Paper 125-2008

A Whole New World: What's New in SAS® Data Integration Studio 4.2

Eric Hunley, SAS Institute Inc., Cary, NC
Gary Mehler, SAS Institute Inc., Cary, NC
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 data-oriented processes more efficiently and with greater control and flexibility. A major focus of this release is to deliver new visualization features that incorporate support for debugging. Improvements have been made to nearly every transformation. Performance analysis and optimization capabilities have been enhanced through the use of integration with system monitoring facilities. Increased integration across the platform is another focus area that provides a better integrated user experience across the SAS suite of products. 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 for the construction, execution, and maintenance of 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 4.2 is a major upgrade to a product that is based on over thirty years of SAS experience managing data for the purposes of solving business problems. SAS Data Integration Studio is the successor to SAS ETL Studio and SAS/Warehouse Administrator. Before SAS/Warehouse Administrator many SAS users or consultants solved data integration challenges with Base SAS programming through home-grown 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.

The history illustrates that the SAS data management and data integration capabilities grew up from a do-it-yourself, nuts and bolts type approach. Today, organizations are expecting tools and solutions that anticipate their needs and provide capabilities or services that provide a custom set of tools to complete the same tasks as the past, but with more elegance, ease, and precision. In today's world, we are expected to deal with challenges that might not have existed 15, 10, or even 5 years ago. These challenges are being magnified from year to year. 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.

This new world of challenges places much higher expectations on the tools and solutions that are designed to get the right data in the hands of the right people at the right time. This is true for SAS Data Integration Studio as well. Existing developers and new users require products to allow them to work as efficiently and effectively as possible. They also expect the product to be easy to use and to increase their productivity from design through development, debugging, and maintenance of jobs to complete a task.

This paper highlights the new features and capabilities in SAS Data Integration Studio 4.2 to meet today's challenges. It discusses how it can help you meet the expectations and requirements 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 available in SAS Data integration Studio 4.2 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, runtime, 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
- runtime progress indicators and status
- detailed performance, warning, and error information
- control over node execution order

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.

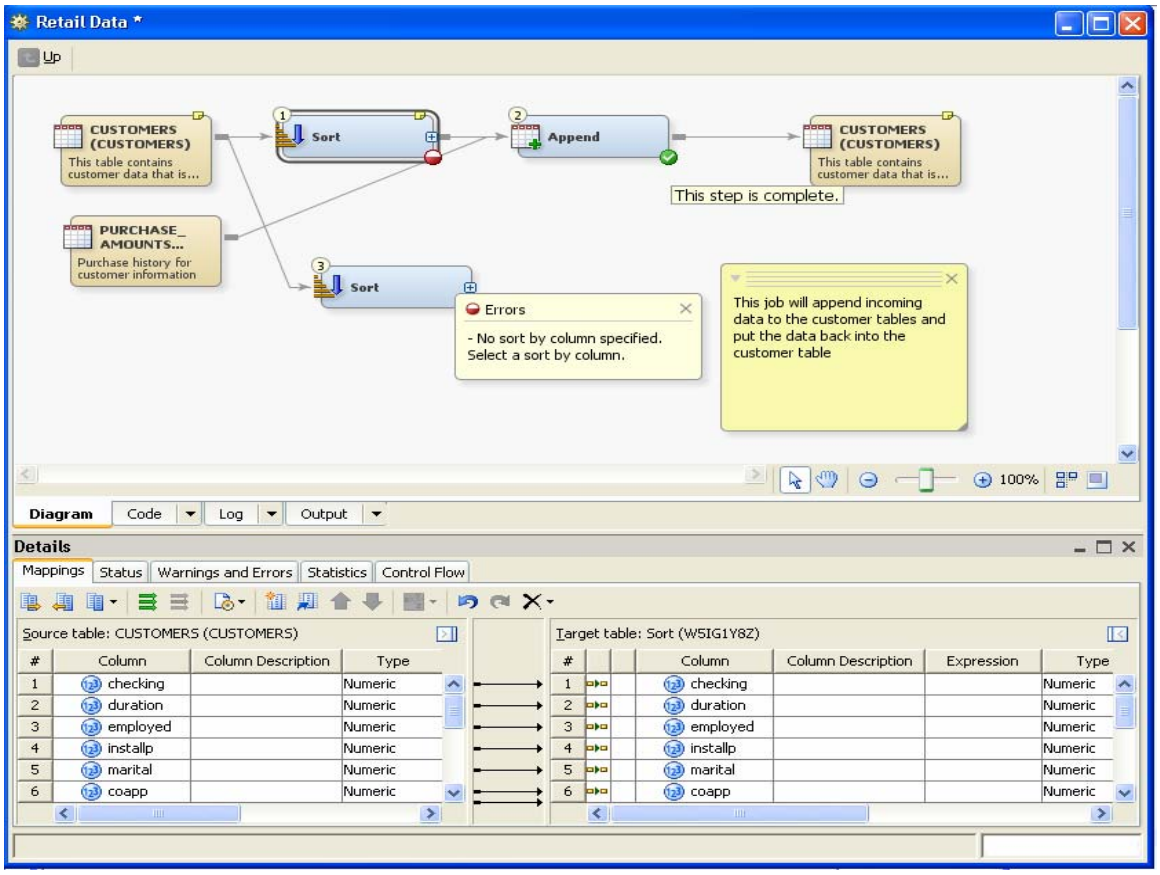


Figure 1.1: This figure shows an example of an open job in the job editor.

The job editor has a number of tabs and panels that help you build and visualize your data integration flow. The diagram panel includes annotated data and transformation nodes. In addition, further annotated comments can be placed in sticky notes in the diagram. The status of each node is displayed as well. In the preceding example, we see 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 4.2 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. 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 Figure 1.2.

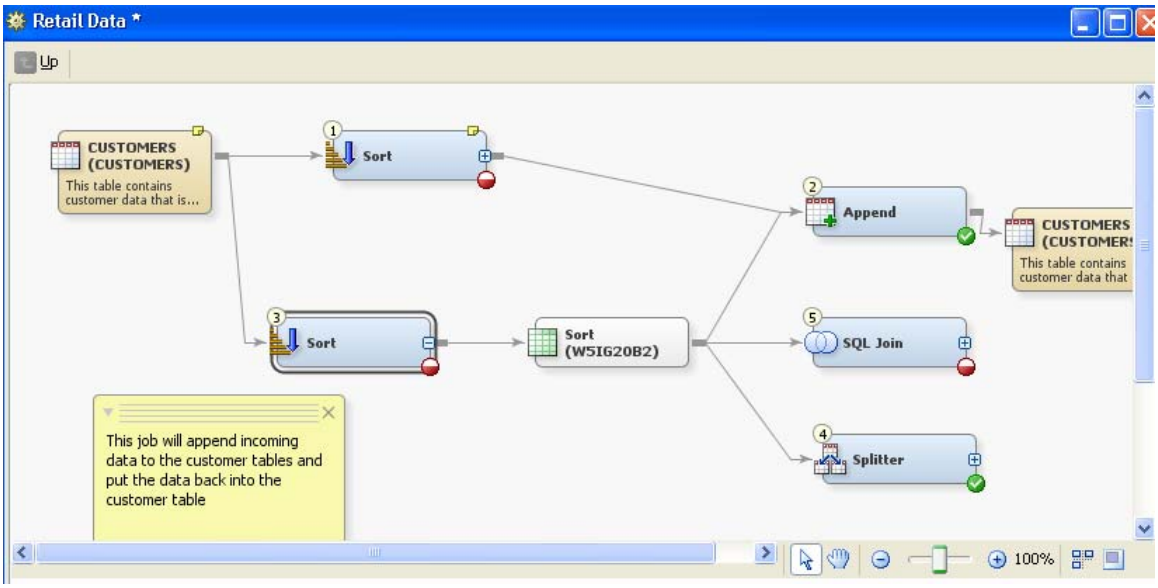


Figure 1.2: In this diagram, the CUSTOMERS table is both source and target of the flow. In addition, output from a Sort transformation feeds into three subsequent steps.

The 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 Figure 1.3, 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.

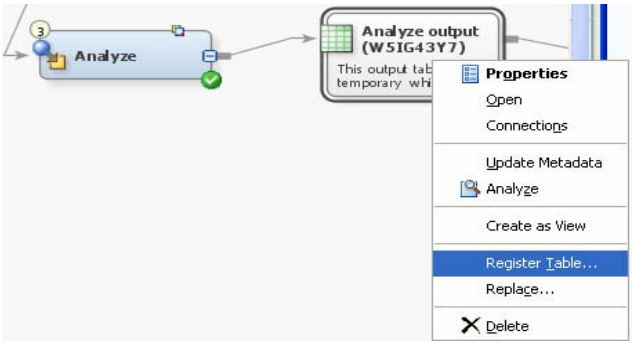


Figure 1.3: The Register Table technique is just one of several techniques available to easily modify table structures as you build your jobs.

Also new in SAS Data Integration Studio 4.2, jobs can be embedded or nested more easily inside of other jobs. For example, you can build a job flow, 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 Figure 1.4.

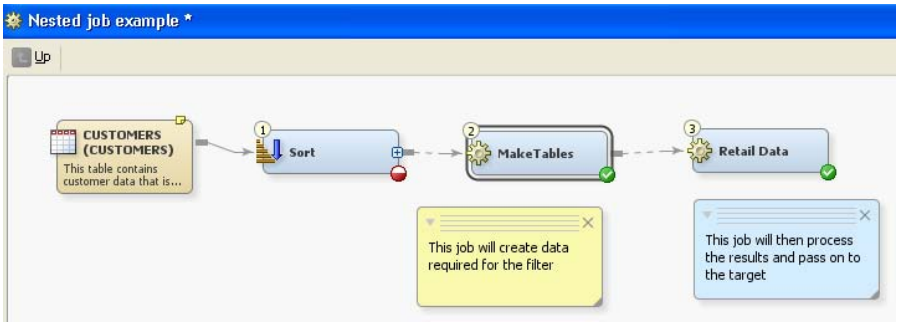


Figure 1.4: The diagram nodes **MakeTables** and **Retail Data** are data integration jobs that are part of the larger job called **Nested Job**. Sticky notes in the diagram document the overall flow.

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.

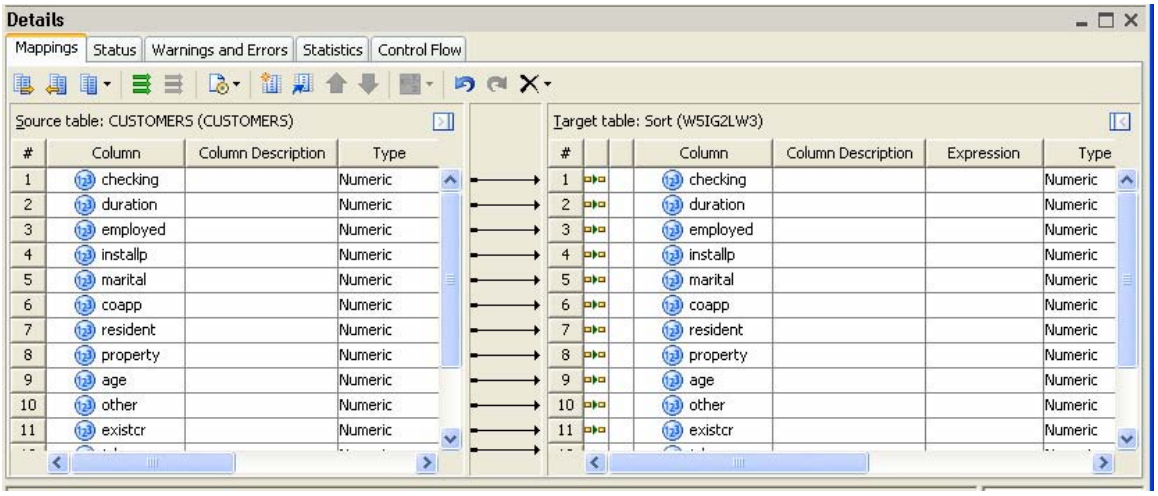


Figure 1.5: The Details panel shows information such as source to target mappings. This lower panel can be viewed at the same time as the upper Diagram panel, simplifying complex mapping design.

Mappings

In any transformation, the Mappings panel allows you to define the propagation of information from table to table in your job. Propagation mappings can be created that originate before or after the transformation, or 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.

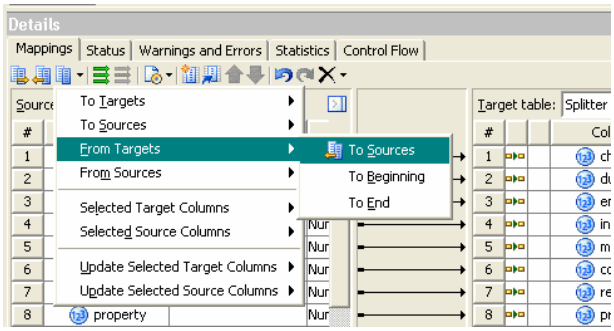


Figure 1.6: Mappings and propagations are specified in a mapping panel.

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.

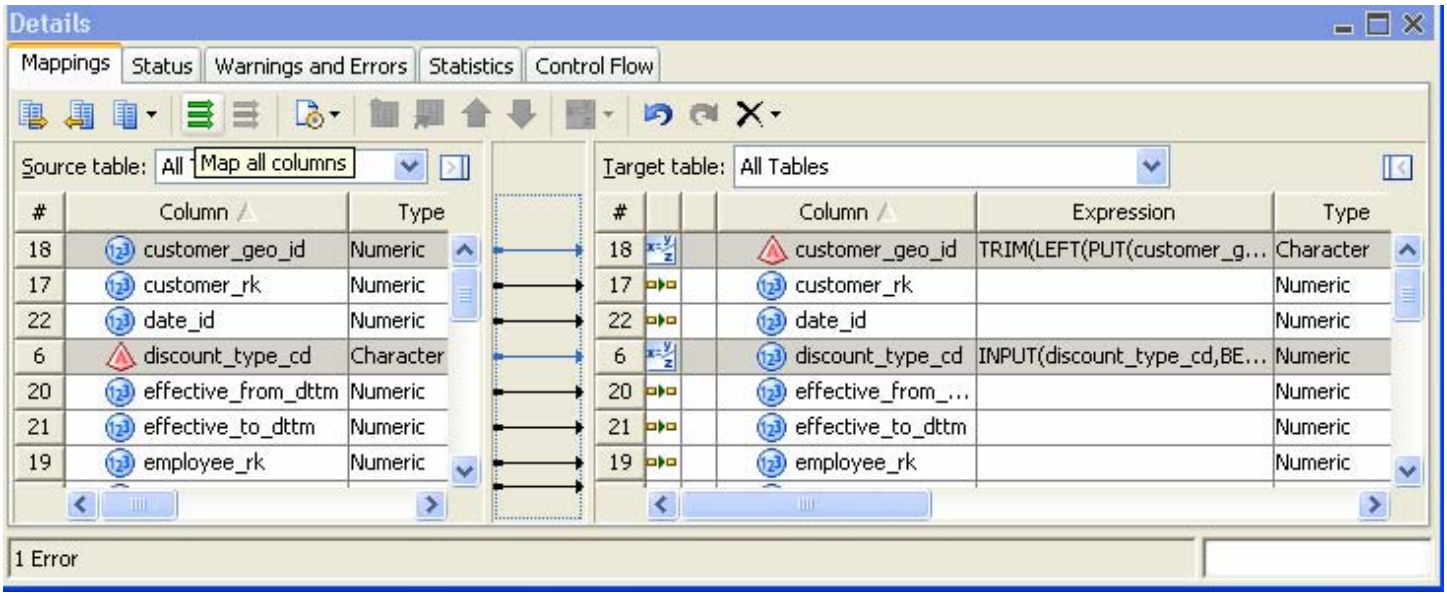


Figure 1.7: This figure shows an example of the default mapping rules being applied when mapping numeric to character columns, and character to numeric columns. Default mapping rules are pre-defined, and can be extended to meet business-specific needs.

A number of search capabilities have been added to SAS Data Integration Studio to make it easier to locate items in the application. Figure 1.8 illustrates some of these search options available. 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 runtime behavior of the selected transformation.

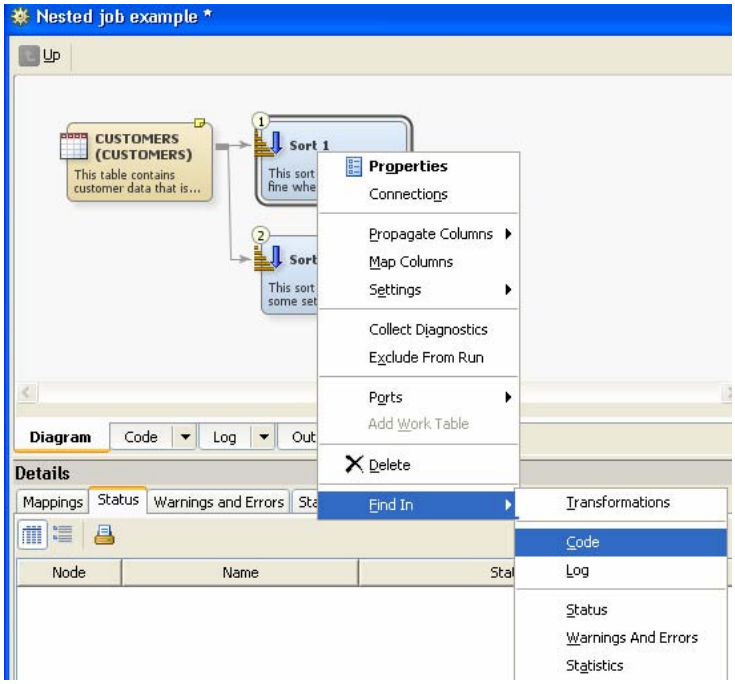


Figure 1.8: Find functionality makes it easier to synchronize among various windows and panels. For the **Sort 1** transformation, you can jump to the code that is generated for this transformation or the log file and warnings that result from its execution.

Integrated Debugger

As you build your job you will probably 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 being designed. SAS Data Integration Studio 4.2 brings a new world of integrated debugging to make 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

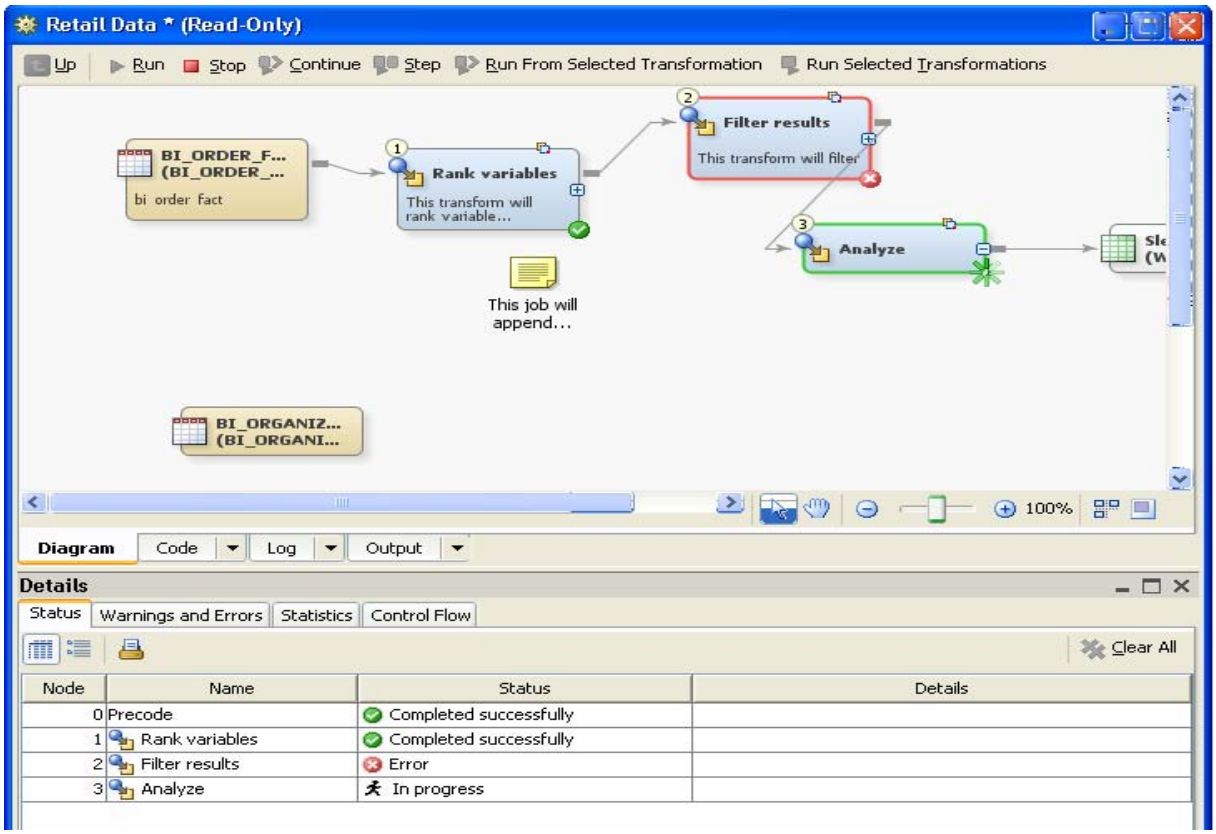


Figure 1.9: This figure shows a snapshot of a running job. Transformation step **Analyze** is currently running, while the **Rank variables** step has completed successfully. The **Filter results** step failed.

While a step is running, a green animated progress indicator is displayed. 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 figure.

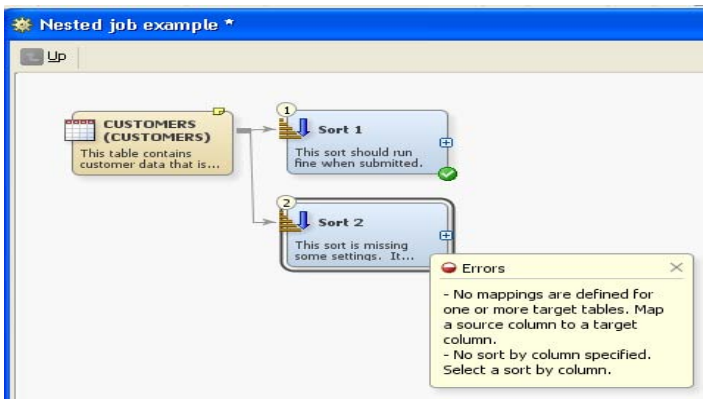


Figure 1.10: Transformation **Sort 1** is complete; **Sort 2** is incomplete, as shown by indicators on the lower right of each node.

In Figure 1.10, 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.

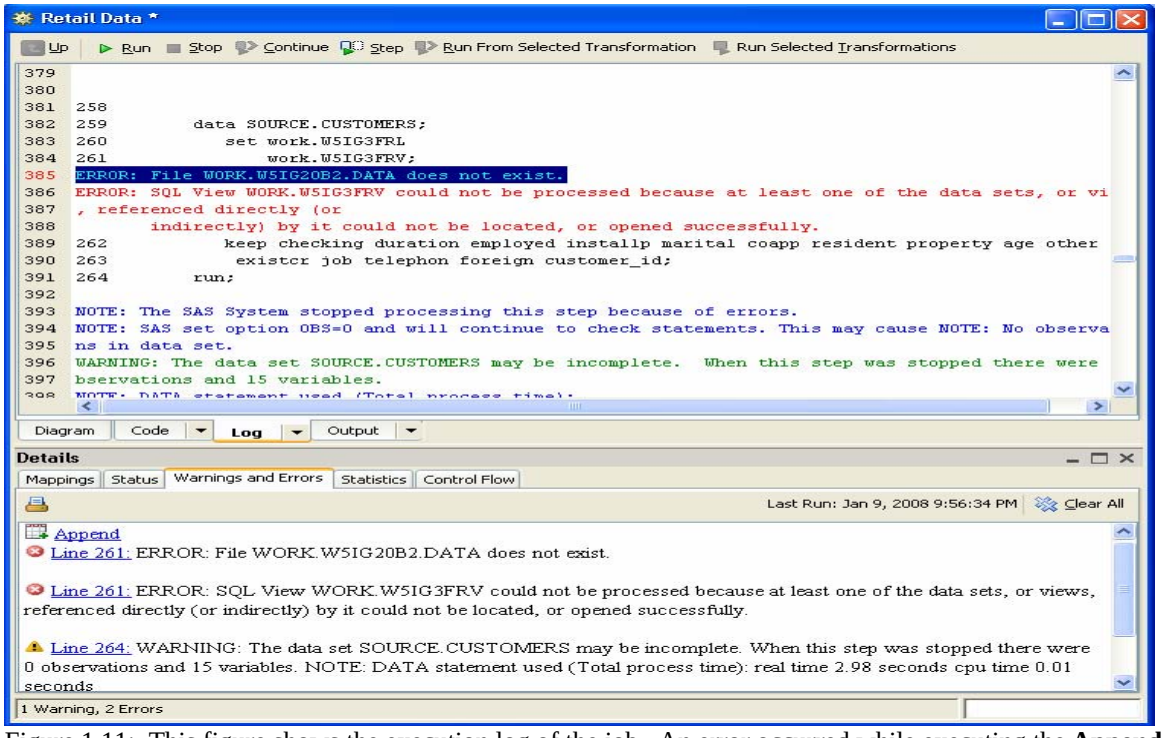


Figure 1.11: This figure shows the execution log of the job. An error occurred while executing the **Append** step. This is highlighted in the general log (upper panel), and items of interest are enumerated in the lower panel.

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.

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 allow you to run all or a portion of any job and stop at any time.

Figure 1.12: This figure shows the debugging toolbar in SAS Data Integration Studio.

Once 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.

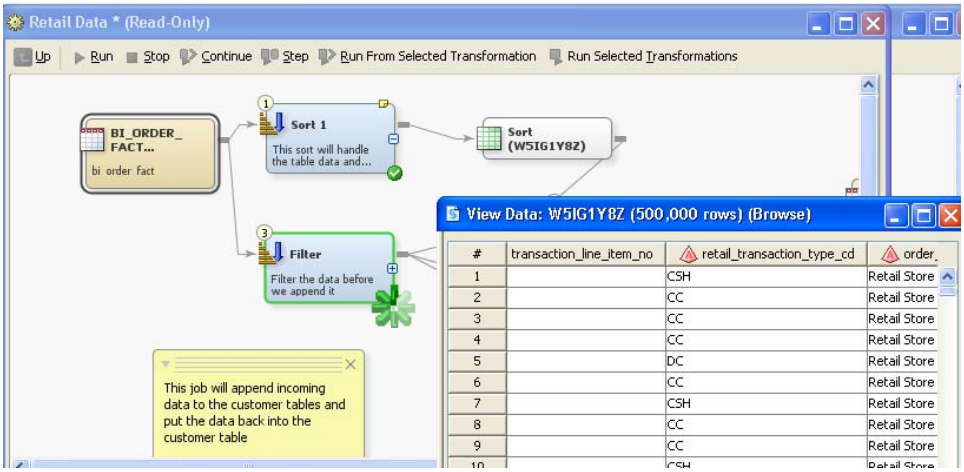


Figure 1.13: During the execution of **Filter** in the running job, output from a previous step is being reviewed.

Runtime Progress Indicators and Status

Often it is not enough to ensure that a job completes successfully. When working with large data flows, it is also important to ensure that the job performs well. SAS Data Integration Studio can capture runtime 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.

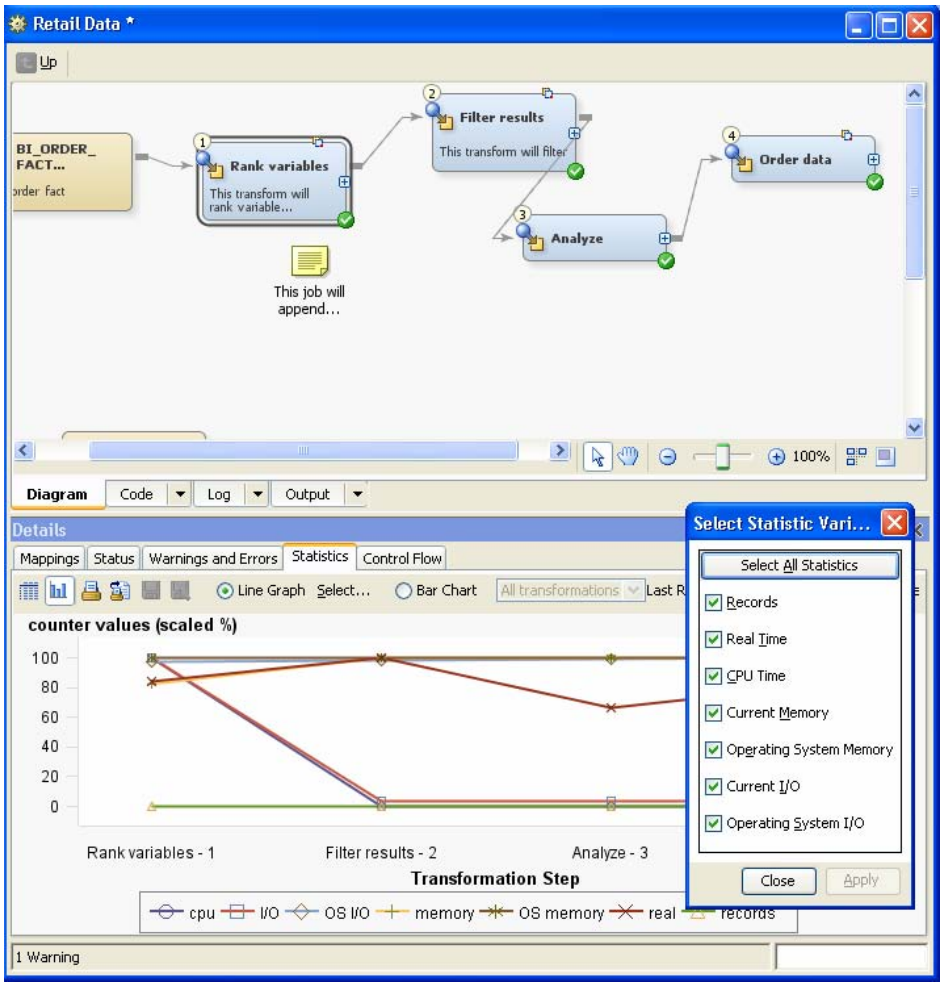


Figure 1.14: During execution of the **Retail Data** job, statistics have been collected for each step.

Saved statistics are available for each transformation step, and can be viewed graphically as shown in Figure 1.14 or in a tabular report (see Figure 1.15). In addition, data can be saved to a file for additional processing later. Various type 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 running with ARM enabled.

Another helpful statistic shows the number of records (or rows) of data processed. This 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, suggesting 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.

Figure 1.15: Runtime statistics are shown 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 various reasons you might have a node that you want to run before other nodes in the job run because it does some data setup. In this case you want to order the steps so that the job runs them in the right sequence. Another reason is to control the 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 allows you full control over run order. The order in which the steps will run is indicated by a number in the top left-hand corner of the node. The **Control Flow** tab allows you to manipulate the order of the steps as shown in Figure 1.16.

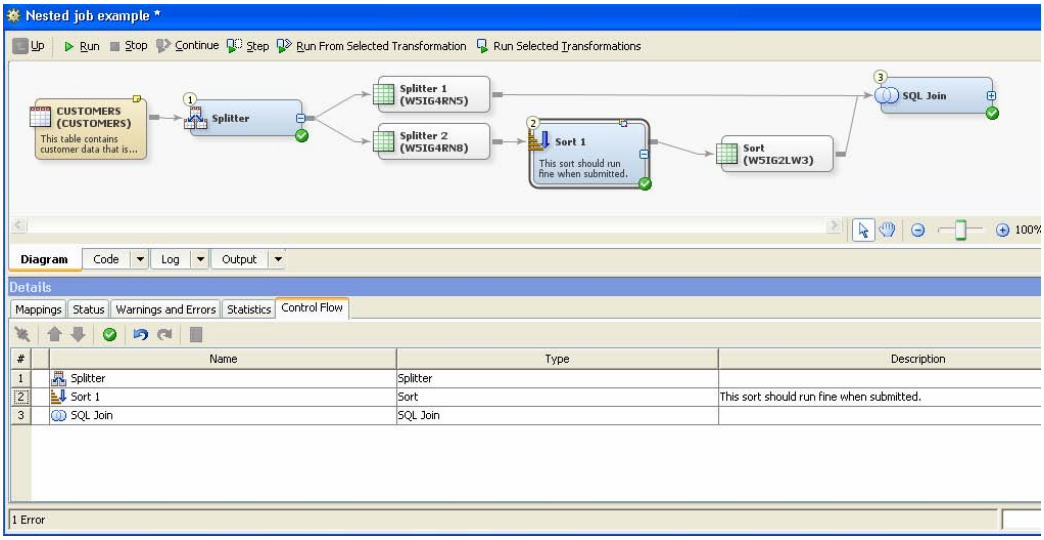


Figure 1.16: The order of nodes is in the standard form, as shown by values in the upper left of each transformation step. The lower panel shows this order in a tabular form, where the order can be altered, if needed.

In the new world of SAS Data Integration Studio 4.2, the updated 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 runtime performance. It also supports many useful workflow features to make you more productive when creating your jobs.

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 metadata management, security, as well as prompts and parameters for generated transformations are beneficial, as will be shown.

Metadata

SAS 9.2 makes available 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 are 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.

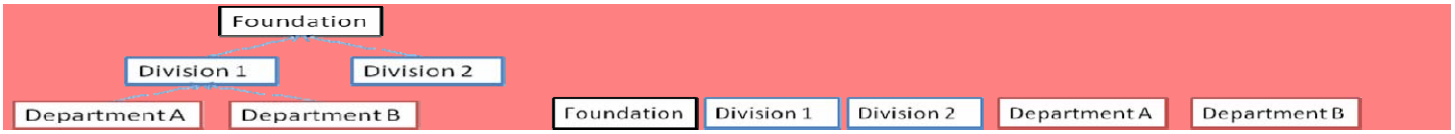


Figure 2.1: This figure shows SAS 9.1 dependent repositories (left) and SAS 9.2 independent repositories (right)

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 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 now allowed, but can be problematic in other cases. Someone who connects to the Foundation repository sees a different view, however. In this case, that person only sees metadata present in the Foundation repository. Another effect of this configuration is that someone connecting to Department A 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 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.

Figure 2.2: This figure shows metadata folders and repositories in SAS Data Integration Studio 4.2, based on SAS 9.2. Separate repositories for Corporate, Finance, and HR are seen as separate folders.

In the preceding figure, no Foundation repository is shown. Simply the root of Folders shown. In addition, separate divisions like Finance and HR 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 of 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, even in an encrypted form, is a concern.

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.

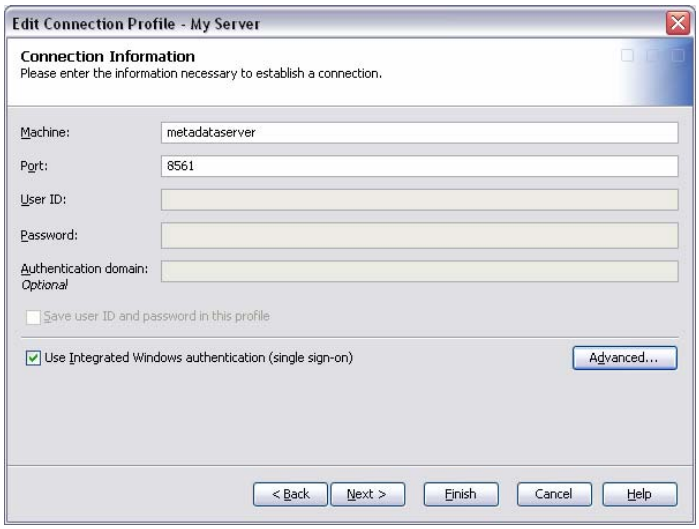


Figure 2.3: Integrated Windows authentication can be selected to avoid entering a user ID and password to use for SAS server connections.

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 as well as any workstation servers needed to view data, execute data integration flows, or perform other activities with SAS servers. This is recommended when a user's same login 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 passwords are looked up from Windows at runtime, the information is always current.

SAS 9.2 Dynamic Prompts

A major improvement of SAS 9.2 is the ability to use dynamic prompts in reports, filters, and SAS Data Integration Studio transformations. When used in transformations, dynamic prompts allow 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.

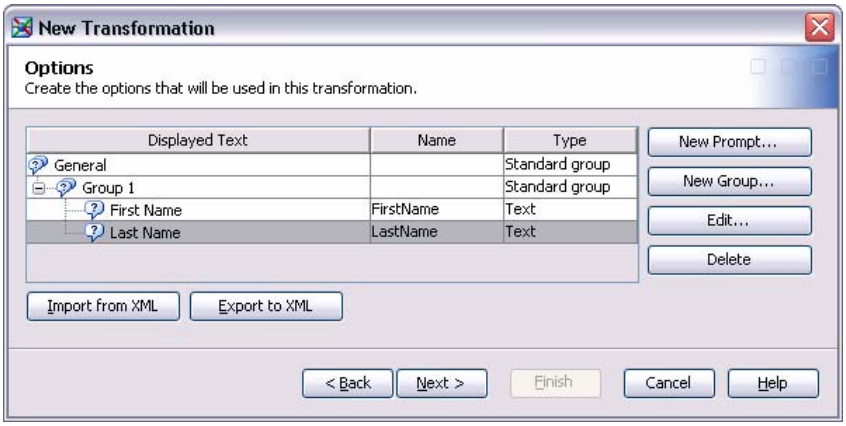


Figure 2.4: This figure shows a SAS Data Integration Studio transformation design that uses two related prompts. In this case, the transformation used requires that a first and last name be selected.

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

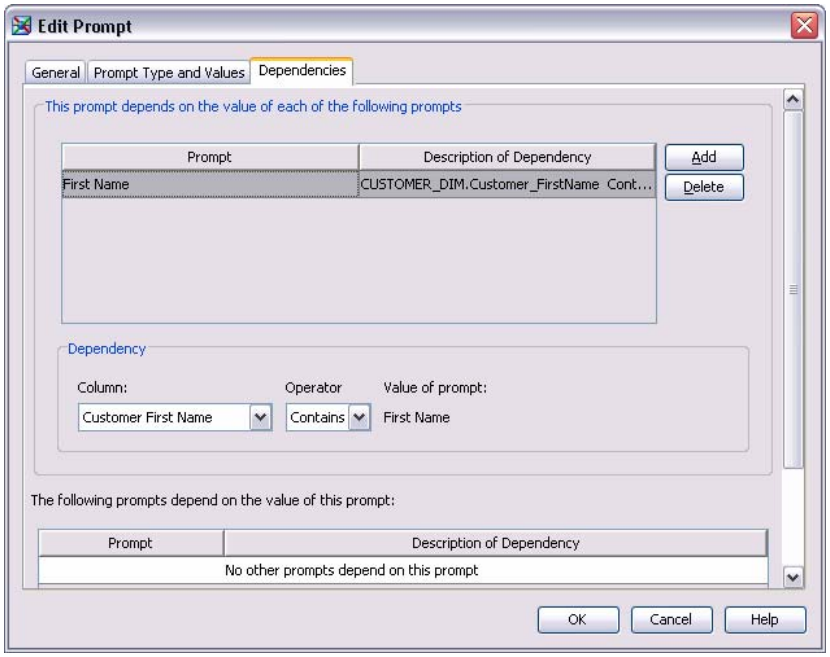


Figure 2.5: This figure shows how to assign a dependency between First Name and Last Name in a customer table from which possible first and last name data pairs are chosen.

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

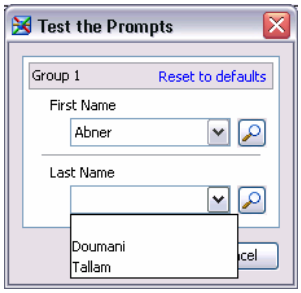


Figure 2.6: In this example, if the First Name "Abner" is entered, only the matching Last Names for individuals in the Customers table can be chosen. In this case, there are two Last Names possible for people named "Abner."

More Interactive Option Selection

In addition, a richer user interface is now available. 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.

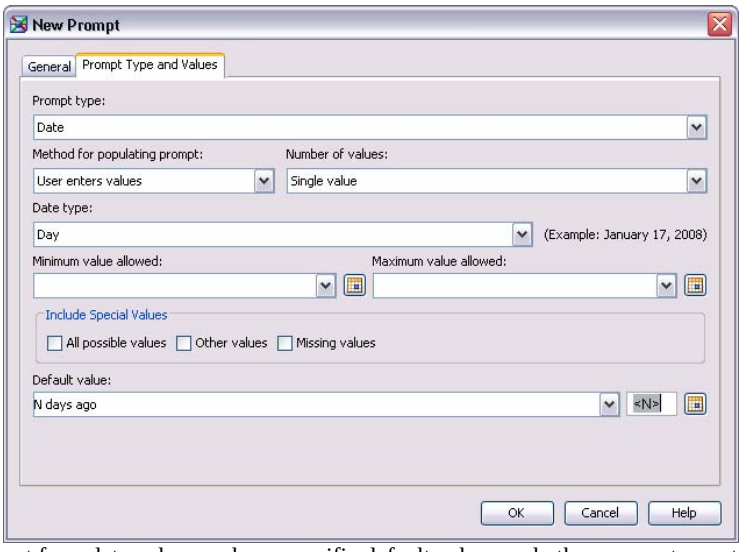


Figure 2.7: Designing an option prompt for a date value can have specific default values and other parameters set.

When using this transformation, users can select a drop-down calendar to simplify their choice, as well as to ensure the right format of data is entered.

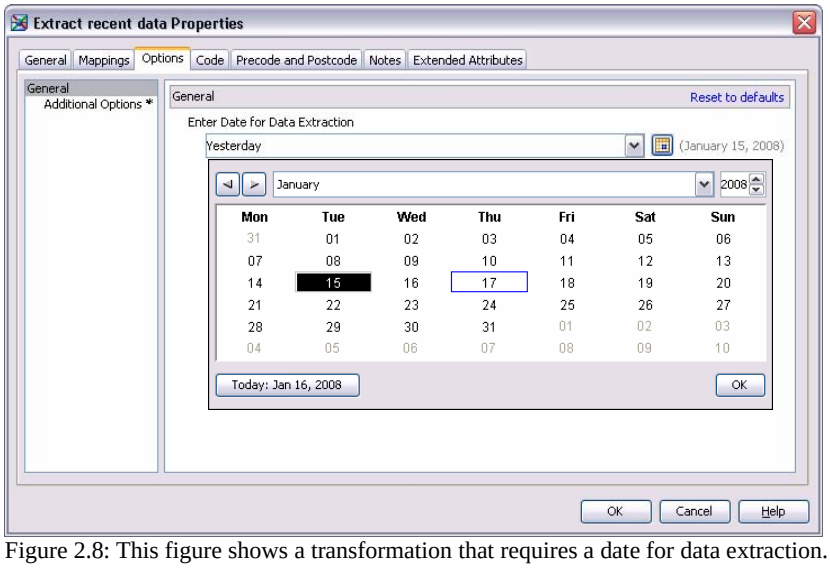


Figure 2.8: This figure shows a transformation that requires a date for data extraction.

Users of this transformation can use a graphical calendar to select a date value for their selection.

TRANSFORMATION ENHANCEMENTS AND ADDITIONS

Up to this point, you have seen how SAS Data Integration Studio 4.2 leverages new functionality in the SAS 9.2 Platform and previewed many of the major enhancements to the user interface, including design capabilities, debugging, and runtime statistics. Another important component of SAS Data Integration Studio is the rich set of table-level and column-level transformations it provides.

In this section we will highlight just a few of the transformations and a quick glimpse at some new metadata reporting capabilities including:

- change data capture
- optimized table loaders
- data quality integration
- impact analysis
- metadata reporting

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, greatly reducing processing requirements as compared with needing 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 needed information in a timeframe that meets expectations or service level agreements (SLAs).

Many database vendors deliver functionality in their products that will capture these changes and store 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 will also provide details about when the actual change took place. This becomes imperative when trying to synchronize the contents of the staging file with the content in the data warehouse in an effort to guarantee accuracy of the resulting data.

In SAS Data Integration Studio 4.2, additional functionality was added to support the CDC capabilities 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, as well as 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.

Figure 3.1 shows the list of CDC Transformations that are part of SAS Data Integration Studio 4.2. These transformations can be used in any job flow for reading and processing the changed records from any of the supported sources.

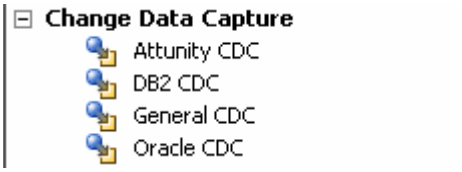


Figure 3.1: This figure shows a list of Change Data Capture Transformations available in SAS Data Integration Studio 4.2.

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.

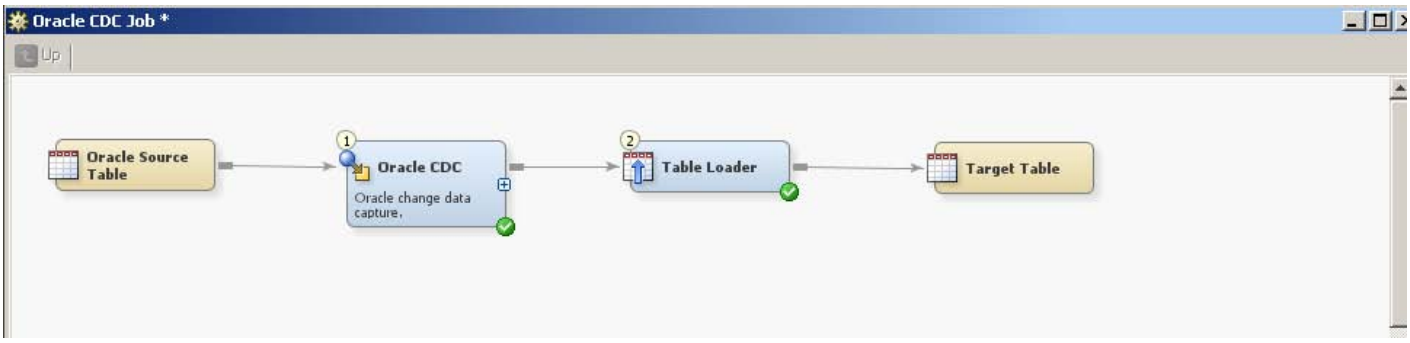


Figure 3.2: This figure shows an example of an Oracle CDC job flow using the Oracle CDC transformation. Here, newly-changed records are extracted and placed in a target table for further processing.

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

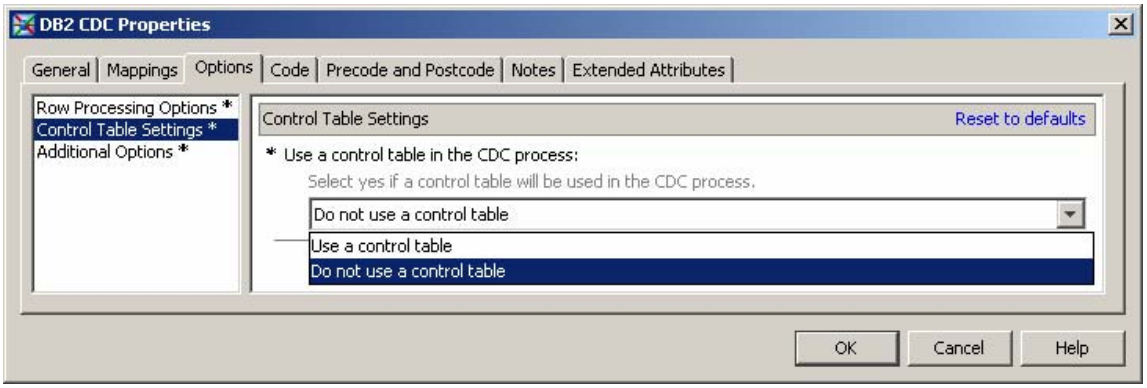


Figure 3.3: This figure shows setting Control Table options for a DB2 CDC source table.

Figure 3.4 shows some of the row processing options 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.

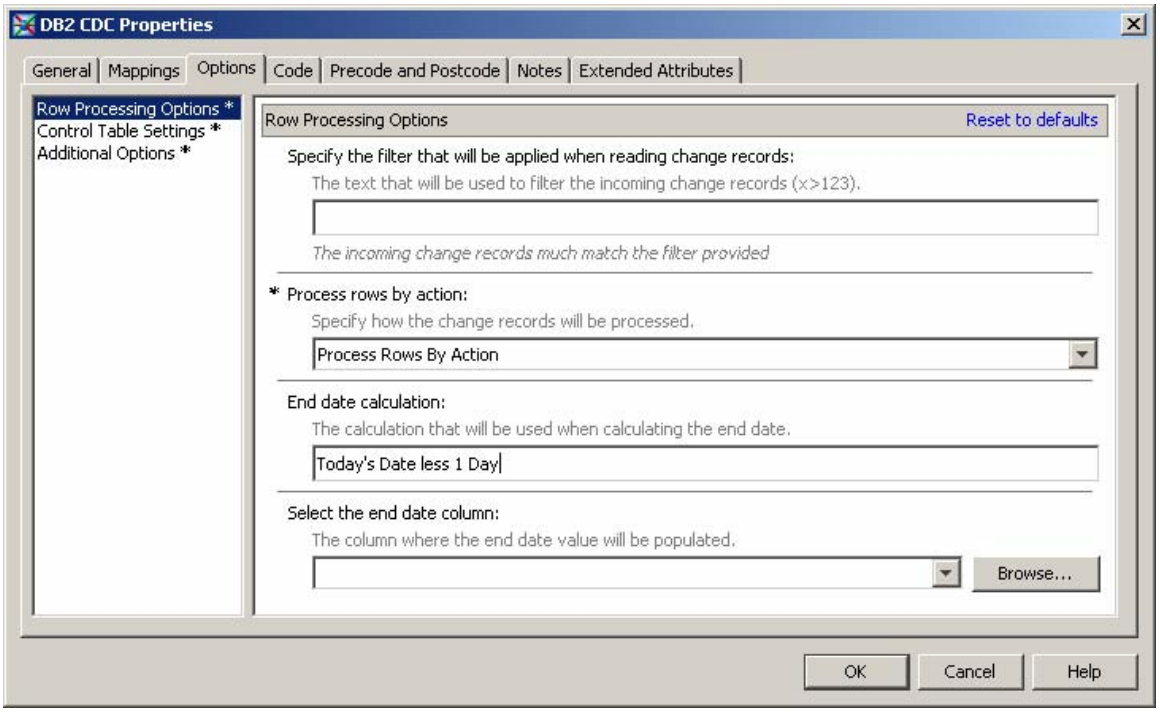


Figure 3.4: This figure shows Row Processing Options.

Optimized Support for Loading and Teradata

Another transformation that was enhanced in SAS Data Integration Studio 4.2 is the Table Loader. In the past the table loader transformation has been a general purpose loader that supported storage in both SAS and other environments. However, there are cases in which more optimized support is needed for some storage and database systems. For example, in a previous release of SAS Data Integration Studio, a loader was created for the SAS Scalable Performance Data Server (SPD Server). The SPD Server Loader was designed to be optimized specifically to offer configuration options and generate the necessary steps to ensure the best performance possible when the target was SPD Server.

In SAS Data Integration Studio 4.2, additional enhancements were made to the existing Table Loader to be more database-aware, meaning that the loader can recognize the target and provide a different set of options or recommendations specifically designed for that database. For example, when Teradata is the target, there is an option under the Load Technique for a MultiLoad upsert. MultiLoad is a load utility or protocol supported by Teradata that is optimized for loading data into an existing table. Upsert is a technique supported within the MultiLoad utility.

In the following figure, various options that are only available based on the target with which the loader is associated. In this example, the loader is associated with a Teradata target. This would result in the Load Style, Techniques, Constraints, and Index options being tailored specifically for Teradata. The available options would be different for situations when the target associated with the loader was SAS storage, Oracle, DB2, or another target.

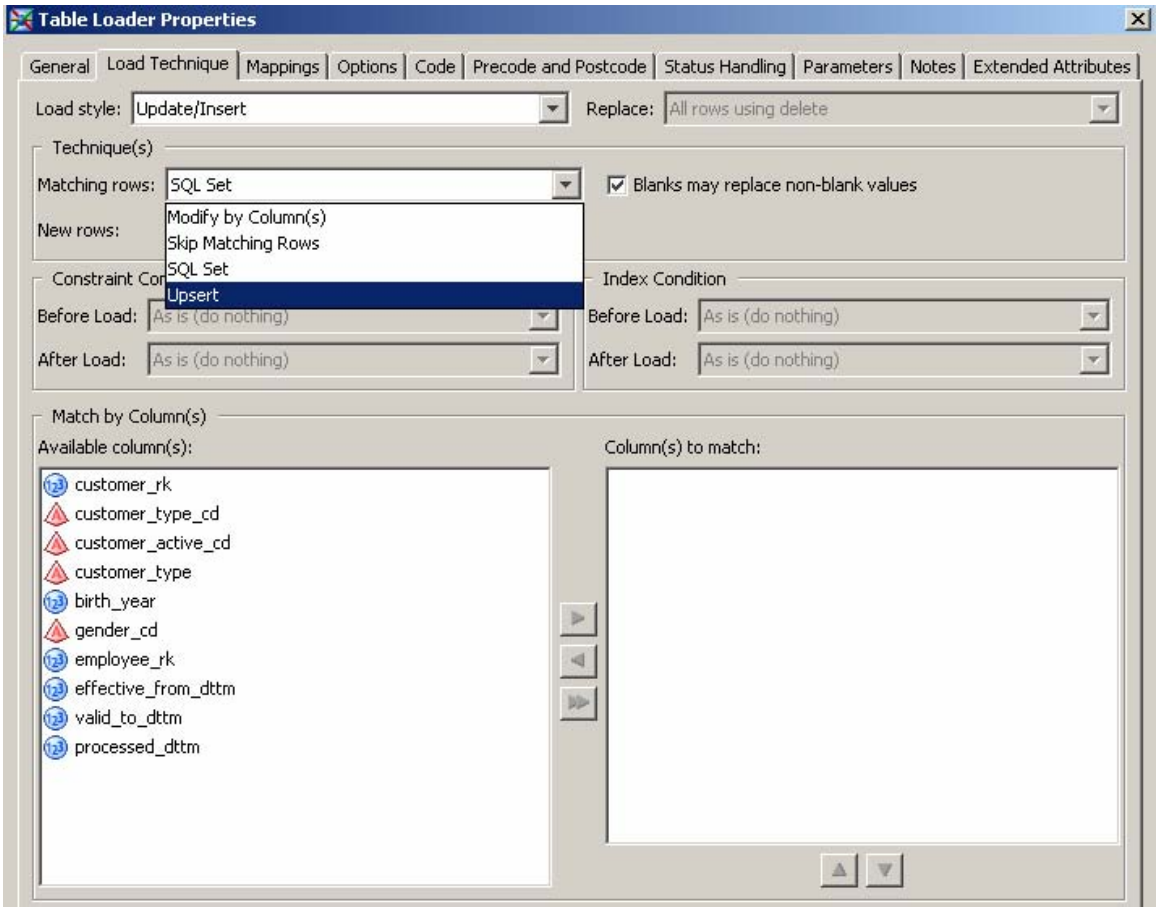


Figure 3.5: This figure shows load-specific options based on a Teradata load target.

Integration with DataFlux Jobs and Services

Data quality continues to be important for users of SAS Data Integration Studio. Integration with DataFlux (a SAS subsidiary) is crucial to meet the continued demand, growth, and expectations on the quality of data that is a result of 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 allows 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. The knowledge of the parameters and values that are expected for these jobs and services to execute are then shared between the two environments through seamless integration.

Figure 3.6 shows selection of the desired server and the associated services available for execution.

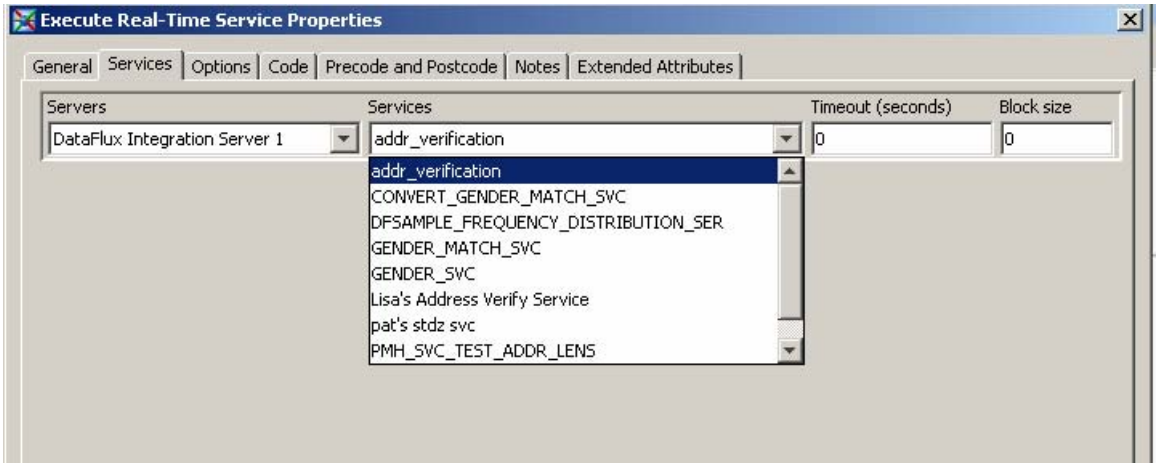


Figure 3.6: After you select the desired server, associated services become available for execution.

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

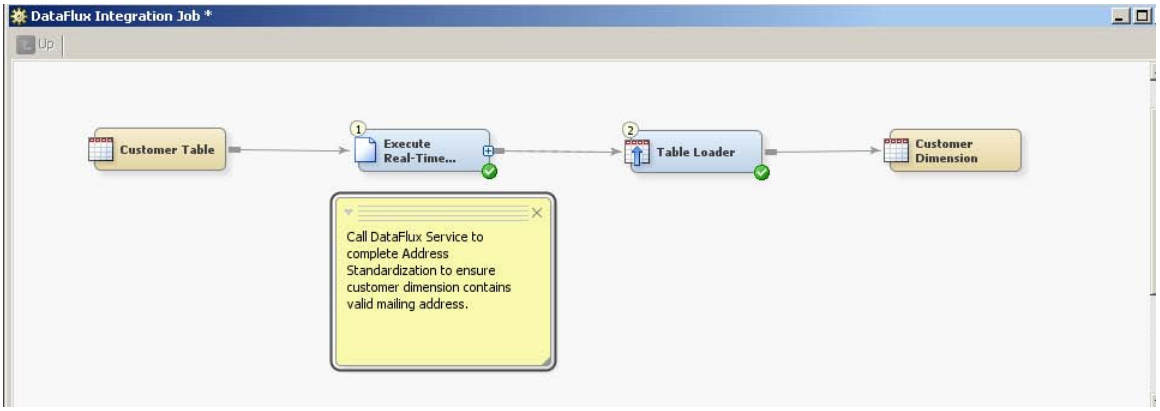


Figure 3.7 This figure shows a job flow that is executing a DataFlux Real-time Service for Address Verification.

Metadata Reporting

Metadata contains information that can be useful to the designers and developers of data integration processes. SAS Data Integration Studio 4.2 introduces the first phase of a multiphase delivery of metadata reporting.

Out of the box in SAS Data Integration Studio 4.2 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 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.

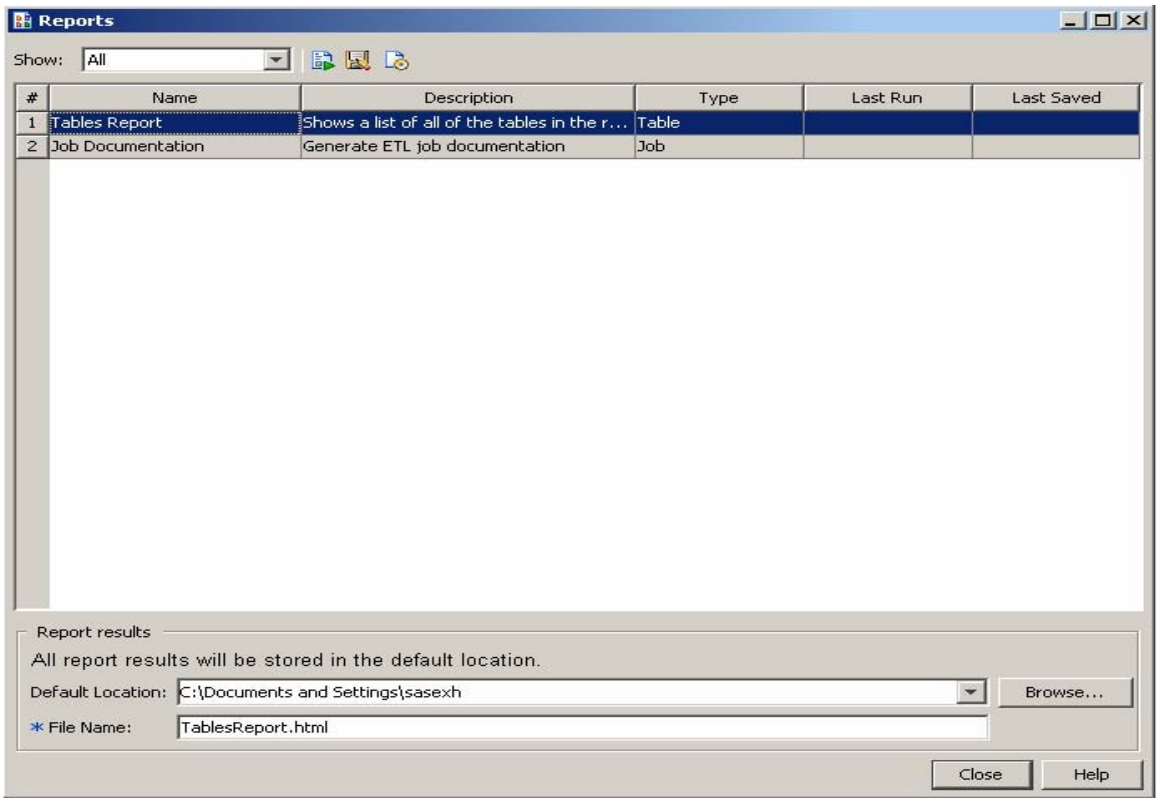


Figure 3.8: From the Reports menu in SAS Data Integration Studio, available reports are displayed. In this case, the default standard reports are present.

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 allows navigation and searching of 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 available from a Job Documentation report.

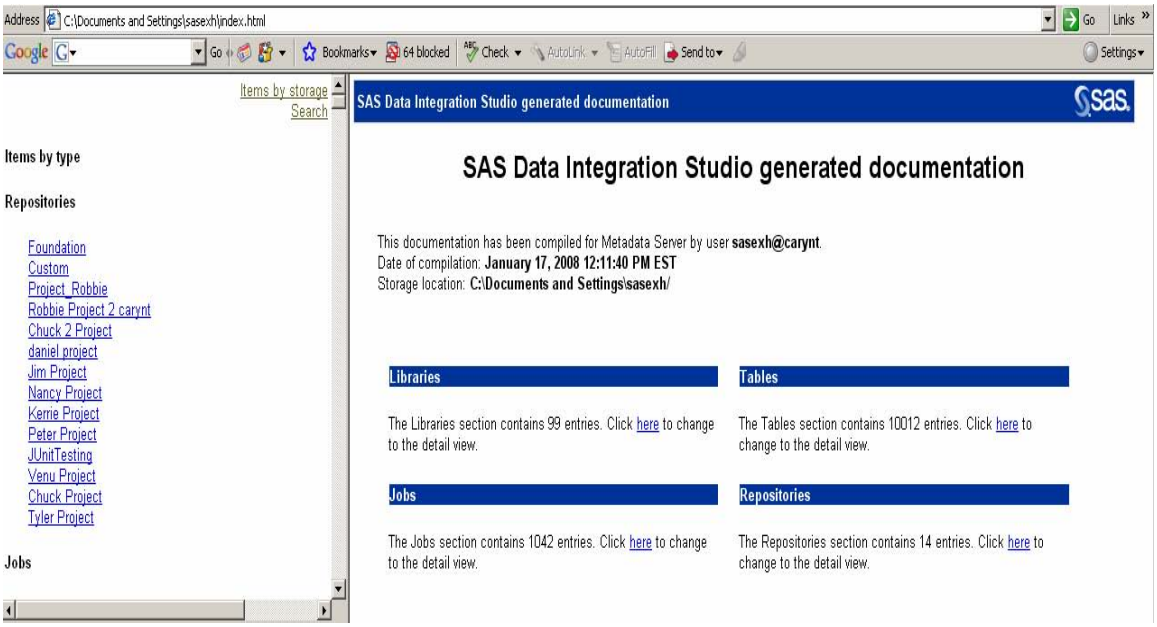
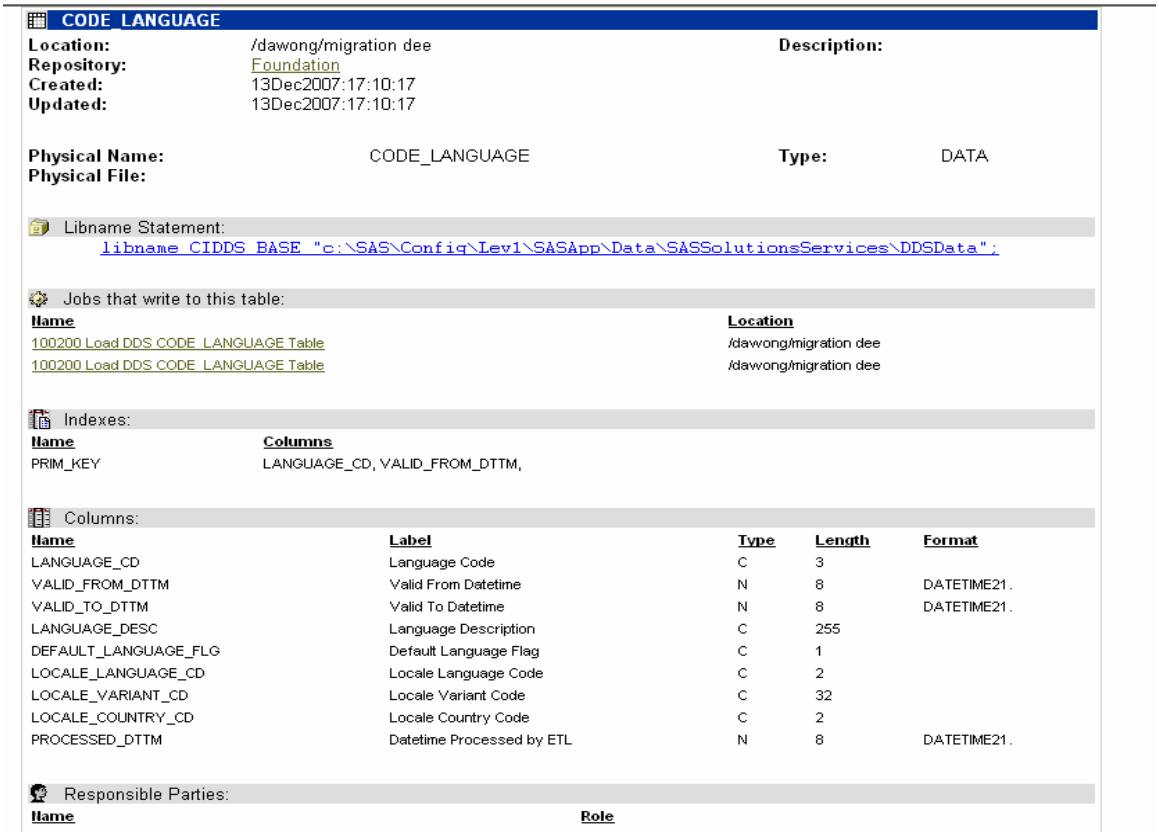


Figure 3.9: This figure shows the main index. A summary of libraries, tables, jobs, and repositories are available.



CODE_LANGUAGE

Location: /dawong/migration dee
Repository: Foundation
Created: 13Dec2007:17:10:17
Updated: 13Dec2007:17:10:17
Description:

Physical Name: CODE_LANGUAGE
Physical File:
Type: DATA

Libname Statement:

```
libname CIDDS BASE "c:\SAS\Config\Lev1\SASApp\Data\SASSolutionsServices\DDSData";
```

Jobs that write to this table:

Name	Location
100200 Load DDS CODE_LANGUAGE Table	/dawong/migration dee
100200 Load DDS CODE_LANGUAGE Table	/dawong/migration dee

Indexes:

Name	Columns
PRIM_KEY	LANGUAGE_CD, VALID_FROM_DTTM,

Columns:

Name	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:

Name	Role

Figure 3.10: This figure shows 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 4.2 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. Quick ways to scan through warnings and errors that can occur during execution help focus attention on these problems. Smarter and faster mapping and propagation support in jobs means you can focus on key content rather than details of transformation-level mappings in most cases.

Advanced users have new capabilities for managing order dependencies for steps within jobs, for handling jobs within jobs, and for managing 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, 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 4.2. 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.

REFERENCES

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
Web: www.sas.com

Gary Mehler
SAS Institute Inc.
Cary, NC 27513
Work Phone: (919) 677-8000
Fax: (919) 677-4444
E-mail: Gary.Mehler@sas.com
Web: www.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
Web: www.sas.com