Implementing a Business Rules Approach Through PowerCenter Parameters

Abstract

A business rule represents business decisions in an organization arrived at because of policies and requirements. Organizations use a Business Rules Management System (BRMS) to define, manage, deploy, execute, and monitor the business rules. You can use mapping parameters and variables within PowerCenter to simulate a Business Rules Management System and make mappings more flexible.

Business Rules

A business rule represents business decisions in an organization or enterprise arrived at because of policies and requirements. Typically, a business rule appears in the form of conditional statements that are used to perform a specific business task.

For example, to align with the new market regulations, a bank wants to deny a loan to a customer if the probability of default is greater than 0.5. This decision can be represented in the form of a business rule as follows:

> IF the probability of customer default is greater than 0.5 THEN deny loan.

Business Rules Approach

Today, processes and applications are built with complex architectures. Applications that are built with tightly coupled application and business logic are difficult to maintain. We cannot deny the fact that we live in a world of changes. It can take a long time to implement a simple change in the business logic and redeploy the application.To respond quickly to new business requirements, an enterprise must be flexible to apply business logic and swift to redeploy applications. This helps an enterprise gain a competitive advantage and abide by compliance.

When you define and maintain business logic outside the application in the form of business rules, you can build applications that are more flexible and ready for changes. In other words, business rules approach creates business agility.

A business rules approach has the following advantages:

- Lowers the cost incurred to modify business logic
- Shortens change cycles
- Promotes sharing and reuse among multiple applications
- Creates more business through the quick response to changes
- Helps meet business and legal regulations

PowerCenter as a Business Rules Management System

A Business Rules Management System is a system used to define, manage, deploy, execute, and monitor the business rules. This allows an enterprise to create rules using a business vocabulary so business users can understand the rules and make changes without depending on IT. You can configure PowerCenter to simulate a Business Rules Management System.

Mapping Parameters and Parameter Files

Use mapping parameters, mapping variables, and parameter files to set up business rules that are volatile in nature and change frequently. Mapping parameters and variables make mappings more flexible. They represent values that can be altered between workflow runs or during the course of a workflow run. This can reduce the overhead of creating multiple mappings when only certain attributes of a mapping need to be changed. This helps you reuse the existing mappings. A parameter file contains a list of parameters and variables with assigned values. This example uses mapping parameters.

When you run a session or workflow, the Integration Service evaluates references to the parameter based on the value defined in the parameter file. You can include an expression in the parameter file and configure the mapping so the Integration Service evaluates the port containing the parameter before it evaluates the port that uses the parameter value.

In the parameter file, you can define expressions that represent business rules so you do not have to change the mappings every time the business rules change. When the business rules do change, you can simply update the value of the mapping parameter in the parameter file. You do not need to update the mapping.

Using IsExpVar

If you want to use expressions in the parameter file, you must enable IsExprVar in the mapping parameter. This property determines how the Integration Service expands the parameter in an expression string. The Integration Service evaluates the business logic in the form of an expression string and assigns the value to the parameter.

The Integration Service expands mapping parameters and variables at run time based on how you configure IsExprVar. When you set IsExprVar to TRUE, the Integration Service evaluates the parameter before it validates the expression. It calculates the mapping parameter value with each row.

Use the following guidelines when you configure the mapping parameter:

- You must set the datatype of the parameter to String, or the Integration Service will fail the session.
- Verify that the precision of the parameter is large enough to accommodate the expression you configure in the parameter file.
- If you set IsExprVar to FALSE, the Integration Service evaluates the parameter after it validates the expression. As a result, all references to the parameter will use the same parameter value.

Note: Default value for IsExpVar is FALSE.

Make the Business Rules Shared and Reusable

Sometimes the business logic is generic so the rule can be reused or shared by other workflows. You group parameters and variables in different sections of the parameter file. Each section is preceded by a heading that identifies the Integration Service, Integration Service process, workflow, worklet, or session to which you want to define parameters or variables.

It is good to keep not only decision logic, but also computational formulae in the parameter file. Use a parameter file to store computational formulae for calculations such as salaries, taxes, and credit risk scores. These types of formulae can frequently change in terms of arithmetic constants, and you can simply change the constant values in the parameter file.

What to Include or Exclude in Business Rules

When you implement the business rules approach with PowerCenter, use parameters and parameter files to set up business rules that are volatile in nature and change frequently. This can include decision statements that drive business policies and default value declarations and calculations.

Transformation logic that is technical and non-volatile in nature can be handled inside the mappings. This can include aggregation, normalization and denormalization, sequence generation, datatype conversion, and data manipulation.

Sample Case

Many banks expect customers to maintain a minimum balance in their accounts on a monthly or quarterly basis. Failure to do so results in deduction of some amount from the existing account balance. The minimum balance to maintain and the amount to be deducted depend on business policies that can change at any time with market conditions.

A bank uses the following rule for the preceding policy:

IF account balance < 20,000 THEN deduct 500 from the account balance.

You can define a parameter in the Designer and define the rule for the parameter in the parameter file. When you run a session, the Integration Service expands the parameter based on the definition in the parameter file.

Set up the Business Rules as Parameters

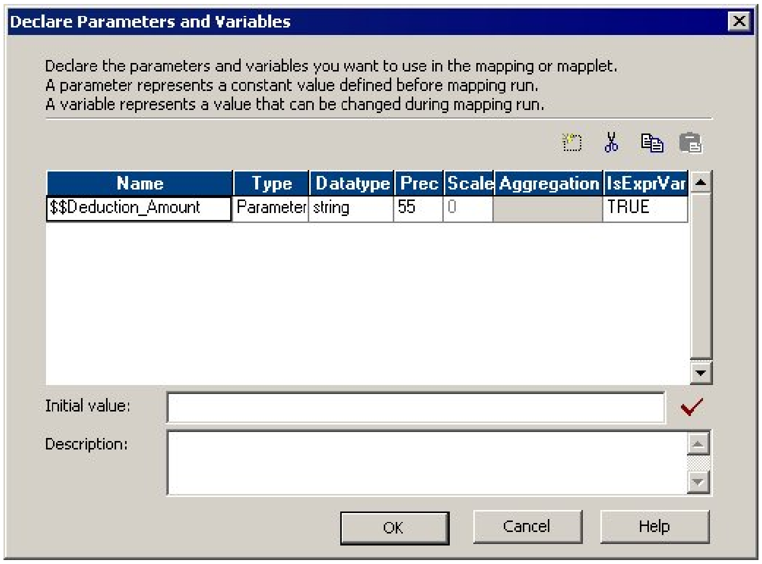
If you set up business rules in PowerCenter and declare a mapping parameter $$Deduction_Amount with IsExprVar set to TRUE, you can assign the business logic to this parameter in the parameter file.

You configure a mapping named m_Determine_Quarterly_Account_Balance to check the customer account balance. If it is less than $20,000, you thereby deduct $500 from the balance at the end of a quarter.

Assume that the bank revises the minimum balance policy and sets it to $10,000. If this mapping is complex, it would be a challenge to implement this simple change. At the end of the quarter, instead of modifying the mapping, you can simply change the business logic in the parameter file.

The Mapping Parameter

When you configure the mapping, define a mapping parameter to represent the business rule. The following figure shows the parameter that you define:



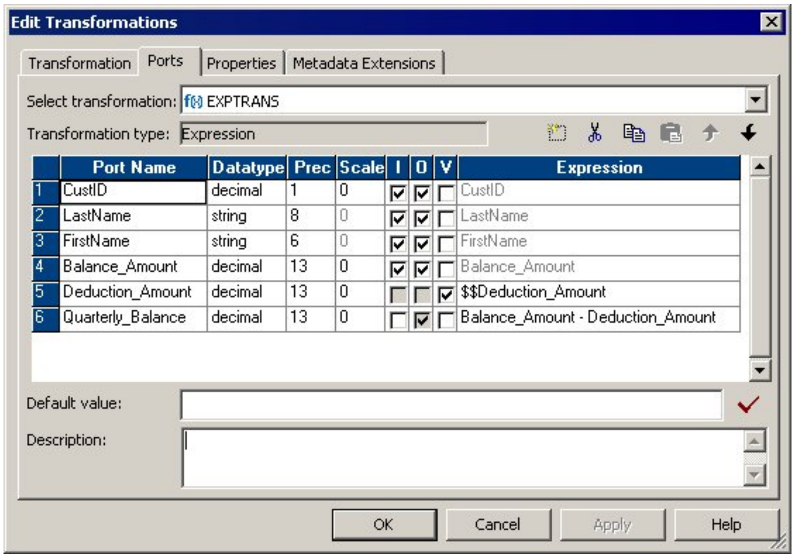
Declare the parameters and variables you want to use in the mapping or mapplet.
A parameter represents a constant value defined before mapping run.
A variable represents a value that can be changed during mapping run.

Name	Type	Datatype	Prec	Scale	Aggregation	IsExprVar
$$Deduction_Amount	Parameter	string	55	0		TRUE

The Expression Transformation

When you define the Expression transformation, call the mapping parameter before you use the value in another expression. Configure a variable port named Deduction_Amount, and use the mapping parameter $$Deduction_Amount as the expression. Configure an output port named Quarterly_Balance and configure an expression to subtract the results of Deduction_Amount from Balance_Amount.

The following figure shows the Expression transformation:



Select transformation: EXPTRANS

Transformation type: Expression

	Port Name	Datatype	Prec	Scale	I	O	V	Expression
1	CustID	decimal	1	0	☑	☑	□	CustID
2	LastName	string	8	0	☑	☑	□	LastName
3	FirstName	string	6	0	☑	☑	□	FirstName
4	Balance_Amount	decimal	13	0	☑	☑	□	Balance_Amount
5	Deduction_Amount	decimal	13	0	□	□	☑	$$Deduction_Amount
6	Quarterly_Balance	decimal	13	0	□	☑	□	Balance_Amount - Deduction_Amount

Default value:

Description:

Note: If you want to link Deduction_Amount to a downstream transformation, you can make it an output port instead of a variable port.

The Parameter File

You want this business rule to be reusable and apply to all workflows assigned to any Integration Service. To do so, configure the following entry in the parameter file:

```
[GLOBAL]
$$Deduction_Amount=IIF(Balance_Amount < 20000,500,0)
```

If you want the business rule to be reused for workflows assigned to a particular Integration Service, you can declare the parameter in a section called [Service:service name].

Author

Noor Basha Shaik,
Informatica Certified PowerCenter Consultant, Data Quality Specialist

Acknowledgements

The author would like to thank Srinivasan Venkataraman, Ashlee Brinan, Daphne Williams, and Bhanuprashanthi Murthy for their encouragement and guidance given during the course of writing this article.