Setting parameters using conditions

Instead of setting a single value for a parameter, you can instruct RBA to select one of several possible values after analyzing a set of expressions.

You can use this type of pattern matching in **Branch** and **Filter Items** flow actions and in Tables of Values to instruct the system to take a specific action when a given string value matches the specified pattern. For example, you might want a specific action to be taken when a job name ends in XYZ, a file name contains a a 6-digit number, or an imposition plan name does not end in a date.

You can use regular expressions to create sophisticated rules; expressions enable you to create much more compact rule sets in RBA.

In RBA, expressions and values are defined in a table with the following structure:

If this is true	Then "<parameter>" is
Expression 1	Value 1
Expression 2	Value 2
Otherwise	Value 3

RBA analyzes the first expression. If the expression is true, RBA uses the value in that row. If the expression is false, RBA analyzes the next row in the table. If RBA reaches the last row and none of the conditions are met, it uses the value in the **Otherwise** row.

The structure is similar to a Select Case statement in the Microsoft Visual Basic programming language.

Create tables for parameters in the Select Parameter Value dialog box, which appears when you click the **Create Table of Values** button in the Parameter Value dialog box.

Example

If you organize jobs into groups named after the operator, you can send an email to the appropriate operator each time a job is created in the operator's group. The table for the **To** parameter of the **Email** action would look something like this:

If this is true	Then "To" is
If Job > **Job Group** > **Name Contains Bob**	bob@printer.com
If Job > **Job Group** > **Name Contains Pete**	pete@printer.com
Otherwise	carol@printer.com

When a job is created, RBA looks at the job name. First, it checks for the letters "Bob". If there is a match, RBA sends an e-mail message to bob@printer.com. If there is no match, it checks for the letters "Pete". If there is a match, RBA sends an e-mail message to pete@printer.com. If there is no match, RBA has no more conditions to check, so it sends an e-mail message to carol@printer.com.

Separate tables for each parameter

The conditions that you create are specific to each parameter.
For example, if you want each operator to receive a slightly different message body, set up one table for the **To** parameter and one table for the **Body** parameter of the **Email** action.
This enables you to use different expressions for each parameter, if necessary. You can analyze different objects in each table or analyze the same objects against different conditions.

Combining conditions with process template overrides

You can combine a condition table with process template overrides to create powerful, short rules without branches. For example, you can use conditions to analyze the job name for the presence of specific characters and use the result to select an ICC profile that overrides the **Destination Profile** box in the process template.

For more information

Although an in-depth discussion of regular expressions is beyond the scope of this document, expressions are widely used in the computer industry and there are many books, articles, and Web sites that provide detailed information about writing and using expressions. Many Web sites also provide regular expression testers that you can use to enter an expression and test it against various target strings to see how well it matches the patterns you are looking for.