Paper SAS1787-2015

Dynamic Decision-Making Web Services Using SAS® Stored Processes and SAS® Business Rules Manager

Chris Upton and Lori Small, SAS Institute Inc.

ABSTRACT

With the latest release of SAS® Business Rules Manager, decision-making using SAS® Stored Processes is now easier with simplified deployment via a web service for integration with your applications and business processes. This paper shows you how a user can publish analytics and rules as SOAP-based web services, track its usage, and dynamically update these decisions using SAS Business Rules Manager. In addition, we demonstrate how to integrate with SAS® Model Manager using SAS® Workflow to demonstrate how your other SAS® applications and solutions can also simplify real-time decision-making through business rules.

INTRODUCTION

Companies need more efficient ways to quickly and efficiently develop flexible decision logic, which is an integral part of dynamic operational processes. Decision management systems support agility by enabling the organization to quickly adapt to changing business requirements and providing integration between the back room model building and front room business decisions. Bringing together multiple stakeholders with varying requirements and challenging timelines to “operationalize” business rules can be a challenge. This paper uses a real business scenario to demonstrate how SAS users who implement rules based on analytical models can rapidly create business rules as SAS Stored Processes and integrate with them as standard SOAP-based web services.

BUSINESS SCENARIO

OVERVIEW

Decision management systems automate the process of making decisions, particularly day-to-day operational decisions. They improve the speed, efficiency, and accuracy of routine business processes, in part by reducing the need for human intervention. SAS® Decision Manager helps organizations manage data, business rules, analytical models, and optimization techniques. SAS Business Rules Manager and SAS Model Manager are integrated into SAS Decision Manager to achieve a consistent interface for easier accessibility and full support for the Decision Lifecycle referenced by the “Introduction to SAS Decision Manager” presentation at SAS Global Forum 2014.

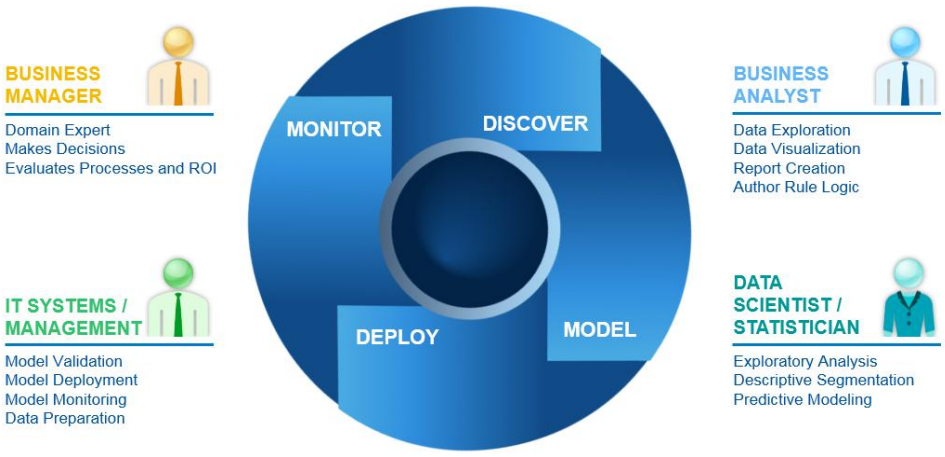


Figure 1. Decision Lifecycle

SAS also provides both horizontal and vertical industry solutions that interface with the operational business user. Some of these solutions support an additional business process abstraction layer that can be customized without changing the underlying implementation, thus providing some level of agility in adapting to changing business objectives. As business requirements evolve, the corresponding business processes can become quite complex. Integrating the process layer with business rules can simplify the process logic and provide a single point of customization with operational decisions.

The following dynamic approval workflow business scenario is based on an actual SAS customer engagement. Specifically, the business requirement specified approval of analytical models by one or more departments based on the model type and other model properties. This particular customer had 12 approval departments, where some models required approval from a single department while others required 2 or 3 approvals depending on the model type, resulting in a complex web set of control flows (Figure 2).

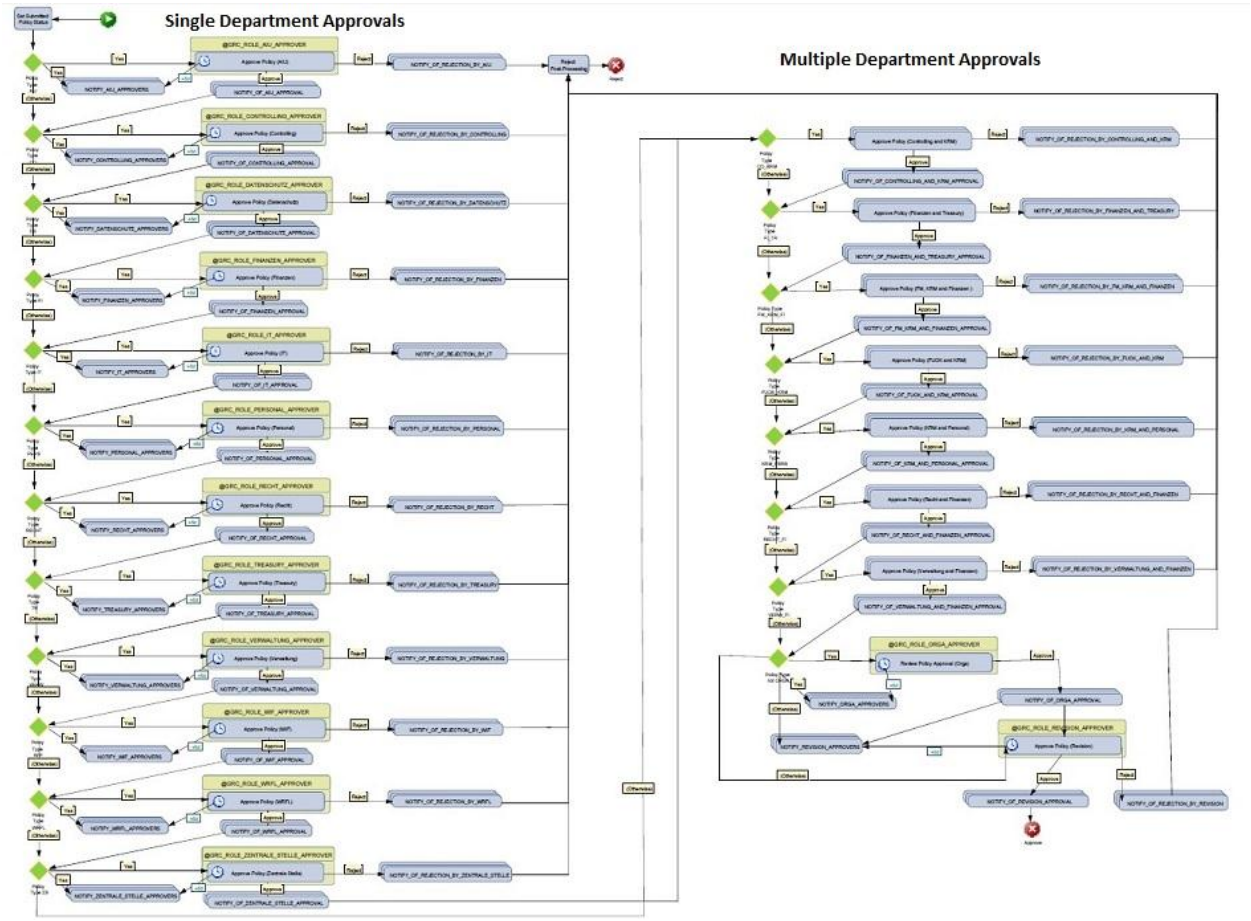


Figure 2. Model Approval Workflow – Multiple, Distinct Approval Paths

The focus of this integration example is to simplify the complex multi-approval workflow by integrating with business rules that are created by SAS Business Rules Manager. The business rules are invoked as a SOAP-based web service that is generated from a published SAS Stored Process to determine the number of approvals required and from which departments (Figure 3).

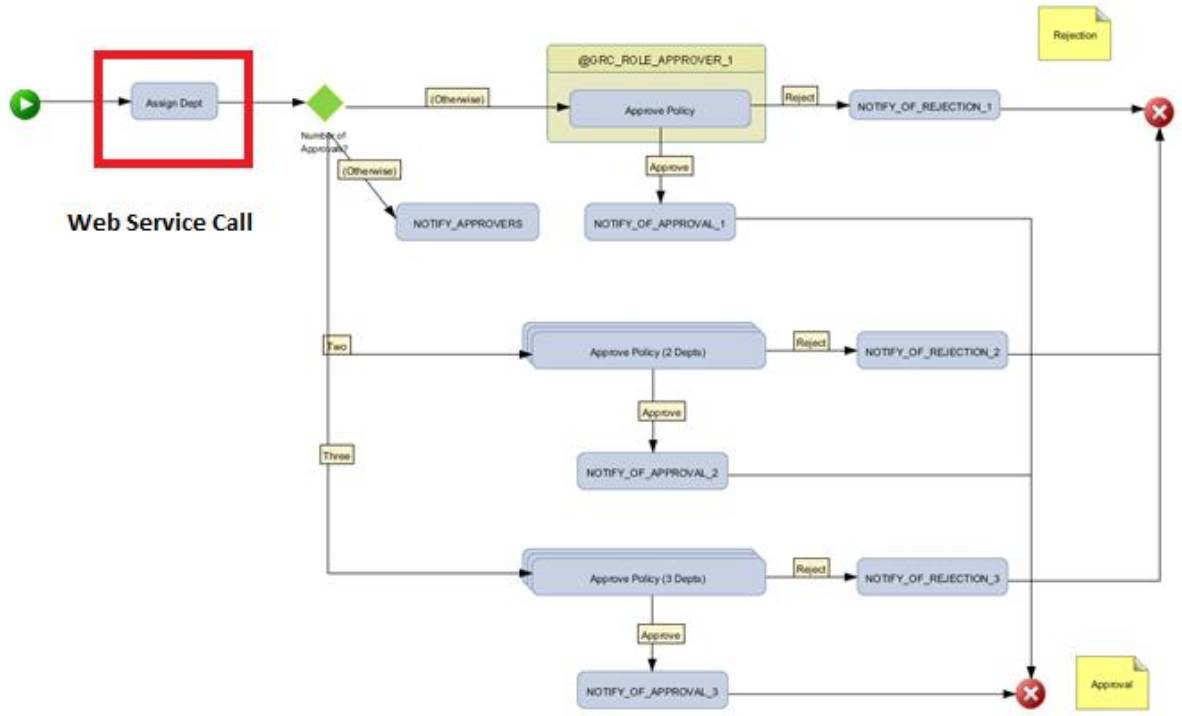


Figure 3. Revised Model Approval Workflow – Integration with Business Rules

IMPLEMENTATION ACTIVITIES

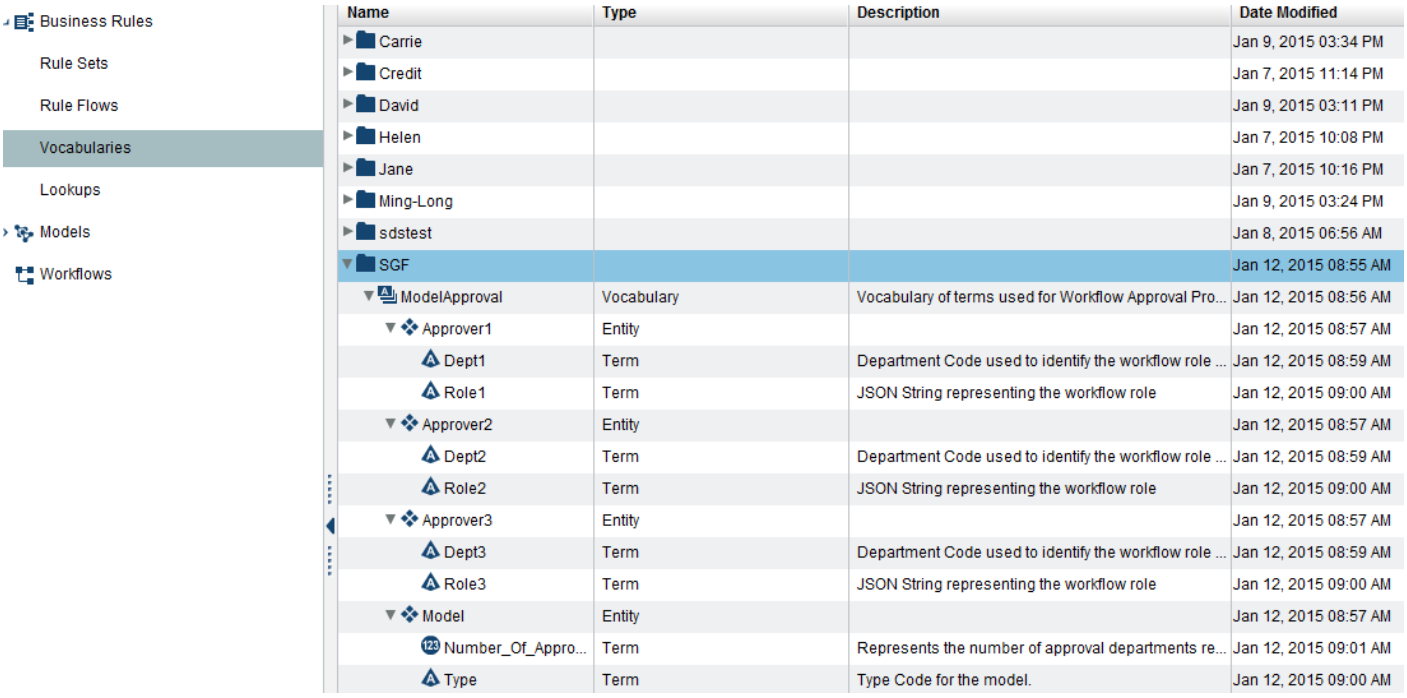
These are the implementation steps that are required to create and publish the rules for this business scenario:

1. Define a Vocabulary
2. Create Lookups - Making Rules Dynamic
3. Create Rules and Rule Sets
4. Create a Rule Flow
5. Rule Flow Testing – Rule Validation
6. Publish the Rule Flow
7. Integrate Generated Web Service Into the Workflow

These implementation activities not only accomplish the goal of simplifying the workflow within the business scenario, but also represent a repeatable implementation cycle that can be reproduced for other business process automation where SOAP-based web service integration is desirable.

DEFINE A VOCABULARY

The primary prerequisite for creating rules is having a vocabulary to use when authoring rules. A vocabulary is a set of variables using common naming that is understandable to any user who needs to change the rules in the future. The vocabulary also enables SAS Business Rules Manager to understand what rules are intra-compatible. Vocabularies can be created manually and generated from data. Figure 4 is an example vocabulary that was created manually within SAS Business Rules Manager for the Model Approval workflow business scenario.



Name	Type	Description	Date Modified
Carrie			Jan 9, 2015 03:34 PM
Credit			Jan 7, 2015 11:14 PM
David			Jan 9, 2015 03:11 PM
Helen			Jan 7, 2015 10:08 PM
Jane			Jan 7, 2015 10:16 PM
Ming-Long			Jan 9, 2015 03:24 PM
sdstest			Jan 8, 2015 06:56 AM
SGF			Jan 12, 2015 08:55 AM
ModelApproval	Vocabulary	Vocabulary of terms used for Workflow Approval Pro...	Jan 12, 2015 08:56 AM
Approver1	Entity		Jan 12, 2015 08:57 AM
Dept1	Term	Department Code used to identify the workflow role ...	Jan 12, 2015 08:59 AM
Role1	Term	JSON String representing the workflow role	Jan 12, 2015 09:00 AM
Approver2	Entity		Jan 12, 2015 08:57 AM
Dept2	Term	Department Code used to identify the workflow role ...	Jan 12, 2015 08:59 AM
Role2	Term	JSON String representing the workflow role	Jan 12, 2015 09:00 AM
Approver3	Entity		Jan 12, 2015 08:57 AM
Dept3	Term	Department Code used to identify the workflow role ...	Jan 12, 2015 08:59 AM
Role3	Term	JSON String representing the workflow role	Jan 12, 2015 09:00 AM
Model	Entity		Jan 12, 2015 08:57 AM
Number_Of_Appro...	Term	Represents the number of approval departments re...	Jan 12, 2015 09:01 AM
Type	Term	Type Code for the model.	Jan 12, 2015 09:00 AM

Figure 4. Model Approval Vocabulary

The vocabulary name is ModelApproval and it contains 4 entities (Approver1, Approver2, Approver3, and Model), which are logical groupings of business terms. Each entity contains 2 business terms. For example, the Model entity contains the business terms Type and Number_of_Approvers. The business terms have data types and can be mapped to data columns when executing batch processes or will be the named inputs and outputs to a generated web service that processes a single set of parameters. This enables business users to use terminology which is accessible to them, but still interchangeable with other applications.

CREATE LOOKUPS – MAKING RULES DYNAMIC

An optional construct that can be leveraged within SAS Business Rules Manager is lookups. A lookup is a dynamic set of name/value pairs that can have a new set of entries uploaded at any point and affect the already deployed rules, making them truly dynamic. Lookups are uploaded independently while being referenced directly in rule expressions, and they need to exist before they can be referenced within a rule. Unlike vocabulary and other constructs, a user does not have to leverage a lookup within rule logic.

In the Model Approver workflow that is being implemented, the determination of the number of approvers is encoded into a lookup to enable the user to upload new values when requirements specify that the number of approvers for a particular approval type should change. When the new lookup is uploaded, it will be leveraged within a generated stored process without having to change any integration, rules, or published artifacts. The following steps were performed to add Lookup references to the example.

Create the Lookup

1. Create a CSV (comma-separated values) file with the first column as the key/name that will be searched on when doing a lookup, and the second column as the value column that can be retrieved for that key/name. A user could generate this content from an existing data table or the content can be manually entered. Table 1 is a sample set of rows from a manually entered Lookup, which maps approval type codes for a workflow process to the number of approvers that are necessary if that status code is returned. **NOTE:** Lookup CSV files should not have a header row. Every row should be data.

AIU	1
CO	1
CO_KRM	2
DS	1
FI	1
FI_TR	2
FM	1

Table 1. Example Lookup CSV Content

2. Create a Lookup within SAS Business Rules Manager.
    a. Log on to the SAS Business Rules Manager web user interface.
    b. Navigate to **Rules** -> **Lookups**.
    c. In the folder view, right-click the desired target folder and select **Create Lookup**.
    d. Name the Lookup and then provide the name of the CSV file that you created. The CSV is assumed to be local to the computer that is running the web browser.
    e. Click **OK** to complete. The lookup is available for reference within rule set expressions.

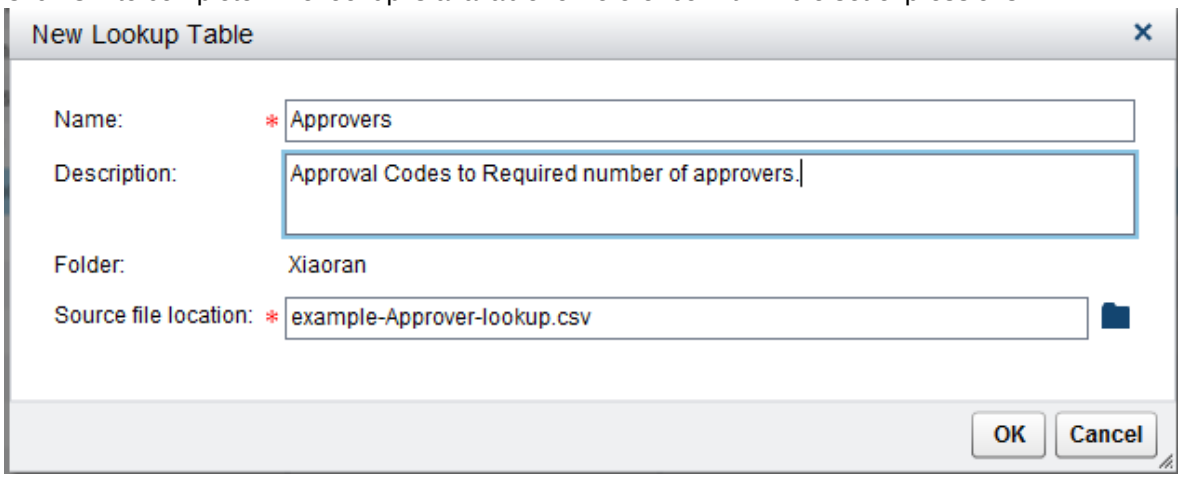


Figure 5. Lookup Creation Dialog Box

Using Lookup Functions

Reference a Reserve Lookup Function in a Rule

Now that the lookup is created, a user can access the content in the lookup using two unique reserve functions that are valid only in SAS Business Rules Manager.

1. Lookup('*LookupName*') - This function attempts to find the decision table column term value as a key within the lookup table. If the decision table term value exists within the lookup, the condition is evaluated to be true. This function is valid only within decision table conditions within a rule. In Figure 6, the decision table column term is Type, and its value is searched for within the 'Approvers' lookup keys. If found and other conditions are all true for that rule, then the rule fires.

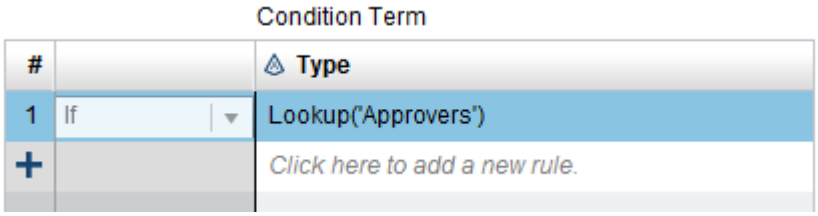


Figure 6. Condition Lookup Function Reference

2. LookupValue('*LookupName*', *key*) – This function attempts to find the passed in 'key' in the lookup referenced, then assigns the value to the decision table column term. If the key is not found, then no assignment is performed. This function is valid only within decision table actions within a rule. In Figure 7, the decision table column term is Number_Of_Approvers, the key that is looked up is the value of the Type term, and the lookup is named 'Approvers'.

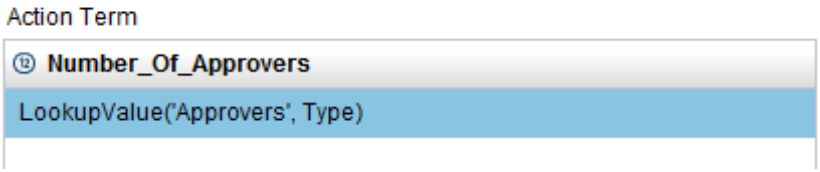


Figure 7. Action Lookup Function Reference

CREATE RULES AND RULE SETS

SAS Business Rules Manager provides a familiar decision table for entering rules within the rule set construct, shown in Figure 8.

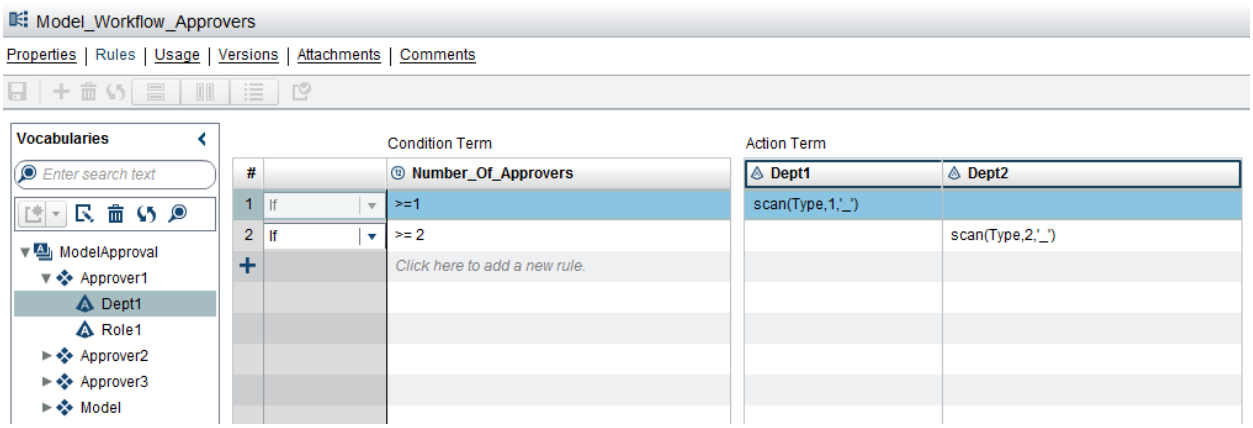


Figure 8. Rule Set Decision Table

Within a rule set, the business user can create a sequenced set of rules that can have conditions and then multiple actions. Each action is considered a variable assignment. Here, they can conditionally set values that can trigger business process changes following rule execution based on output values.

For the Model Approval workflow, the Dept1-3, Role1-3, and Number_of_Approvers variables are populated by the rules for the workflow to more concisely route the model through the required approval process.

CREATE A RULE FLOW

After the vocabulary is created, any necessary lookups are added, and rule sets are authored, a rule flow, which is an ordered execution of several rule sets, can be created. Rule flows enable users to encapsulate their logic/requirements within rule sets for reuse across multiple rule flows and makes

contributions by multiple teams simpler by allowing independent editing. Rule flows are also the construct that can be published. There are multiple publish targets that facilitate batch execution or web service execution. This paper focuses on publishing to a SAS Stored Process for web service execution.

The example displayed in Figure 9 requires only a single rule flow to reduce the complexity of the workflow that it becomes intergrated with.

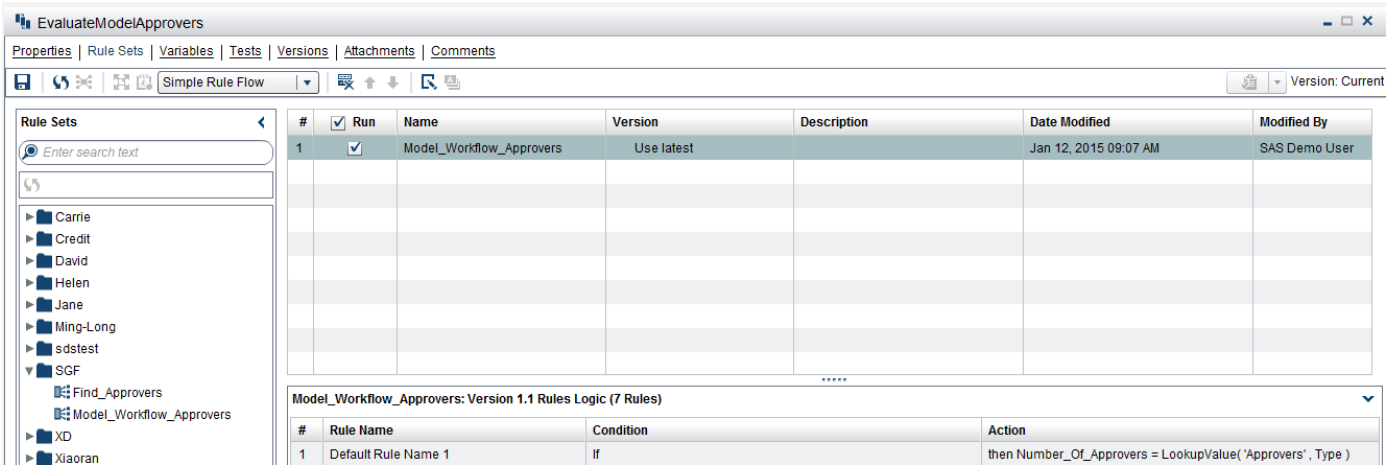


Figure 9. Rule Flow Editor

RULE FLOW TESTING – RULE VALIDATION

Each Rule Flow in SAS Business Rules Manager has a testing interface where a user can create multiple test cases against several input data sources and validate the output of their rule's execution. This enables a user to validate their rules before integration and cuts down greatly on communication overhead. The primary method of rule validation is through examining Rules Fired Analysis.

Rules Fired Analysis

Each time a rule's conditions are evaluated as true and trigger the actions to be executed or fired during testing, a rule fired tracking record is written. A detailed view is provided to the user so that they can see the number of rules that fired actions for each output record (Figure 10).

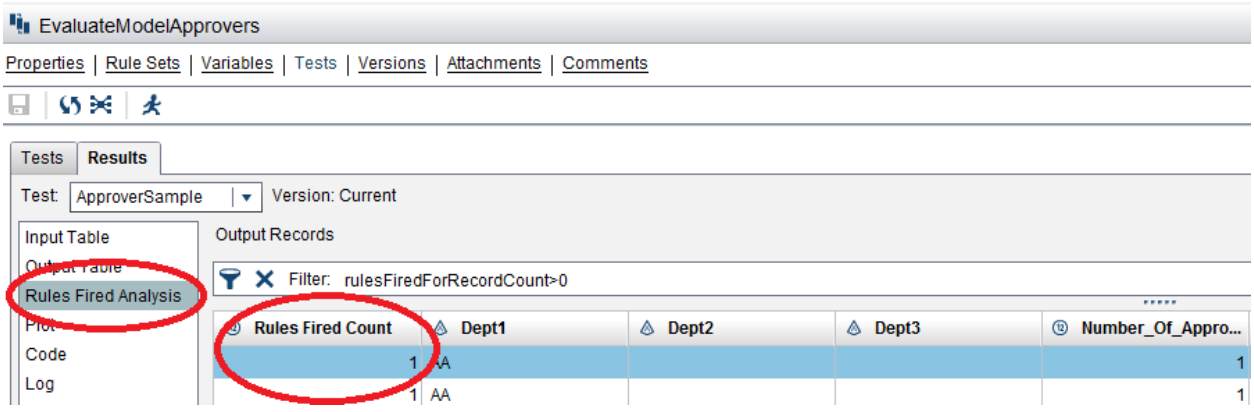


Figure 10. Rules Fired Analysis – Per Output Record Rule Fired Count

Moreover, by selecting the specific output record within the Rules Fired Analysis view, the business user can see the precise rules that fired to validate that the proper rules were executed (Figure 11). This enables the rule author who is creating the Model Approval rules to precisely validate that the expected output data was generated and that the correct rules executed to create that data, as well, before integration into the workflow.

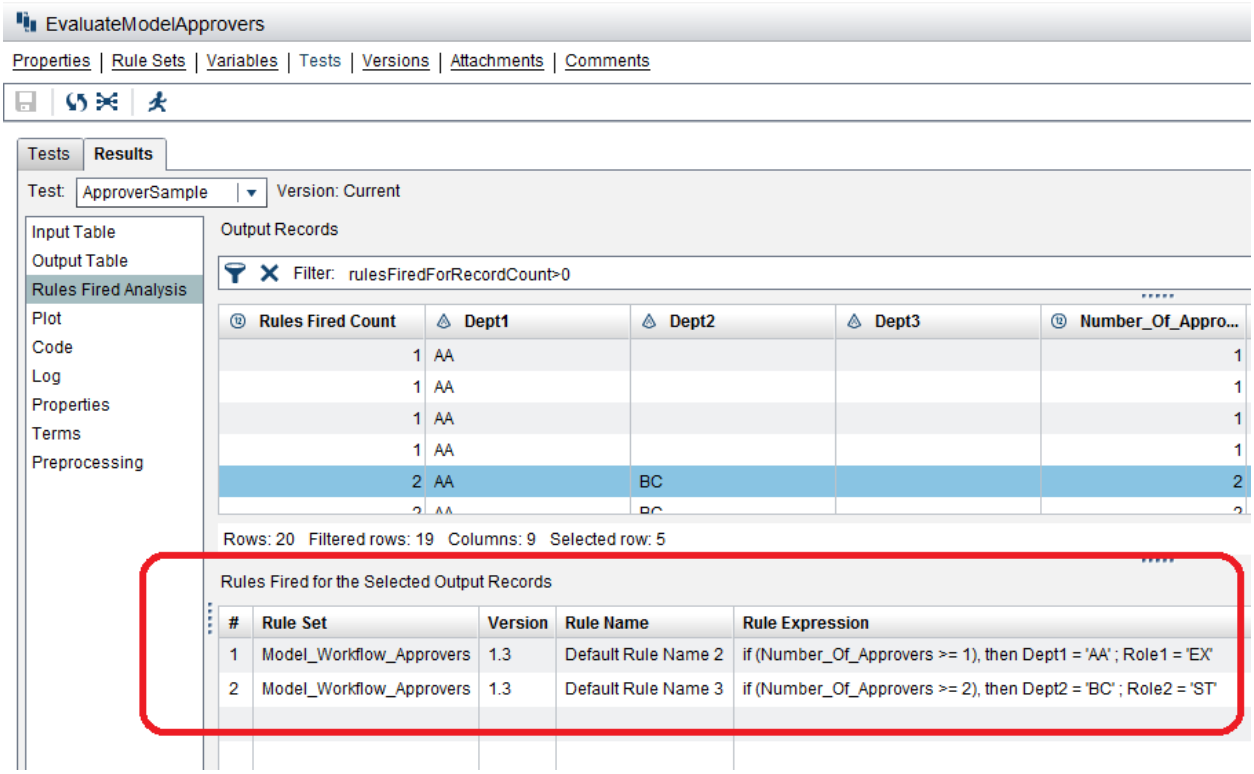


Figure 11. Rules Fired Analysis – Per Output Record Rule Fired Detail Panel

PUBLISHING THE RULE FLOW

After a rule flow is created and tested, the user needs to publish to make the rules available for integration. Within SAS Business Rules Manager, there are two publishing options for users. Both can be done with the single push of a button. However, both options lend themselves to different integrations.

1. Publish as a stored process
    a. Customized toward execution as a SOAP-based web service.
    b. Becomes immediately available for execution with a single set of input and output variables.
    c. Accessible via any stored process accessible technology, such as SOAP-based web service, PROC STP, and SAS Stored Process Web Application.
    d. Editable code stored within the stored process metadata object.
2. Publish as a BusinessRuleFlow metadata object
    a. Requires further integration in ETL tooling.
    b. Stores unmodifiable versions of the generated rule code.
    c. Batch execution available via transform in SAS Data Integration Studio.

Why Stored Process?

The example of integrating into the Model Approval workflow has several characteristics that a user can look for to understand that a stored process should be considered as an option for publishing a Rule Flow.

1. Single Transaction Processing - Each execution of the workflow only needs to process a single transaction record to properly determine how to route the approval process. This is true for many operational business processes.

2. SOAP-based web services support – The integrating application, SAS Workflow, fully supports integrating with SOAP-based web services from its user interface with minimal coding. Because

SOAP is a mature technology, many platforms have similar support for SOAP-based web services.

3. No XML structure requirements - There are no predefined XML structure requirements for how SAS Workflow creates or processes requests. Some platforms might carry legacy requirements on how they exchange data for web services. The generated stored process from SAS Business Rules Manager has limited ability to change its input and output structure without significant customizations.

Creating a Stored Process from a Rule Flow

To publish as a stored process, click the highlighted button in Figure 12.

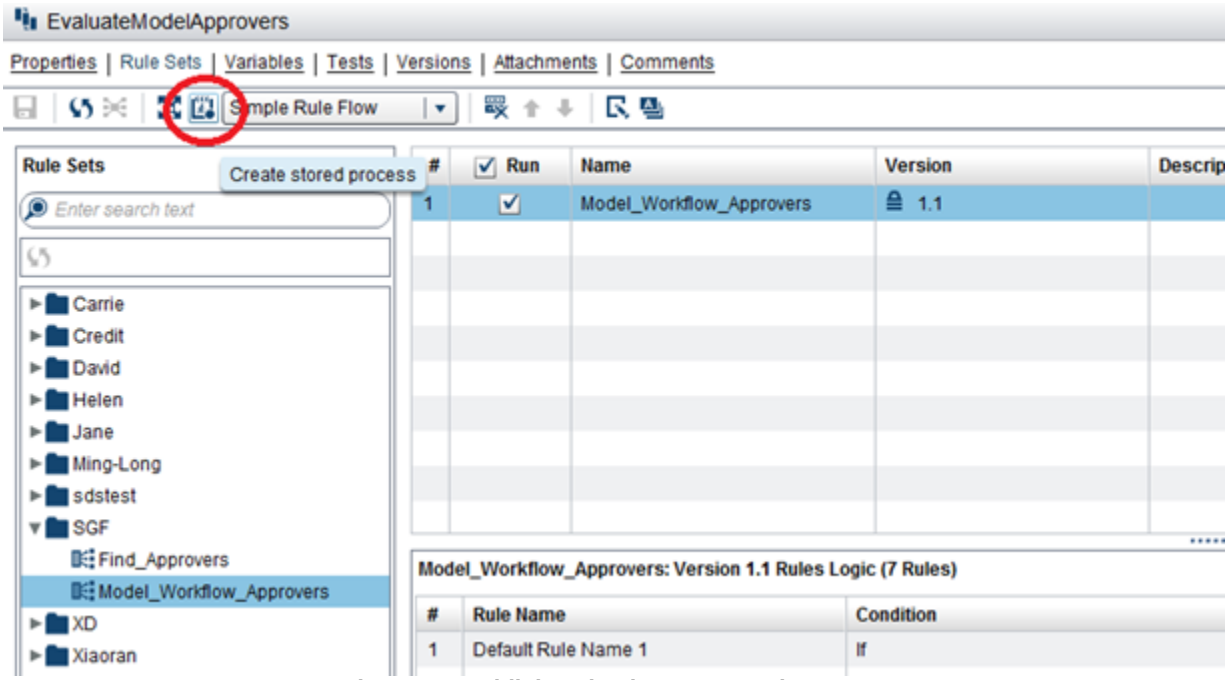


Figure 12. Publish Rule Flow as Stored Process

After clicking the button, the user needs to select a location for the stored process in the SAS metadata folders. The stored process is created with a registered set of input and output parameters based on the rule flow's inputs and outputs. The inputs compose one logical input record for the rules to process so that the stored process can be accessed via a SAS Business Intelligence Web Service endpoint immediately, or the process can serve as a template for a technical user in which they can manipulate the code as needed to handle more advanced input and output scenarios.

In previous releases of SAS, the options for creating a stored process for a rule flow were either to manually code a stored process or engineer a SAS Data Integration Studio job that is compatible with SAS Business Intelligence Web Service execution. These options are still available for more advanced applications. However, the new steps that are available as of SAS Business Rules Manager 2.2 are greatly simplified.

INTEGRATING WITH THE WEB SERVICE

The final step to complete the Model Approval workflow example is integrating the published stored process into the workflow via SAS Workflow. To do this, the user must understand the structure of the SOAP messages for the stored process and how to locate the web service endpoint. These two aspects of the web service are common for any integration platform, not just SAS Workflow.

Locating the Web Service

For this integration, the SAS Business Rules Manager user has published a stored process to the following location in SAS metadata:

/Products/SAS Business Rules Manager/EvaluateModelApprovers_Current_sp

To find the web service endpoint, apply the convention for stored processes and SAS Business Intelligence Web Services, where the default URL for the web service includes the SAS metadata path. Table 2 shows an example URL.

SAS Metadata Location of Stored Process	/Products/SAS Business Rules Manager/EvaluateModelApprovers_Current_sp
BIWS SOAP endpoint	http://${sasmidtierhost:port}/SASBIWS/services/Products/SAS+Business+Rules+Manager/EvaluateModelApprovers_Current_sp

Table 2. Business Intelligence Web Service SOAP Endpoint Formulation

The only change necessary is to URL encode any special characters. In the previous example, the spaces were turned into plus signs (+) to make the URL compatible with all standard SOAP clients.

Security

A prerequisite to successfully using the URL is understanding the security that will be enabled on that endpoint. By default, the standard SAS authentication is the only security applied to the endpoint. See SAS Intelligence Platform: Web Application Adminstration Guide for information about customizations to further secure the endpoint. Within this Model Approval workflow example, standard SAS authentication was leveraged.

Some sites might require NO authentication or anonymous access to the web service, as well, in which case enabling the SAS Anonymous user would be necessary for that installation.

Structure of the SOAP

The structure of the SOAP is dependent on the actual inputs and outputs to the Rule Flow that was published. The example stored process takes in a type field that determines how many approvers are required for the workflow, along with the required departments and roles those approvals need to come from. Here is a listing of the inputs and outputs that were determined by SAS Business Rules Manager for the stored process, based on the logic that was provided by the user:

- Inputs
  - Type
- Outputs
  - Dept1 – Department that the first approval must come from.
  - Dept2 – Department that the second approval must come from.
  - Dept3 – Department that the third approval must come from.
  - Number_of_approvers – Total number of approvals needed for the workflow.
  - Role1 – Required Role of the user that the first approval must come from.
  - Role2 – Required Role of the user that the second approval must come from.
  - Role3 – Required Role of the user that the third approval must come from.
  - Type – The original input is included within the output for verification purposes

With the endpoint URL, the required structure of the SOAP envelope can be discovered by accessing the endpoint with “?WSDL” appended to it, as follows:

http://${sasmidtierhost:port}/SASBIWS/services/Products/SAS+Business+Rules+Manager/EvaluateModelApprovers_Current_sp?WSDL

When it is retrieved, the SAS Business Intelligence Web Service middle tier dynamically generates a standard WSDL that includes the current input and output parameters that are listed for the stored process, which can be used to build a request or parse a response.

If the integrating application does not provide built-in SOAP process support as SAS Workflow, which is being used in this example, custom code needs to be written to make a request and parse the response. Many APIs exist that can process the WSDL and provide a programmatic interface to generate correct syntax for a request and process responses. Apache Axis and JAX-WS are two examples.

Formatted Input and Output Values

By default, formats are not conveyed within the structure of the stored process. This is because when SAS Business Rules Manager generates a stored process, it normalizes input and output data types to either numeric or character to simplify the stored process. If the integration requires formatted inputs such as dates or date times, the following two options can be used:

1. Within the rules, create actions that take a formatted vocabulary term and convert it into a numeric data type for use in arithmetic comparisons and other operations. Figure 13 shows an example of doing this conversion of a date using the 'inputn' function.

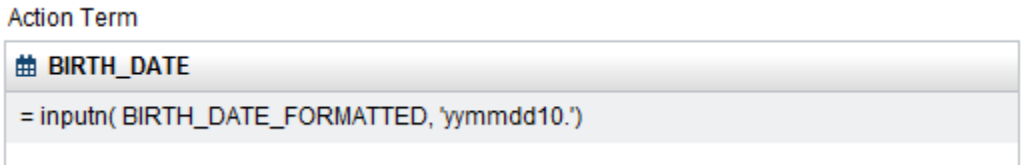


Figure 13. Example Input Formatting in Rule

2. Update the stored process symbol table retrieval within the generated code. To pass input and output parameters, the stored process server uses the macro symbol table. Using SAS Management Console, a user can navigate to the stored process and use the Properties dialog box to edit the code. The first and last DATA steps within that code set the inputs and outputs, respectively, for the symbol table. You can change these steps to format using standard SAS functions. The following code is an example of input date formatting:

```
data work.TEMPLIB;
   BIRTH_DATE=input(symget('BIRTH_DATE'),yymmdd10.);
   output;
run;
```

NOTE: Make sure that the applicable stored process parameter has the correct data type based on the value that is being formatted within the stored process Properties dialog box in SAS Management Console. In the previous example, the BIRTH_DATE field must be changed from numeric type to character.

Using PROC STP to Understand SOAP Structure

A simple SAS built-in way to validate the understanding of the SOAP structure that is required is to use PROC SOAP to send example requests to validate the request structure, then examine the response structure.

The following code is an example request to the EvaluateModelApprovers_Current_sp stored process, which is submitted via PROC SOAP:

```
FILENAME REQUEST 'C:\REQUEST.xml';

/* Build the example request within a file. */
data _null_;
   file request;
   input;
```

```
   put _infile_;
   datalines4;
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:biw="http://www.sas.com/xml/namespace/biwebservices">
   <soapenv:Header/>
   <soapenv:Body>
      <biw:evaluateModelApprovers_Current_sp>
         <biw:parameters>
            <biw:Type>AIU</biw:Type>
         </biw:parameters>
      </biw:evaluateModelApprovers_Current_sp>
   </soapenv:Body>
</soapenv:Envelope>
;;;;
run;

/* Submit the example request to the endpoint based on the standard
endpoint for the stored process in SAS Business Intelligence Web Services.
Below the wssusername and wsspassword options submit the necessary
information for standard SAS authentication. */
FILENAME RESPONSE 'C:\RESPONSE.xml';
proc soap in=REQUEST
          out=RESPONSE
url="http://XXXXXXX/SASBIWS/services/Products/SAS+Business+Rules+Manager/Ev
aluateModelApprovers_Current_sp"
          wssusername="XXXXX"
          wsspassword="XXXXX";
run;
```

Looking at the generated output file (RESPONSE.txt) enables an integrator to validate the web service availability and format XPath expressions, XML maps, or other strategies to extract output variables, if required.

Integrating with SAS Workflow

The SOAP-based web service is now ready to be integrated into SAS Workflow. Using the SAS Workflow Studio application, the integrator can add SOAP web service calls into the workflow and then make the updated logic available in SAS Workflow. The following information is required to add the SOAP web service call, and is displayed in Figure 14. This information is similar to many applications that provide user interface driven SOAP integration:

1. The location of the SOAP endpoint
2. Action Namespace
3. Input Content
4. Output Location
5. Error Handling
6. Authentication

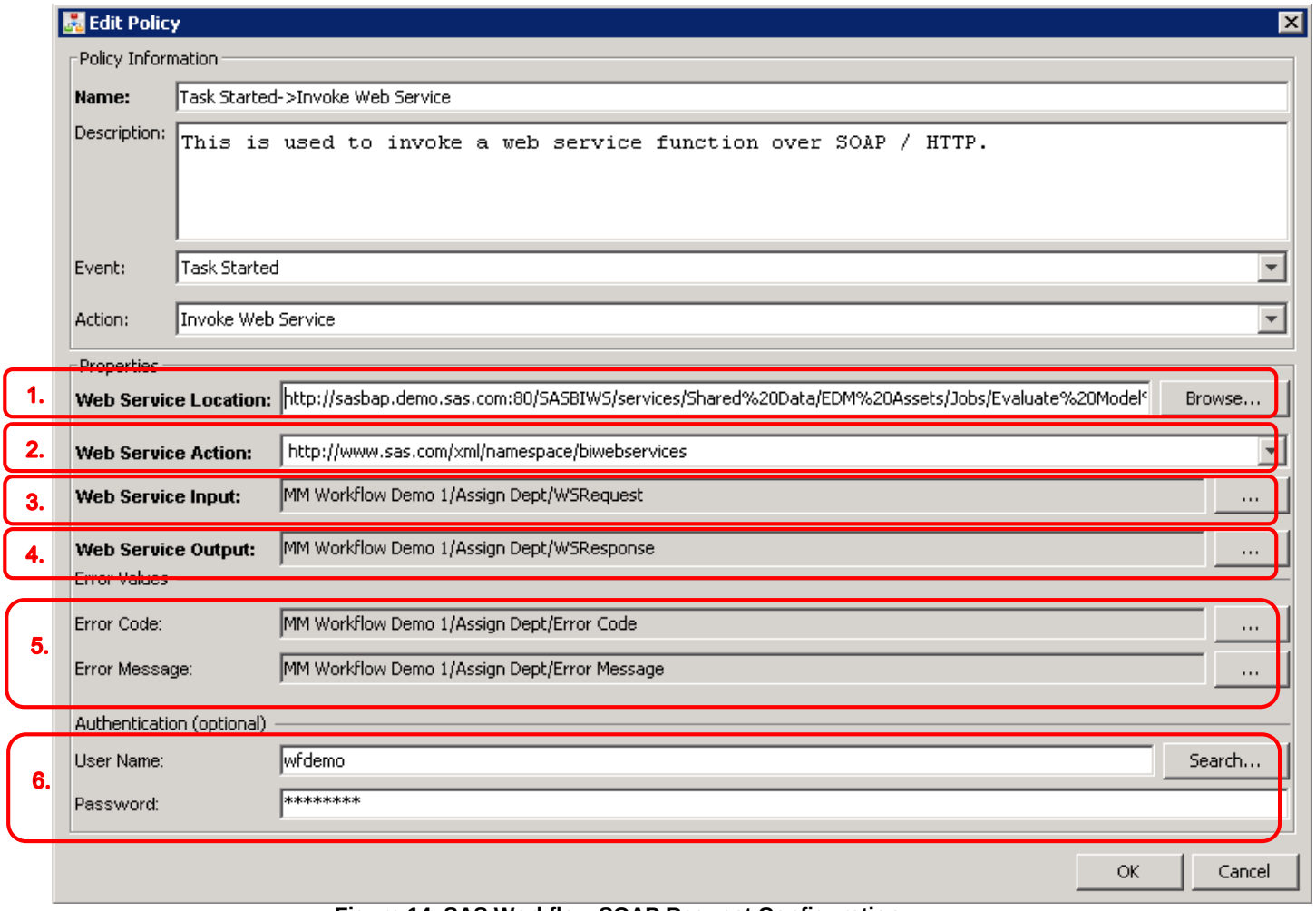


Figure 14. SAS Workflow SOAP Request Configuration

Because SOAP is ultimately an XML document that conforms to additional standards, SAS Workflow treats the response as XML data and extraction of output variables is done with XPath Statements to fetch only the necessary information (Figure 15). This could be repeated in any integration platform that supports XML and XPath processing, not just SAS Workflow.

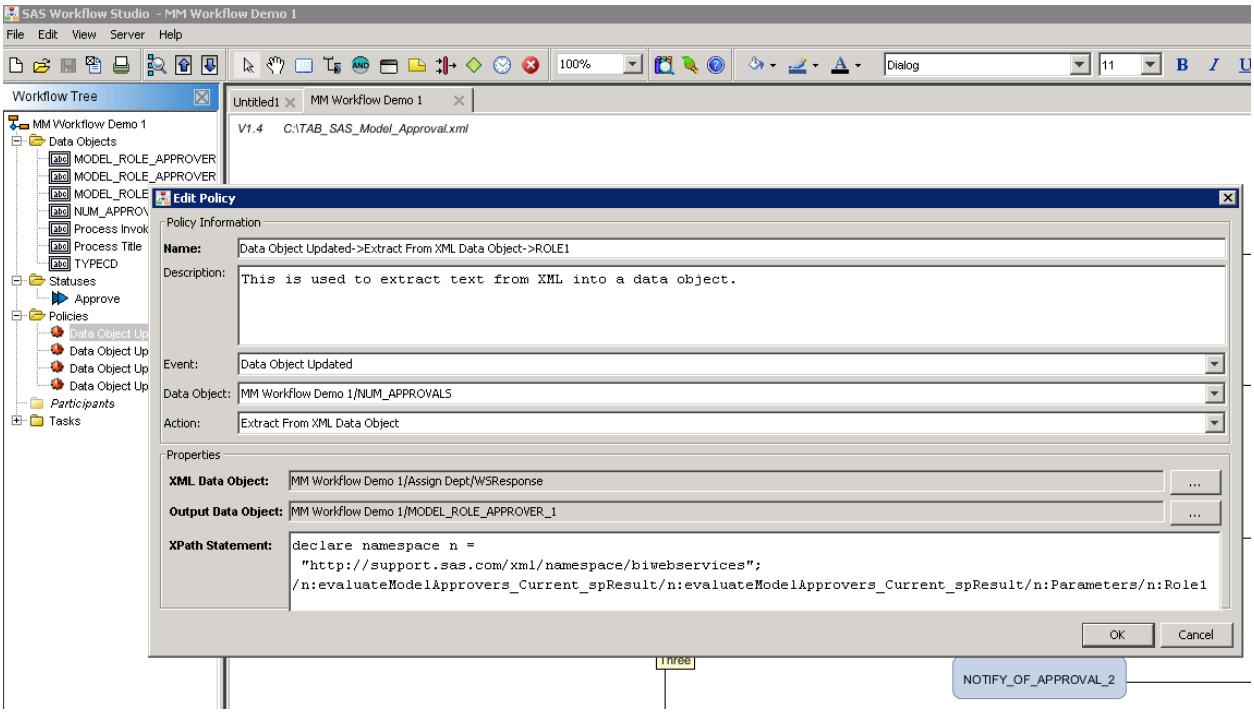


Figure 15. SAS Workflow Policy Editing – Data Extraction

Executing the Rules

After extracting all output variables, the workflow can be saved and uploaded for use by SAS Model Manager. The user can execute the rules by opening a model within the web user interface of SAS Model Manager and executing the workflow (Figure 16).

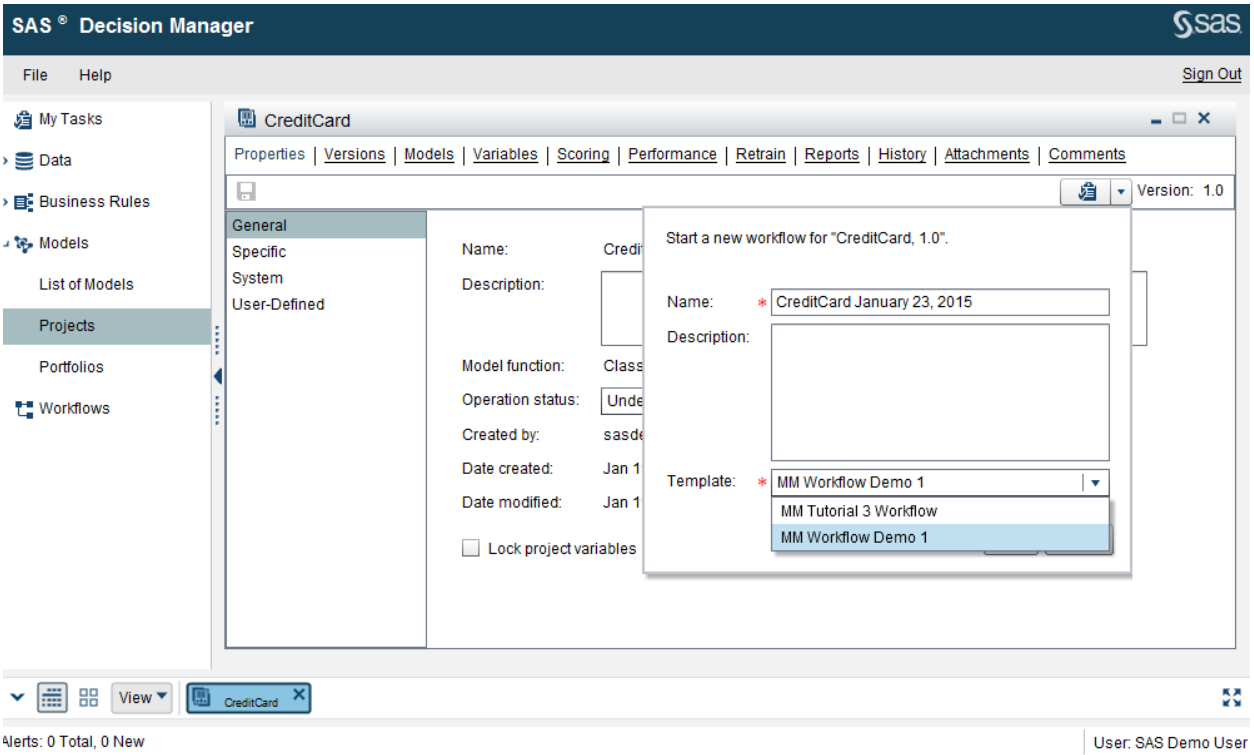


Figure 16. SAS Model Manager Workflow Execution within SAS Decision Manager

When executed, the number of approvers, departments of the approvers, and roles of the approvers are all controlled by a dynamic set of business rules that can update the existing business process logic. No technical user intervention is needed. Updates are made through versioning and republishing the underlying stored processes.

CONCLUSION

This scenario demonstrates how you can achieve operational agility by publishing your important business rules as stored processes to enable integration as a SOAP-based web service. SAS Business Rules Manager provides the features that businesses need as they look for easier and faster ways to deploy their rule logic to their business processes. These same businesses also know that business analysts and IT need to collaborate effectively to maximize business value when implementing their decisions.

By offering an integrated suite for business analysts and IT, SAS Business Rules Manager streamlines the analytical process by supporting these roles working together to deliver decisions to the operational side of the business faster and more efficiently, thereby reducing costs and reducing the lag between the time a business can react to their changing environments and customer demands.

REFERENCES

Sparano, Steve, Charlotte Crain, and Dave 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.

ACKNOWLEDGMENTS

The authors of this paper would like to thank the members of the various teams in R&D, Product Management, Marketing, and of course, our customers.

RECOMMENDED READING

- *SAS Business Rules Manager 2.2: User's Guide*
- *SAS Data Integration Studio 9.4: User's Guide*
- *SAS Model Manager 13.1: User's Guide*
- *SAS Workflow Studio 1.3: User Guide*

CONTACT INFORMATION

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

Chris Upton
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
Chris.Upton@sas.com
http://www.sas.com