Paper SAS605-2017

Data Grids in Business Rules, Decisions, Batch Scoring, and Real-Time Scoring

Carl Sommer, Chris Upton, and Ernest Jessee, SAS Institute Inc.

ABSTRACT

Users want more power. SAS® delivers. Data grids are a new data type available to users of SAS® Business Rules Manager and SAS® Decision Manager. These data grids can be deployed to both batch and web service scoring for data mining models and business decisions. Users will learn how to construct data with grid data types, create business rules using high-level expressions, and deploy decisions to both batch and web services for scoring.

INTRODUCTION

Decision management systems perform a very straightforward job: given a set of input values, apply an algorithm of models, conditional logic, and business rules to produce an output set of values. Regardless of the problem domain, the standard approach for near-real-time decision systems is that the input and output vectors of data are entirely composed of scalar values (numeric and character data). Figure 1 provides an example:

$210,000	**$24,000**	**Auto Loan**	**$45,000**	**2017-XYZ**	**$300,000**
debts	*amount*	*type*	*salary*	*request id*	*assets*

Figure 1. Traditional Scalar Value Input Data Vector

To improve the decision process in this example, a more finely grained examination of details about the individual debts or assets might be necessary. This creates architectural challenges for the decision process, and the following time-honored, but problematic, techniques are often attempted to enhance the input data vector:

- Add some number of columns (for example, ***DEBT1-DEBT10***). This results in wasteful, rigid structures that will require additional conditional processing to use the data within the decision process.
- Add a varying number of columns, perhaps along with a column that details the number of columns to expect (for example, ***DEBTS_COUNT=5***, with columns ***DEBT1-DEBT5***). This strategy is likely to prove even more problematic for the decision process to implement, and is also unwieldy for batch-oriented table input.
- Join the debts or assets of interest from other sources of data when the decision process executes. While this seems like the best answer, the configuration, performance, and dependency complexities that are brought on by the join introduce fragility to the decision process.

SAS Decision Manager 3.2 provides a more robust alternative, through adding support for a **data grid** data type. Conceptually, a data grid is a table within a cell, as shown in Figure 2. These data grids are supported by a set of methods that allow full exploitation of the data contained within a given data grid. Note that any number of data grids can be used in a decision flow.

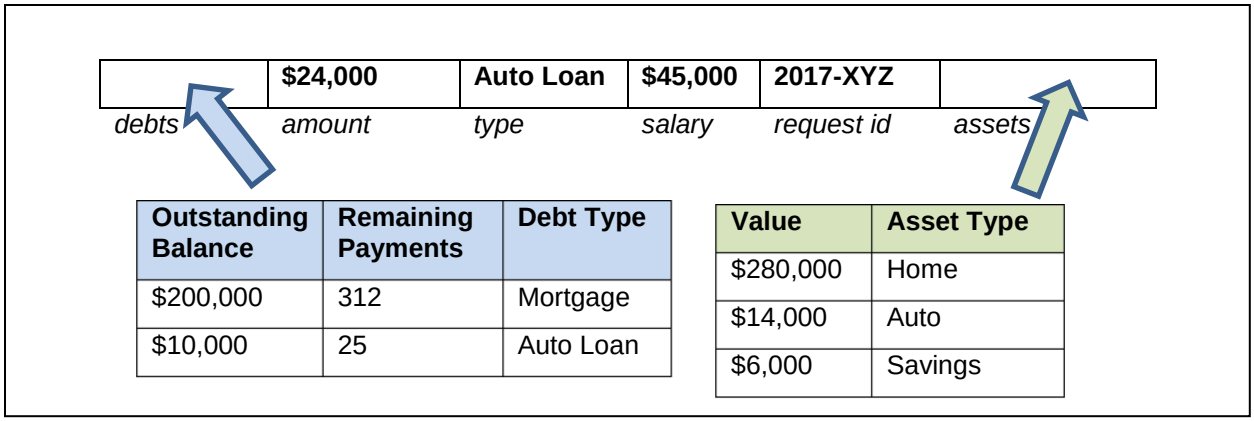
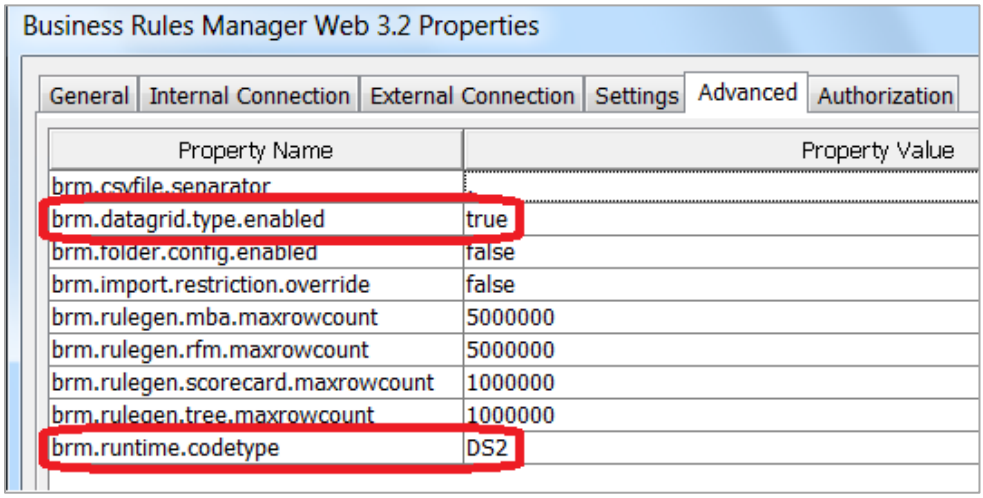


Figure 2. Input Data Vector with Scalar Values and Data Grids

Data grids support any number of rows, including an empty grid of no rows. This feature addresses the need for a varying number of cells, and it also removes the requirement to perform joins at decision execution time.

ENABLING DATA GRID FUNCTIONALITY IN SAS DECISION MANAGER

By default, data grids are not enabled for use. Enable the use of data grids by navigating the following path from the **Plug-ins** tab in SAS® Management Console: **Application Management Configuration Manager → SAS Application Infrastructure → Enterprise Decision Manager 3.2 → Business Rules Manager Web 3.2**. Once there, set the **brm.datagrid.type.enabled** configuration property to **true**, and set the **brm.runtime.codetype** configuration property to **DS2**, as shown in Display 1:



Display 1. Enabling Data Grid Functionality in SAS Management Console

You must restart SASServer7 in order for changes to these configuration properties to take effect.

Once data grid functionality is enabled, data grids can be used by rule flows that were created by SAS Business Rules Manager. You can incorporate data grids into stand-alone rule flows or into rule flows that are included in a decision that was created by SAS Decision Manager.

Note that data grid functionality is only available when generating and executing DS2 code. The generated DS2 code is capable of executing in-database using SAS® Code Accelerator.

DATA GRID TERMS

Data grid terms are defined in SAS Business Rules Manager vocabularies, just as all other character and numeric terms are. Data grid terms can be read from input, written to output, or populated during execution of the rule flow. For input or output purposes, a data grid is represented by a JSON (JavaScript Object Notation) string. The JSON string has a metadata component, which describes the columns and data types in the data grid, and a data component, which lists the tuples of data for each row, using the column order that is expressed in the metadata section.

The column data types that are supported are **decimal** (for all numeric data) and **string** (for character data). Column names in data grids are case-insensitive, but character data values are case-sensitive. Missing values are represented in this JSON representation by the word **null**.

Example 1 shows the JSON string for a data grid consisting of two columns (***ASSET_VALUE***, and ***ASSET_TYPE***) with three rows of data:

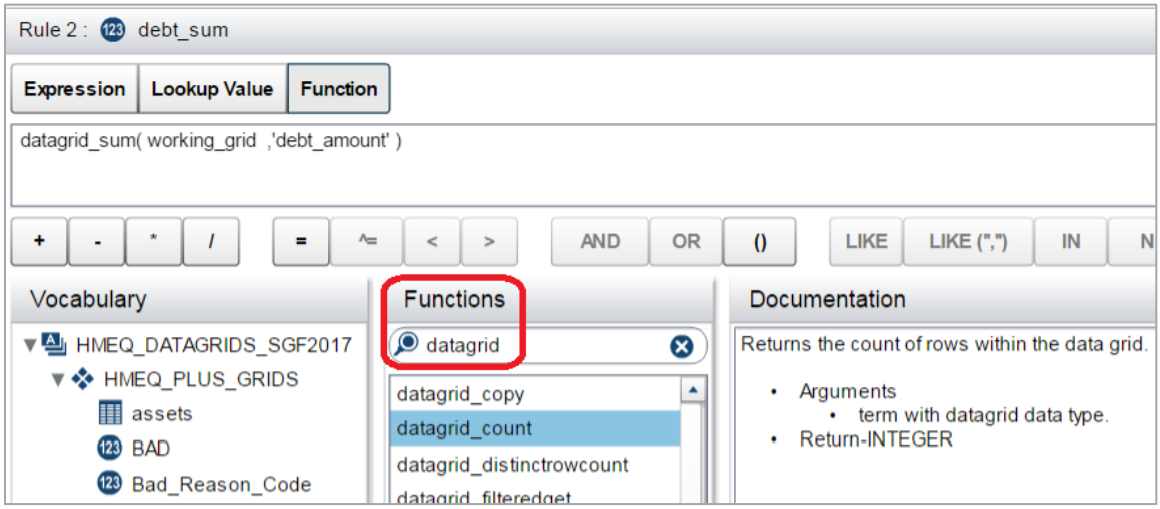
```
[  {  "metadata": [{" VALUE": "decimal"},{"ASSET_TYPE": "string"}]},
   {  "data": [[280000,"Home"],[14000,"Auto"],[6000,"Savings"] }  ]
```

Example 1. JSON Representation of a Data Grid

Note that the columns within the data grid are not defined in the vocabulary and are not available as terms in the rule sets. As we will see, functions are provided to fully manipulate the information in the data grid.

USING DATA GRIDS IN BUSINESS RULES EXPRESSIONS

A robust set of data grid functions is provided for use in condition and action expressions in SAS Decision Manager. These functions are available in the expression editor of SAS Business Rules Manager. When working with data grid functions in the expression editor, you can filter the list to include only the data grid functions by applying a filter for **datagrid** as shown in Display 2.



Display 2. Data Grid Functions in the Expression Editor

Assuming that the appropriate argument and return types are used, data grid function calls can be nested. For example, you can request the **datagrid_count()** of a data grid that was returned by a call to **datagrid_subsetByValue()**, as shown in Example 2:

```
= datagrid_count(datagrid_subsetByValue(assets, 'Asset_Type', 'Savings'))
```

Example 2. Nesting of Data Grid Function Calls

WHAT DATA GRID FUNCTIONS ARE AVAILABLE?

Data grid functions have been provided that allow for descriptive statistics on data grid columns, retrieval and setting of individual cell values in a data grid, creation of additional grids, and matching as part of statistics or grid creation. A summary of the available functions is provided in Table 1:

Functional Area	Functions available

Statistics	`datagrid_count()`
	`datagrid_distinctRowCount()`
	`datagrid_gridMatchCount()`
	`datagrid_lookupMatchCount()`
	`datagrid_matchCount()`
	`datagrid_max()`
	`datagrid_mean()`
	`datagrid_median()`
	`datagrid_min()`
	`datagrid_sum()`

Get / Set values	`datagrid_filteredGet()`
	`datagrid_filteredSet()`
	`datagrid_filteredSetAll()`
	`datagrid_set()`
	`datagrid_setAll()`

Grid creation	`datagrid_innerJoin()`
	`datagrid_leftJoin()`
	`datagrid_lookupSubset()`
	`datagrid_rightJoin()`
	`datagrid_subsetByValue()`

Table 1. Data Grid Functions by Area

The function names are relatively self-documenting; for complete information, refer to *SAS Decision Manager 3.2: Using Data Grids*, which is available on request from SAS Technical Support.

DATA GRID FUNCTION PARAMETERS

The expression editor presents the documentation for each data grid function. The data grid functions share a common parameter framework. Table 2 provides a summary of the types of arguments for the various data grid functions:

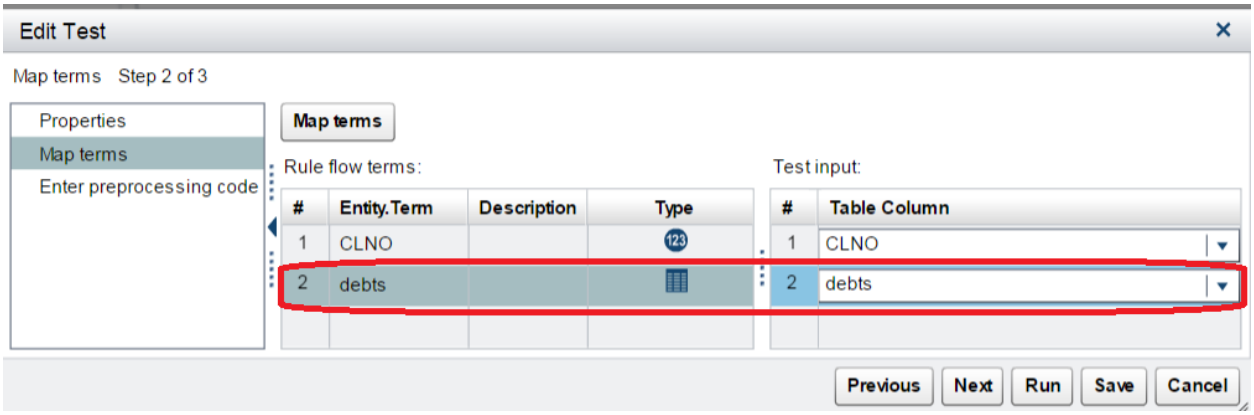
Parameter Specification	Notes	Example usage
Term of DATAGRID data type	Specified as a term name (no quotation marks).	`datagrid_count(`**`assets`**`)`
Column name inside the data grid	Specified as a literal value in single quotation marks, or as a character term name that resolves to the data grid column name.	`datagrid_min(assets, '`**`Value`**`')`
Lookup name	Specified as a literal value in single quotation marks. This is the convention for existing SAS	`datagrid_lookupSubset(` `  '`**`AssetLookup`**`',`

Parameter Specification	Notes	Example usage
	Business Rules Manager lookup functions.	assets, 'Asset_Type')
Comparison operator	Specified as a literal character value in single quotation marks, or as a character term name that resolves to the comparison operator.	datagrid_subsetByValue(assets, 'Asset_Type', **'EQ'**, **'Savings'**)
Comparison value	Literal value for comparison, or name of a term that contains the comparison value.	
Row number inside the data grid	Specified as a literal numeric value, or as a numeric term name that resolves to row number.	datagrid_get(assets, 'Asset_Type', **2**)

Table 2. Data Grid Functions Argument Specification Summary

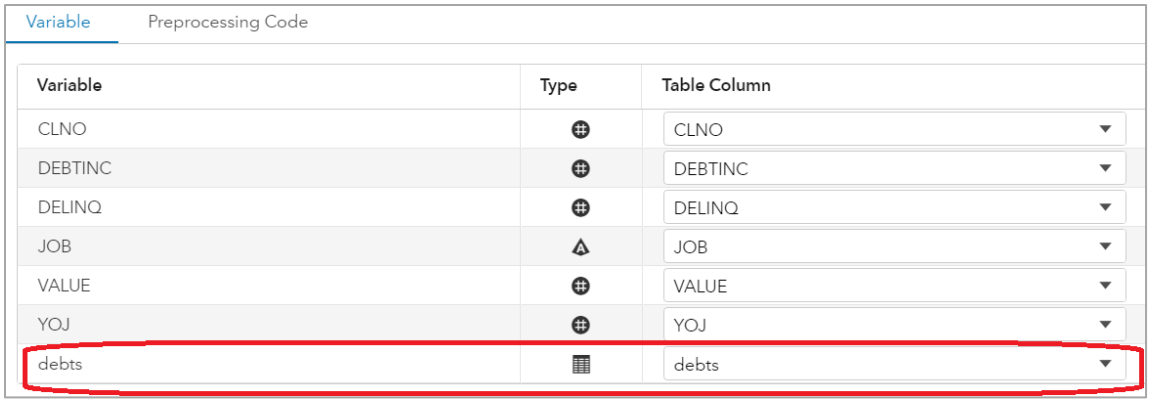
PREPARING DATA GRIDS FOR USE IN THE TEST INTERFACE

SAS Decision Manager and SAS Business Rules Manager both support a test interface, which is available from the graphical user interface. The test interface enables you to map a data source (a table of data, already registered in metadata and physically available to your SAS workspace server) as input to a decision flow or rule flow. The data grid columns in this table must be character columns that contain the appropriate JSON for the data grid. Display 3 shows such a mapping. In this case, the column name in the data source table and the data grid term name that is used in the rule flow are identical.



Display 3. Test Interface Mapping in SAS Business Rules Manager

In Decision Builder, the mapping concept is the same; however, the user interface appears as shown in Display 4:



Display 4. Test Interface Mapping in Decision Builder

The data grid term can be mapped to any character column in the input table. It is the user's responsibility to make sure that the column contains the JSON representation for a data grid.

The **%DCM_SERIALIZEGRID** macro has been provided to aid users in the tedious task of creating JSON. Users may also find the **%DCM_MERGESERIALIZEDGRIDS** macro helpful if multiple data grid columns are being prepared.

THE %DCM_SERIALIZEGRID MACRO

The **%DCM_SERIALIZEGRID** macro creates a table with a data grid JSON column and a specified number of class columns from an input table. For example, suppose you wanted to create data grids based on several, but not all, columns from **SASHELP.CARS**, and you wanted to have a separate data grid for each MAKE of cars. Example 3 shows the code to create an output table called **WORK.CarsGridByMake** that has a data grid JSON string column called **carsGrid** and the **MAKE** column from the original **SASHELP.CARS** table.

```
%dcm_serializeGrid(
  gridSourceTable=sashelp.cars,
  outputTable=work.carsGridByMake,
  gridColName=carsGrid,
  gridCols=mpg_city msrp weight,
  classvars=make)
```

Example 3. Creating Data Grid JSON Using %DCM_SERIALIAZEGRID

An invocation of this macro creates a single data grid column.

Each data grid in this example has the columns **MPG_CITY**, **MSRP,** and **WEIGHT** for the specific value of **MAKE**. One row is written to the **WORK.CarsGridByMake** table for each distinct value of **MAKE**. Output 1 shows the values of the **MAKE** and **CarsGrid** columns for the first row of the output table. The data grid contains 7 rows of data.

```
MAKE=Acura
carsGrid=[{"metadata":[{"MPG_CITY":"decimal"},{"MSRP":"decimal"},{"WEIGHT":"
decimal"}]},{"data":[[24,23820,2778],[17,36945,4451],[18,43755,3880],[17,897
65,3153],[18,46100,3893],[22,26990,3230],[20,33195,3575]]}]
```

Output 1. Output of %DCM_SERIALIZEGRID

In order to create multiple data grid columns, it would be necessary to specify additional invocations of this macro for the required data grids, and then merge the tables together.

THE %DCM_MERGESERIALIZEDGRIDS MACRO

The **%DCM_MERGESERIALIZEDGRIDS** macro can used if multiple tables need to be merged using a single key column. One table in the merge must be considered the source merge table; this is where the traditional scalar values originate. Each of the other tables contributing to the merge has a merge key column and a data grid column. The output table consists of the scalar columns and the merge key column from the source merge table, and the matching data grid columns from the specified grid tables.

Consider a scenario with a table of asset data grids, a second table with debt data grids, and a third table with loan requests. In this scenario, we want to create a table with all the data grids and loan request data combined as a row that will provide input for a decision.

The ***OWNER_ID*** columns from the **ASSETS_GRID** and **DEBTS_GRID** tables match each other, and they also match the ***ID*** column from the **LOAN_REQUESTS** table. The code shown in Example 4. Merging Data Grid Tables would be used to perform this merge:

```
%dcm_mergeSerializedGrids(
   mergeTable=loan_Requests,
   mergekey=ID,
   outputTable=loanRequestData,
   gridTables=assets_Grid debts_Grid,
   gridMergeKeys=owner_ID owner_ID,
   gridColumns=assets debts)
```

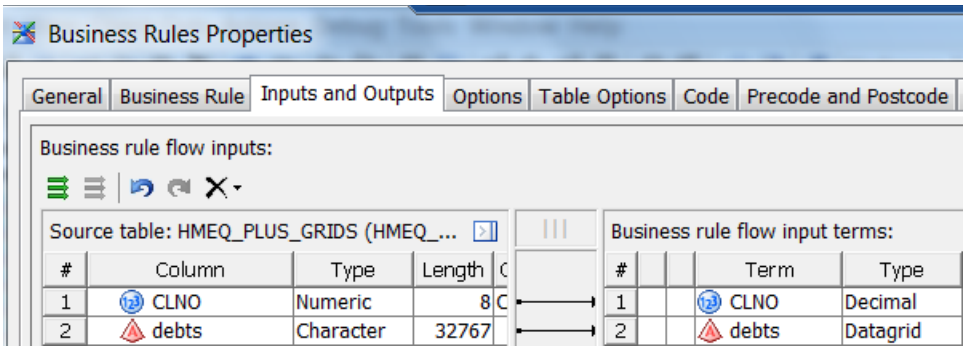
Example 4. Merging Data Grid Tables

Merges with more complexity would instead require user-supplied code.

USE OF DATA GRIDS IN BATCH ENVIRONMENTS

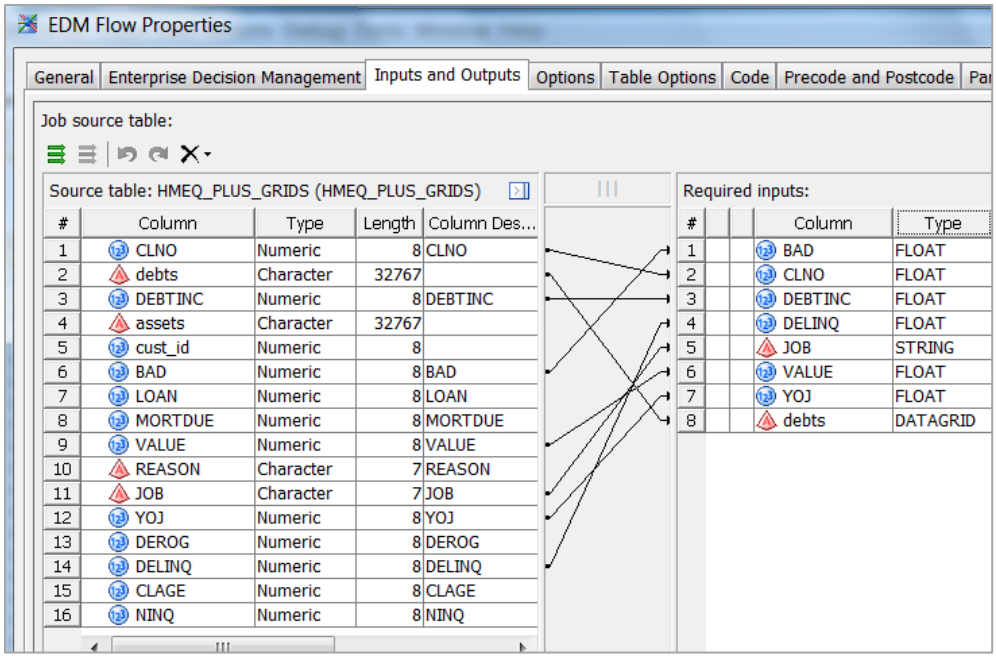
Decision flows and rule flows can both be published to SAS® Metadata Server for use in defining jobs in SAS® Data Integration Studio. The process for publishing rule flows has already been covered in David Duling's 2014 *Introduction to SAS Decision Manager,* and although data grids were not part of the solution when that paper was written, the mapping process is largely unchanged.

As with the mapping that is required for the test interface, a data grid term in a rule flow or decision flow is mapped to a character column from the input table, as shown in Display 5:



Display 5. Mapping of a Data Grid Term for a Rule Flow in SAS Data Integration Studio

Similar to rule flows, decision flows can be published to metadata for the creation of a job in SAS Data Integration Studio. The mapping for a decision flow works the same way, as shown in Display 6:



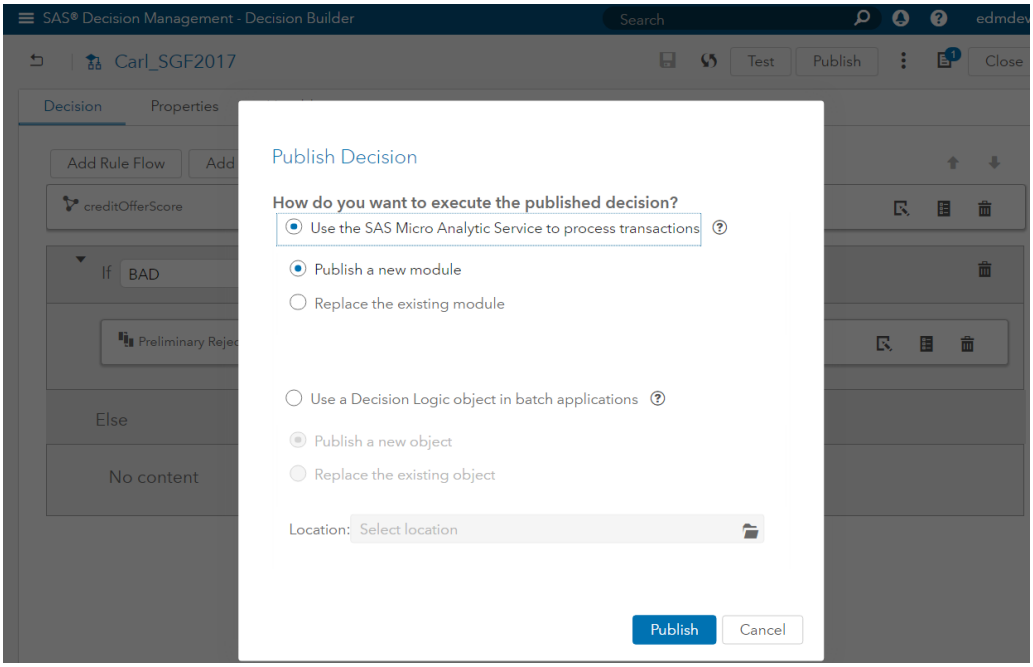
Display 6. Mapping of a Data Grid Term for a Decision Flow in SAS Data Integration Studio

Again, the user is responsible for ensuring that appropriate JSON is in the mapped input column.

USE OF DATA GRIDS IN WEB SERVICES

A hidden gem in SAS Decision Manager 3.2 is SAS® Micro Analytic Service. SAS Micro Analytic Service enables decisions to be published as DS2 package code, which can then be called as web services via standard RESTful web interfaces. The input and output payloads for these calls use JSON. For a comprehensive treatment of this topic, refer to Chris Upton's 2017 paper "*Power to the People: Web Service Scoring for the Masses.*"

A decision can be published to SAS Micro Analytic Service from the same dialog boxes that are used to publish to SAS Metadata server, as shown in Display 7:



Display 7. Publishing a Decision to SAS Micro Analytic Service

CONCLUSION

Augmenting traditional decisions with data grids enables more robust decisions, while avoiding the complexities of joining data at decision execution time. In addition, the use of data grids avoids introducing awkward architecture. The ability to work with rich and varying data backed by a solid architecture allows for more precise and more accurate decision-making. This paper shows a small part of the potential for data grid exploitation with SAS.

REFERENCES

Sparano, Steve, Charlotte Crain, and David Duling, 2014. “Introduction to SAS® Decision Manager.” *Proceedings of the SAS Global 2014 Conference*. Cary, NC: SAS Institute Inc. Available http://support.sas.com/resources/papers/proceedings14/SAS276-2014.pdf.

Upton, Chris and Lori Small, 2015. “Dynamic Decision-Making Web Services Using SAS® Stored Processes and SAS® Business Rules Manager.” *Proceedings of the SAS Global 2015 Conference*. Cary, NC: SAS Institute Inc. Available https://support.sas.com/resources/papers/proceedings15/SAS1787-2015.pdf.

Upton, Chris and Prasenjit Sen, 2017. “Power to the People. Web Service Scoring for the Masses.” *Proceedings of the SAS Global 2017 Conference*. Cary, NC: SAS Institute Inc. Available https://support.sas.com/resources/papers/proceedings17/SAS606-2017.pdf.

SAS Institute Inc. 2016. *SAS® Decision Manager 3.2: Using Data Grids*. Cary, NC: SAS Institute Inc. Available on request from SAS Technical Support.

ACKNOWLEDGMENTS

The authors of this paper would like to acknowledge and thank several staff members from SAS Institute, Inc for their help, including the following:

- Randy Evans, Senior Analytics Software Tester
- Prasenjit Sen, Senior Software Development Manager
- Ted Dyer, Principal Software Developer

RECOMMENDED READING

- *SAS® Decision Manager 3.2: User's Guide.*
- Ecma International. "Introducing JSON." Accessed March 1, 2016. http://www.json.org.

CONTACT INFORMATION

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

Carl Sommer
100 SAS Campus Drive
Cary, NC 27513
Carl.Sommer@sas.com
http://www.sas.com