

Gabriel Magariño

Software Engineer

gabriel.magarino@gmail.com

www.javapassion.com/idm

Workflows - Variables

Disclaimer and Acknowledgments

- ▶ The contents here are created as a own personal endeavor and thus does not reflect any official stance of Sun Microsystems on any particular technology
-

Variables



IDM

▶ You can define several types of variables that are used within a process:

» Process variables

- ▶ Accessible to all activities. Process variables defined within a process display in the BPE tree view.

» Activity variables

- ▶ Defined within an activity. Activity variables are used solely within that activity.

» Action variables

- ▶ Defined within an action. Action variables are used solely by the action where they are defined.
-

Variables Types

The logo for IBM, consisting of the letters 'I D M' in a bold, sans-serif font, centered within a square frame. The frame is composed of multiple overlapping, slightly offset lines, creating a layered effect. The logo is positioned in the top right corner of the slide.

IDM

▶ Variables types include:

» String

- Simple string constant.

» List

- Static list, such as an XML object list. This type is infrequently used in workflows (but used occasionally in forms).

» Expression

- Complex expression.

» Reference

- Simple reference to a variable.
-

Variables Types

The logo for IBM, consisting of the letters 'I D M' in a bold, sans-serif font, centered within a square frame. The frame is composed of multiple overlapping, slightly offset lines, creating a layered effect. The logo is positioned in the top right corner of the slide.

▶ Variables types include:

» Rule

- Simple reference to a rule.

» Map

- Static map, such as an XML object map. Used rarely.

» Integer

- Integer constant. Can be specified as String; the BPE coerces the string into the correct type.

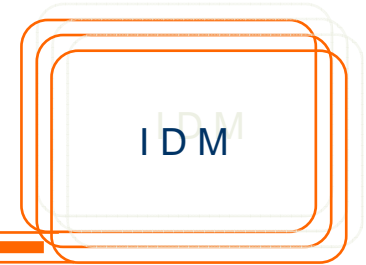
» Boolean

- Boolean constant. Can be specified as String; the BPE coerces the string into the correct type. Boolean values are specified with the strings “true” and “false.”

» XML Object

- Complex object that allows you to specify any of a number of complex objects with an XML representation. Some examples include EncryptedData, Date, Message, TimePeriod, and WavesetResult.
-

Variables



- ▶ The `<Variable>` tag is used to define workflow variables.
- ▶ Similar to a Java object implementation, where one or more object variables are declared for holding the “state” of the object, `<Variable>` tags declare data to be used by the containing workflow implementation.

```
<Variable name='TestVariable' input='true'>  
  <Comments>Account ID must be passed in</Comments>  
</Variable>
```

- ▶ In order to get the values of variable we need to use `<ref>` tag

```
<ref>TestVariable</ref>
```

Pass variables value as arguments

- ▶ You can also use the variable as an argument for any workflow activity.
- ▶ If the component that you are using is a string, you must include `$(your_variable)` syntax.
- ▶ Example

```
» <Activity id='0' name='Re-Provision'>
  <Action id='0'>
    <Argument name='TestVariable' value='$(TestVariable)'/>
    <Return from='applicationError' to='applicationError'/>
    <SubProcess>
      <ObjectRef type='Configuration' name='Sub Process 1'/>
    </SubProcess>
  </Action>
</Activity>
```

External Variables

- ▶ If a variable is declared as external in a sub-process, the sub-process may reference the variable defined in the parent process without requiring that it be explicitly passed as an argument in the sub-process call.

- ▶ Example

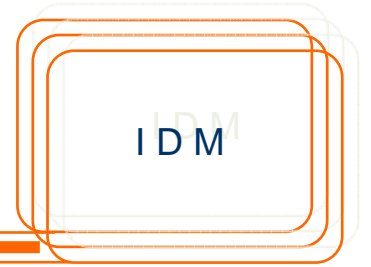
```
<Variable name='user' external='true'>  
  <Comments>The user view in the parent process</Comments>  
</Variable>
```

- ▶ e.g., If there is a <Field> declared in the Edit User Form as

```
<Field name='TestVariable'>  
  <Expansion>  
    <ref>MyTestValue</ref>  
  </Expansion>  
</Field>
```

- ▶ On Click on Save button a workflow “Update User” gets called in order to get the value we will declare the variable with same name and input =‘true’ as

```
<Variable name='TestVariable' input='true'/>
```



Workflow Built-in Variables

- ▶ **The workflow engine uses several built-in variables.**
 - ▶ **Most of these variables do not need to be declared in the workflow.**
 - ▶ **Built-in variables can be used to find out the state of the workflow execution.**
-

Workflow Built-in Variables...



Name	Description
WF_ACTION_ERROR	A built-in variable that will be set to true if the previously executed action returned a result containing an error or a thrown exception.
WF_ACTION_RESULT	A built-in variable that will be set to the WavesetResult object returned by the previous action. Use this variable when you want to capture the action's WavesetResult and process it without adding it to the global TaskInstance result.
WF_ACTION_SUPPRESSED	This built-in variable will be set to true if the action was suppressed due to a <Condition> expression evaluating to false.
WF_ACTION_TIMEOUT	A built-in variable that will be set to true if the previously executed action timed out.
WF_CASE_OWNER	A built-in variable that contains the name of the administrator that launched the workflow task.
WF_CASE_RESULT	A built-in variable that contains the WavesetResult of the TaskInstance. This can be used in Actions implemented in XPRESS or JavaScript to get a hold of the result. For Actions implemented as WorkflowApplication classes, they can obtain the result through the WorkflowContext. Since the entire WorkflowContext is exposed through the WF_CONTEXT variable, this is not absolutely necessary, but convenient.
WF_CONTEXT	A built-in variable that contains a WorkflowContext object. This can be used in Actions implemented in XPRESS or JavaScript to get a hold of the WorkflowContext. For Actions implemented as WorkflowApplication classes, the context is passed in.

Variables using XPRESS



IDM

▶ defvar Function

- » Defines a new variable. The variable can then be referenced by any expression within and below the containing expression in which the variable was defined.
- » The variable must be given a name using the XML attribute name.
- » A defvar statement should not reference itself. If it does, it will cause a loop.
- » *Example 1* - The following expression defines a variable and initializes its value to a list of two elements.

```
<defvar name='theList'>  
  <list>  
    <s>apples</s>  
    <s>oranges</s>  
  </list>  
</defvar>
```

- » *Example 2* - The following expression defines a variable and initializes its value to the integer zero.

```
<defvar name='counter'>  
  <i>0</i>  
</defvar>
```

Variables using XPRESS...



IDM

► ref Function

» References the value of a variable. The variable can either be an external variable supported by the host application or an internal variable defined with <defvar>.

» *Example 1*

```
<ref>waveset.role</ref>
```

» *Example 2*

```
<defvar name='milk'><s>milkvalue</s></defvar>
```

```
<defvar name='shake'><s>milk</s></defvar>
```

```
<ref><ref>shake</ref>
```

```
evaluates to <s>milkshake</s>
```

Gabriel Magariño

Software Engineer

gabriel.magarino@gmail.com

www.javapassion.com/idm

Workflows - Variables