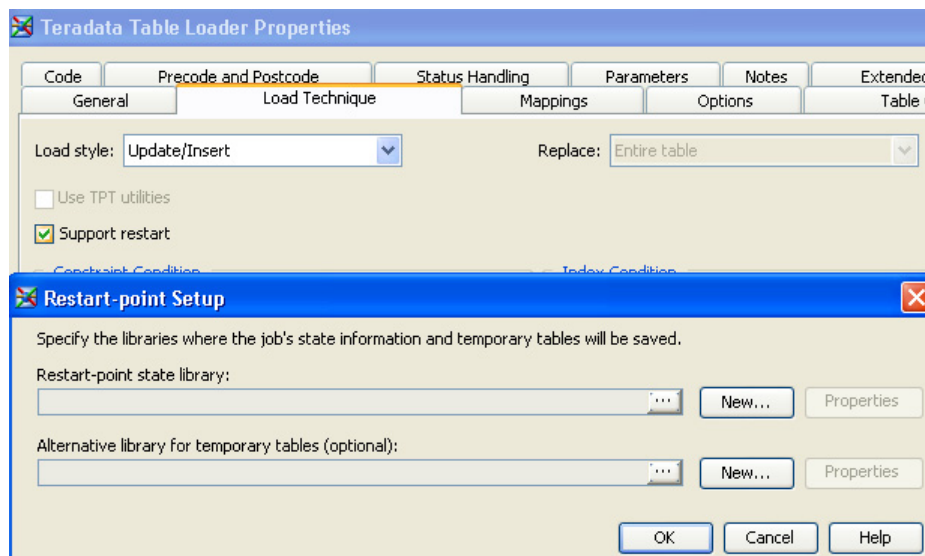
**Display 31. The Teradata Table Loader is Customized to Teradata Systems and Leverages Optimized Loading Capabilities for Teradata tables**

When moving data from one Teradata table to another, the loader is further optimized to ensure that the data stays in the database and does not incur any extra data transfers. The loader transform will also detect the number of records in the target table attached to it and make a best-guess selection, with tailored options, for the right set of options that can ensure the fastest load time.



**Display 32. Selecting the Best Load Techniques Available When Going from a Teradata Source to a Teradata Target**

Restart is also supported inside of the loader. When a load fails, you can restart the load from the last database commit point instead of having to restart the entire load of the table again. This can save a significant amount of time when you are loading large tables.



**Display 33. Restart Ability Leveraging Teradata Restart Capabilities is Supported in the Loader**

This can provide significant performance gains when working with large data.

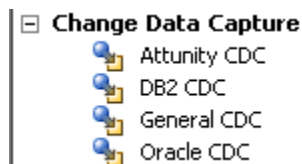
## CHANGE DATA CAPTURE

The basic premise behind change data capture (CDC) is to determine which values or records have changed in a particular source of information, capture the changes, and then deliver them to the appropriate environment. Change data capture can help reduce the volumes of data that are extracted to only the changed records, which greatly reduces the processing requirements as compared with requiring a full extraction whenever the data is needed. This reduction in data extraction can greatly speed up data integration activities. This results in delivering the required information in a timeframe that meets expectations or service-level agreements (SLAs).

Many database vendors deliver functionality in their products that captures these changes and stores them in either log files or a staging area. These staging areas provide details about the updated record, such as whether it was a new record, a change to an existing record, or a deletion of a current record. These systems also provide details about when the actual change took place. This becomes imperative when you are trying to synchronize the contents of the staging file with the content in the data warehouse in order to guarantee accuracy of the resulting data.

In SAS Data Integration Studio 4.21, additional functionality was added to support the CDC capabilities that are offered in Oracle, IBM DB2, and Attunity software. The current releases of Oracle and DB2 support both the log file and staging area concepts mentioned. Attunity provides integration with other sources of data, such as past and present versions of many RDBMS, in addition to file structures like VSAM, ADABAS, and others. The CDC capabilities offered in SAS Data Integration Studio focus on reading the change records and incorporating the results into a standard job flow.

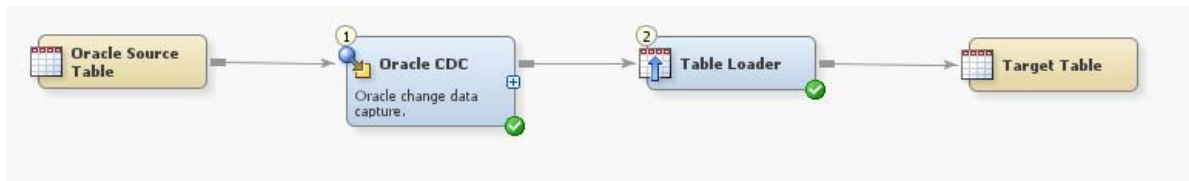
Display 34 shows the list of CDC Transformations that are part of SAS Data Integration Studio 4.21. These transformations can be used in any job flow for reading and processing the changed records from any of the supported sources.



**Display 34. Change Data Capture Transformations Available in SAS Data Integration Studio 4.21**

The following figure is an example of a simple job flow that reads data from an Oracle table, processes the changed records, and then loads them into a target table. A typical job might have additional transformations and can use an alternative load technique like a type-2 slowly changing dimension.

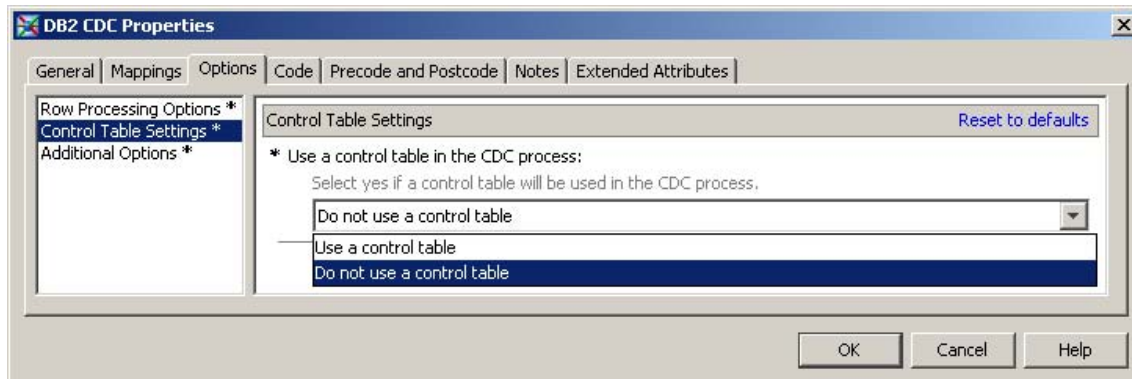




**Figure 2. Example of an Oracle CDC Job Flow Using the Oracle CDC Transformation**

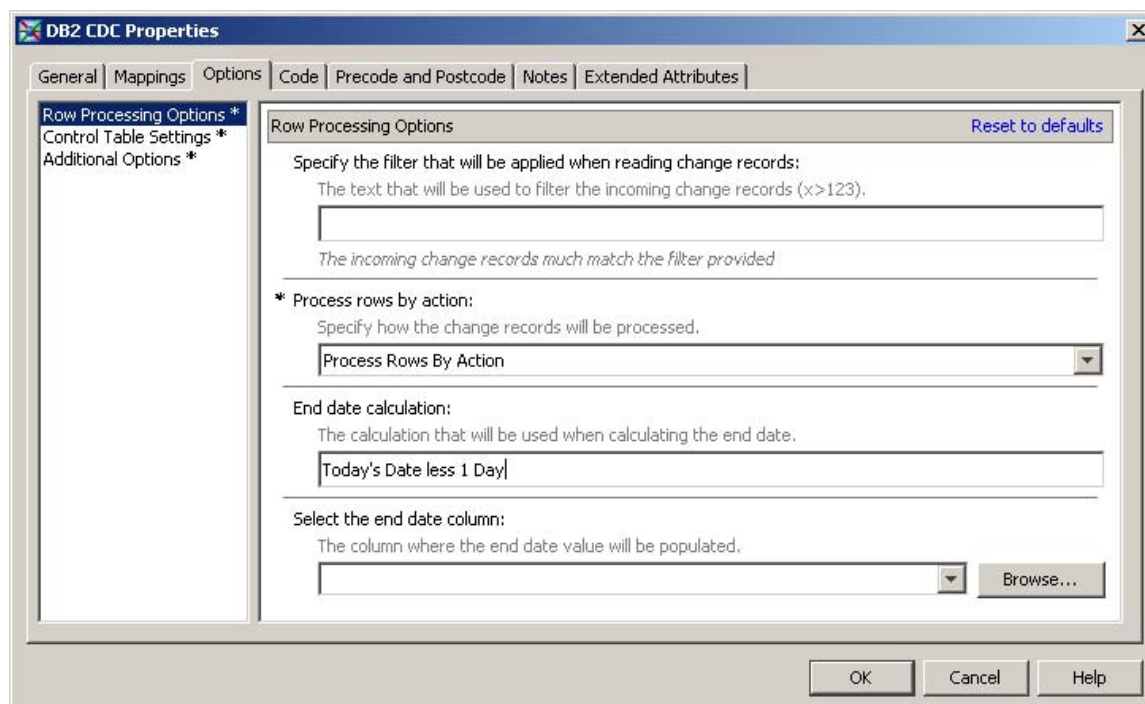
Here, newly changed records are extracted and placed in a target table for further processing.

The CDC transformations provide a set of row, column, and table-specific options that are supported by the various CDC sources. The transformations have been customized to match the data record structure produced by the various source suppliers.



**Display 35. Setting Control Table Options for a DB2 CDC Source Table**

Display 36 shows some of the row processing options that are available in the transformations. These options tell the transformation how to process the records in the staging file. For example, a filter can be applied to process the records only after a specific datetime value. Typically, this value is based on the previous job execution time, so only the records that have been added to the staging file since the last extraction will be processed. The row processing options are also used to indicate the preferred target load style. For example, if the design flow calls for historical data to be maintained (as in a type-2 slowly changing dimension), then selecting **Process Rows By Action** allows the developer to choose the desired end-data calculation to be used. This value is what will be stored as an indicator that the record has been logically deleted from the target table.



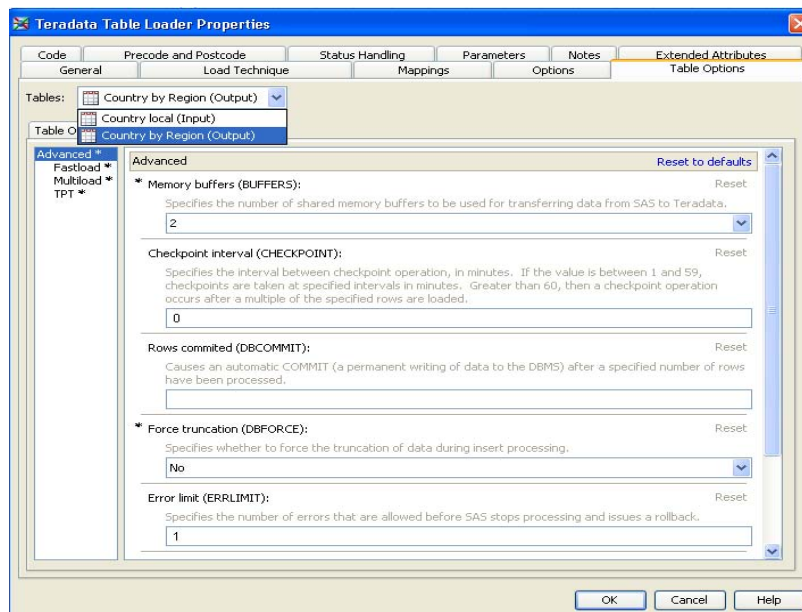
**Display 36. Row Processing Options**

### ENHANCED OPTIONS HANDLING

SAS Data Integration Studio enables you to customize how tables are read from or loaded into using table options. There are many options available in the SAS® language for tables. Some options apply to table reads only; others to table writes only. In some cases, if you apply a read option to a table that is being written an error is generated. Therefore, it can be useful to scope the options that are to be applied based on how the tables are being used. SAS Data Integration Studio has enhanced its options panels to help you:

- configure table options based on whether the table is being read from or written to
- configure options specific to each transform
- graphically see and configure database options based on table type
- get additional help for options

The following figure shows an example table options panel on a transform in a job. Each transform has one or more panels based on the number of tables coming in and out of the transform. Each panel is customized to the type of table being used, and whether the table is being read from or written to. Only the options that apply to how the table is being used are displayed to make it easy for you to select from only valid options. In addition, for database tables, an additional view is shown that displays the most common database-specific table options to allow you to further customize options based on table type. Options that apply to all uses of a type of table are also available on table properties.



**Display 37. Example table options panel on a Loader transform**

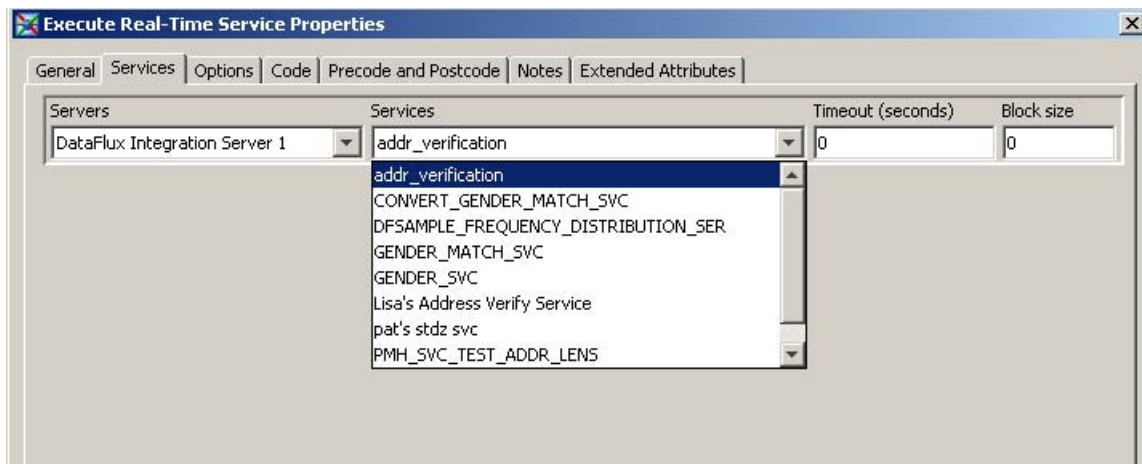
Table options are customized by type of table and how the table is being used on the new table options panels. These panels are available on all transforms and on tables.

## INTEGRATION WITH DATAFLUX JOBS AND SERVICES

Data quality continues to be important for users of SAS Data Integration Studio. Integration with software from DataFlux (a SAS subsidiary) is crucial to meet the continued demand, growth, and expectations on the quality of data that results from all data integration processes. SAS Data Integration Studio has traditionally included two table-level transformations (Apply Look-up Standardization and Create Match Code). The Apply Look-up Standardization transformation is designed to apply a standardization scheme created in a data quality client, DataFlux® dfPower® Studio. The Create Match Code transformation enables the creation of a unique identifier based on one or more columns using fuzzy logic algorithms that allow information to be related based on phonetics and associations. SAS Data Integration Studio also offers multiple column-level transformations or functions that accomplish tasks such as gender identification, parsing, standardizing values, and more.

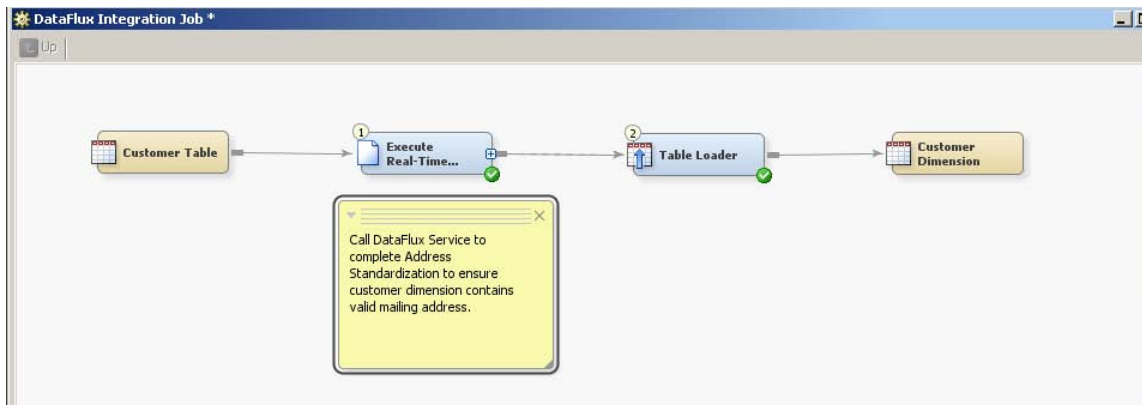
DataFlux has also introduced DataFlux Enterprise Integration Server, which supports the ability to execute both DataFlux® dfPower® Architect jobs and services that can accomplish any task within the DataFlux suite. Some examples of these services include address verification or address certification. In SAS Data Integration Studio 4.2, the ability to call and embed architect jobs and services like these within a job flow is now possible. The DataFlux Integration Server is registered with the SAS® Metadata Server, so it is easy to dynamically discover the jobs and services that are available to execute. SAS can detect the interfaces in the jobs and services defined in the DataFlux integration server and provide the right information to communicate with the server.

Display 38 shows the sequence to select the desired server and the associated services that are available for execution.



**Display 38. Select a Server to Display a List of Associated Services That are Available for Execution**

Once they are selected, these services add to the collection of transformational logic that is available in SAS Data Integration Studio jobs. When used in process flows, DataFlux Integration Server performs the required service on data specific to the job.

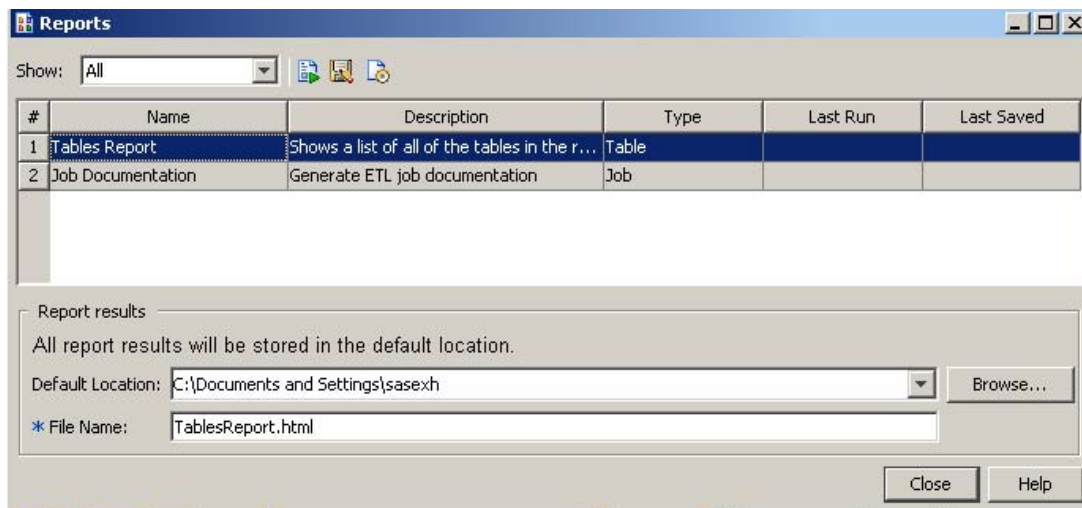


**Figure 3. Job Flow Executing a DataFlux Real-time Service for Address Verification**

## METADATA REPORTING

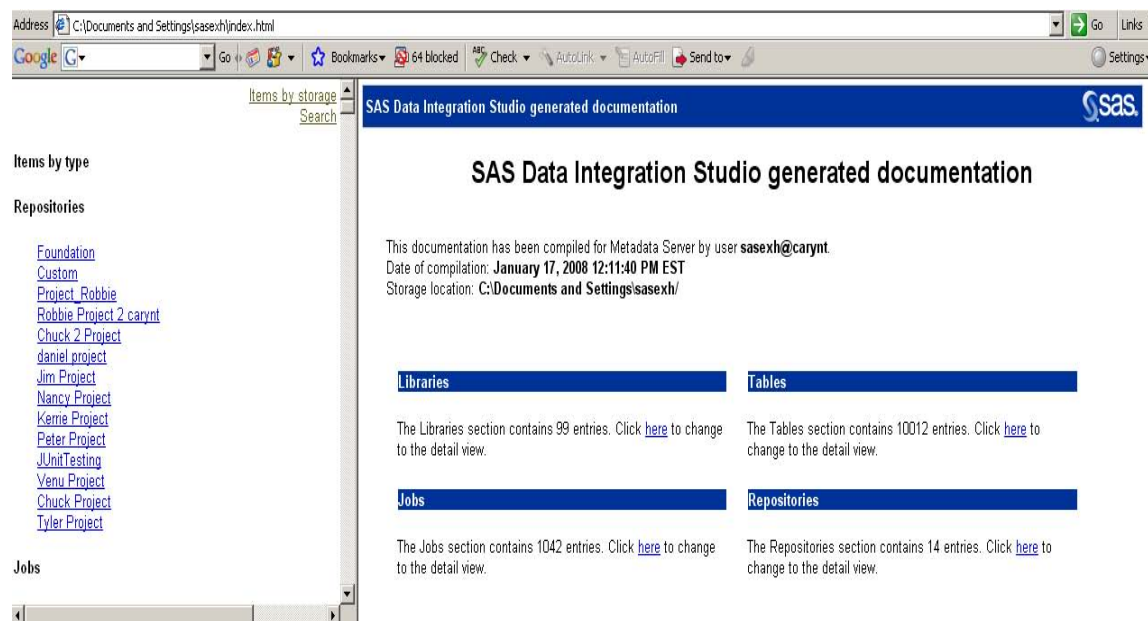
Metadata contains information that can be useful to the designers and developers of data integration processes. SAS Data Integration Studio 4.21 introduces the first phase of a multiphase delivery of metadata reporting.

Out of the box in SAS Data Integration Studio 4.21 are two different reports, Tables Report and Job Documentation. The tables report is a static report that provides a detailed list of all the tables that are defined in the metadata. This report can be executed, stored, and viewed from the Reports window, as shown in the following figure. Users can add their own reports to this reporting system if desired.



**Display 39. Available reports are displayed from the Reports Menu in SAS Data Integration Studio.**

In this case, the default standard reports are present. The **Job Documentation** report can also be executed, stored, and viewed from within the Reports window. The Job Documentation report constructs a dynamic navigation and reporting environment. This report produces a series of HTML documents that enable you to navigate and search the entire metadata contents. The report provides details about the repositories, libraries, tables, and jobs that are captured in metadata as part of a data integration design and development process. The following figures show some sample reports that are available from a Job Documentation report.



**Display 40. The Main Index**

A summary of libraries, tables, jobs, and repositories are available.



CODE LANGUAGE				
Location:	/dawong/migration dee			
Repository:	Foundation			
Created:	13Dec2007:17:10:17			
Updated:	13Dec2007:17:10:17			
Physical Name:	CODE_LANGUAGE			
Physical File:				
Type:	DATA			
Libname Statement:				
libname CIDD5 BASE "c:\SAS\Config\Levl\SASApp\Data\SASSolutionsServices\DDSDData";				
Jobs that write to this table:				
lName	Location			
100200 Load DDS CODE LANGUAGE Table	/dawong/migration dee			
100200 Load DDS CODE LANGUAGE Table	/dawong/migration dee			
Indexes:				
lName	Columns			
PRIM_KEY	LANGUAGE_CD, VALID_FROM_DTTM,			
Columns:				
lName	Label	Type	Length	Format
LANGUAGE_CD	Language Code	C	3	
VALID_FROM_DTTM	Valid From Datetime	N	8	DATETIME21.
VALID_TO_DTTM	Valid To Datetime	N	8	DATETIME21.
LANGUAGE_DESC	Language Description	C	255	
DEFAULT_LANGUAGE_FLG	Default Language Flag	C	1	
LOCALE_LANGUAGE_CD	Locale Language Code	C	2	
LOCALE_VARIANT_CD	Locale Variant Code	C	32	
LOCALE_COUNTRY_CD	Locale Country Code	C	2	
PROCESSED_DTTM	Datetime Processed by ETL	N	8	DATETIME21.
Responsible Parties:				
lName				Role

**Display 41. Details About a Selected Table, Including Information About the Columns in the Table, Jobs That Consume the Table, and Other Information**

## CONCLUSION

SAS Data Integration Studio delivers a wide range of new features and enhancements that make it easier for you to design, deploy, and monitor execution of complex data-integration flows. An interactive debugging environment provides many productivity improvements that make it easier to quickly move from initial design of new processes to their deployment and performance monitoring on an ongoing basis. Improved methods that enable you to quickly scan warnings and errors that can occur during execution help to focus attention on these problems. Smarter and faster mapping and propagation support in jobs means that you can focus on key content rather than details of transformation-level mappings, in most cases.

New capabilities are available to help advanced users manage order dependencies for steps within jobs, handle jobs within jobs, and manage intermediate and end results. Features such as searching, log processing, mapping rules, job nesting, and multi-table use capabilities within jobs provide ways to design and use very complicated processes that were previously difficult to represent.

SAS 9.2 features like dynamic prompting, which is available in user-generated transformations, provide many enhancements for designers of data-aware transformations, which can lead to smarter transformation use. Change Data Capture transformations support greater performance by leveraging vendor technologies for fast data extracts. Data-specific load options, such as upsert and MultiLoad for Teradata, open up a larger, new world of very large data processing capabilities in those environments.

In summary, there are many new reasons to use and benefit from SAS Data Integration Studio. From novice users to the most advanced data integration designers, a new world of capabilities is available to help make your data integration work easier, faster, and more productive.

## RECOMMENDED READING

Hunley, Eric, Gary Mehler, and Nancy Rausch. 2007. "Speed It Up – Active Warehousing with SAS® Data Integration: From Batch to Real-Time." Proceedings of the SAS Global Forum 2007 Conference. Cary, NC: SAS Institute Inc. Available at <http://www2.sas.com/proceedings/forum2007/100-2007.pdf>.

Rausch, Nancy A., and Nancy J. Wills. 2007. "Super Size It!!! Maximize the Performance of your ETL Processes." Proceedings of the SAS Global Forum 2007 Conference. Cary, NC: SAS Institute Inc. Available at <http://www2.sas.com/proceedings/forum2007/108-2007.pdf>.

SAS Institute Inc. 2007. "ETL Performance Tuning Tips." Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/resources/papers/ETLperformance07.pdf>.

SAS Institute Inc. 2008. "SAS Data Integration & Grid Benchmarking Results." Available at <http://support.sas.com/rnd/scalability/grid/benchmarking.html>.

SAS Institute Inc. 2007. SAS Institute white paper. "The New Data Integration Landscape." Available at <http://www.sas.com/apps/whitepaper/index.jsp?cid=3498>.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors:

Eric Hunley  
SAS Institute Inc.  
Cary, NC 27513  
Work Phone: (919) 677-8000  
Fax: (919) 677-4444  
E-mail: [Eric.Hunley@sas.com](mailto:Eric.Hunley@sas.com)  
Web: [support.sas.com](http://support.sas.com)

Nancy Rausch  
SAS Institute Inc.  
Cary, NC 27513  
Work Phone: (919) 677-8000  
Fax: (919) 677-4444  
E-mail: [Nancy.Rausch@sas.com](mailto:Nancy.Rausch@sas.com)  
Web: [support.sas.com](http://support.sas.com)

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

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