

Understanding the Business Rules Method Palette

Copyright © 2008, 2011, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS. Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Understanding the Business Rules Method Palette	5
Java CAPS Business Rules Method Palette Overview	5
Operator Methods	6
String Methods	7
Number Methods	9
Boolean Methods	9
Node Methods	10
XSD Operation	11
Conversion	11
Encoder Parameter Values	11

Understanding the Business Rules Method Palette

The topics listed here provide reference information about each method that appears in the Method Palette of the Business Rule Designer . Many of these methods are based on standard XPath functions. Refer to the W3C XML Path Language documentation for additional information.

- [“Java CAPS Business Rules Method Palette Overview” on page 5](#)
- [“Operator Methods” on page 6](#)
- [“String Methods” on page 7](#)
- [“Number Methods” on page 9](#)
- [“Boolean Methods” on page 9](#)
- [“Node Methods” on page 10](#)
- [“XSD Operation” on page 11](#)
- [“Conversion” on page 11](#)
- [“Encoder Parameter Values” on page 11](#)

Java CAPS Business Rules Method Palette Overview

The Method Palette displays a menu of methods that you can incorporate into your process models to manipulate data in a variety of ways. These methods are grouped logically by the type of function performed. By default, not all methods appear in the Method Palette menus. To view complete lists of operators for each type, click Settings from any of the Method Palette menus. The Method Palette dialog appears, and you can select and deselect functions to indicate which functions should appear in each menu.

To display a method on the Business Rule Designer canvas, click the menu containing the method you want to use and then click the method name or symbol. A method box appears on the canvas, and you can then map the input and output values for the method.

For most of the methods, you can either drag an object from the attributes tree to the method box to enter a value into a parameter, or you can right click a parameter in the method box to add a literal value directly into the box.

Operator Methods

Operator methods allow you to manipulate data with standard mathematical and comparison operators. To display a complete list of available operator methods, click Operator and then click Settings.

TABLE 1 Operator Methods

Description
The addition method adds the numerical value of <i>number1</i> to the numerical value of <i>number2</i> , and returns the sum.
The AND method returns Boolean true if both <i>boolean1</i> and <i>boolean2</i> are true; otherwise, it returns Boolean false.
Note: Boolean parameters cannot accept values with capital letters. For example, “True” and “TRUE” are not valid Boolean values, but “true” is.
The div method divides the numerical value of <i>number1</i> by the numerical value of <i>number2</i> , and returns the quotient.
The EQUAL method returns Boolean true if <i>object1</i> is equal to <i>object2</i> ; otherwise, it returns Boolean false.
The greater or equal method returns Boolean true if the numerical value of <i>number1</i> is greater than or equal to the numerical value of <i>number2</i> ; otherwise, it returns Boolean false.
The greater than method returns Boolean true if the numerical value of <i>number1</i> is greater than the numerical value of <i>number2</i> ; otherwise, it returns Boolean false.
The lesser or equal method returns Boolean true if the numerical value of <i>number1</i> is less than or equal to the numerical value of <i>number2</i> ; otherwise, it returns Boolean false.
The lesser than method returns Boolean true if the numerical value of <i>number1</i> is less than the numerical value of <i>number2</i> ; otherwise, it returns Boolean false.
The mod method divides the numerical value of <i>number1</i> by the numerical value of <i>number2</i> , and returns only the remainder.
The multiplication method multiplies the numerical value of <i>number1</i> by the numerical value of <i>number2</i> , and returns the product.
The negative method converts the numerical value of <i>number1</i> to its negative. The result is a number having the same absolute value as the input number. For example, if the input is “10”, it is converted to “-10”; if the input is “-10”, it is converted to “10”.
The not equal method returns Boolean true if <i>object1</i> is not equal to <i>object2</i> ; otherwise, it returns Boolean false.
The OR method returns Boolean false if both <i>boolean1</i> and <i>boolean2</i> are false; otherwise, it returns Boolean true.
Note: Boolean parameters cannot accept values with capital letters. For example, “True” and “TRUE” are not valid Boolean values, but “true” is.

TABLE 1 Operator Methods (Continued)

Description
The subtraction method subtracts the numerical value of <i>number2</i> from the numerical value of <i>number1</i> , and returns the difference.

String Methods

The String methods allow you to manipulate string data. To display a complete list of available string methods, click String and then click Settings.

TABLE 2 String Methods

Description
The bytes to text method decodes a bytes object into text using the specified encoding. If no encoding is specified, the platform's default encoding is used. For a list of possible values for the <i>encoding</i> parameter, see “Encoder Parameter Values” on page 11.
The concat method returns the string created by concatenating the value of the <i>str</i> parameter to the end of the value of the <i>string</i> parameter.
Note: You can add multiple <i>str</i> parameters to the method. The method automatically adds an unmapped node as needed.
The contains method returns Boolean true if the value of <i>string1</i> is contained within the value of <i>string2</i> ; otherwise, it returns false
The convertUnicode method returns the equivalent string value to the Unicode value in the <i>string</i> parameter. The Unicode value can be hexadecimal, octal, or decimal in format. For example, the Unicode number for a new line character is: <ul style="list-style-type: none"> ▪ 0X00A in hexadecimal format ▪ 012 in octal format ▪ 10 in decimal format Use this method to insert other special characters, such as spaces, tabs, and so on.
The copy from method allows you to type in an XPath expression as the source of a copy operation. This is useful for entering XPath predicates. When you select copy from , the Copy From dialog box appears for you to enter the XPath.
Note: This feature is for advanced users who are familiar with XPath and BPEL syntax.
The copy to method allows you to type in an XPath expression as the destination of a copy operation. This is useful for entering XPath predicates. When you select copy to , the Copy To dialog box appears for you to enter the XPath.
Note: This feature is for advanced users who are familiar with XPath and BPEL syntax.
The normalize space method removes leading and trailing spaces from the string contained in <i>string1</i> .

TABLE 2 String Methods (Continued)

Description
<p>The starts with method returns Boolean true if the value of <i>string1</i> starts with the value of <i>string2</i>; otherwise, it returns false.</p>
<p>The string method converts the value of <i>object1</i> to a string.</p>
<p>The string length method returns the number of characters in <i>string1</i>.</p>
<p>The string literal method allows you to create a literal string; that is, a sequence of characters of fixed length and content. Type the literal value directly into the method box.</p>
<p>Tips:</p> <ul style="list-style-type: none"> ■ Use <i>Control+Enter</i> to insert a line break in a literal. ■ To add a string literal directly into a parameter without using the string literal method, right-click the parameter, and then click Add Literal. Type the literal directly into the parameter field. ■ To edit an existing literal from a method, right-click the literal, click Edit, and then modify the value. ■ To delete a literal from a method, right-click the literal, and then click Delete Literal.
<p>The text to bytes method encodes the input text into a sequence of bytes using the specified encoding. If no encoding is specified, the platform's default encoding is used. For a complete list of available values for the encoding parameter, see “Encoder Parameter Values” on page 11.</p>
<p>The substring method returns a substring of <i>string1</i>, with <i>number2</i> indicating the position of the first character in the substring and <i>number3</i> indicating the number of characters to include. <i>number2</i> represents the position of a character from the beginning of <i>string1</i>. If <i>number3</i> is empty, the return string includes the characters from the position indicated by <i>number2</i> to the end of <i>string1</i>.</p> <p>For example, if the parameters are (“Input Data”, 6, 3), the return string is “Dat”. If the third parameter is empty, the return string is “Data”.</p> <p>Note: The first character position is 1, not 0.</p>
<p>The substring after method returns the substring in <i>string1</i> that occurs after the first instance of the substring in <i>string2</i>.</p> <p>For example, if the parameters are (“Input Data”, “Input ”), the return string is “Data”. You would get the same return string if the second parameter contained only a space.</p> <p>Note: The first character position is 1, not 0.</p>
<p>The substring before method returns the substring in <i>string1</i> that occurs before the first instance of the substring in <i>string2</i>.</p> <p>For example, if the parameters are (“Input Data”, “ Data”), the return string is “Input”. You would get the same return string if the second parameter contained only a space.</p> <p>Note: The first character position is 1, not 0.</p>

TABLE 2 String Methods (Continued)

Description
<p>The translate method returns the value of <i>string1</i>, replacing any occurrences of the characters in <i>string2</i> with the characters in the corresponding positions in <i>string3</i>. If <i>string2</i> contains more than one instance of a character, only the first instance is taken into account. If <i>string3</i> is longer than <i>string2</i>, the extra characters in <i>string3</i> are ignored. If <i>string2</i> is longer than <i>string3</i>, the extra characters in <i>string2</i> are removed from <i>string1</i>.</p> <p>For example, if the parameters are (“Sun”, “un”, “MI”), the original string, Sun, is translated to SMI.</p>

Number Methods

The Number methods allow you to manipulate numeric data. To display a complete list of available number methods, click Number and then click Settings.

TABLE 3 Number Methods

Description
<p>The ceiling method returns the smallest integer that is not less than the value of <i>number1</i>.</p>
<p>The floor method returns the largest integer that is not greater than the value of <i>number1</i>.</p>
<p>The number method converts <i>object1</i> to a numerical value according to the following rules:</p> <ul style="list-style-type: none"> ▪ Boolean true is converted to 1; Boolean false is converted to 0. ▪ A string that includes whitespace at the beginning and end, a minus sign, and a number (where the whitespace and minus sign are optional) is converted to the closest IEEE 754 number. ▪ Strings that do not compute to a numerical value are converted to NaN. ▪ A set of nodes is converted to a string and then converted as a string (as described above).
<p>The number literal method allows you to create a number of fixed length and content. You can type the value directly into the method box.</p>
<p>The round method rounds the value of <i>number1</i> to the nearest integer.</p>
<p>The sum method returns the total value of the set of numeric values in <i>node-set1</i>.</p>

Boolean Methods

Boolean methods allow you to apply boolean logic to your data. To display a complete list of available Boolean methods, click Boolean and then click Settings.

TABLE 4 Boolean Methods

Description
<p>The boolean method converts the value of <i>object1</i> to Boolean, and returns true or false.</p>

TABLE 4 Boolean Methods (Continued)

Description
The exists method checks whether a value is present in <i>object1</i> . If a value is present, it returns Boolean true; otherwise, it returns Boolean false.
The false method always returns Boolean false.
The lang method returns Boolean true if the value of <i>string1</i> matches the language of the <code>xsl:lang</code> element; otherwise, it returns Boolean false.
The not method returns Boolean true if <i>boolean1</i> is false, and returns Boolean false if <i>boolean1</i> is true.
Note: Boolean parameters cannot accept values with capital letters. For example, “True” and “TRUE” are not valid Boolean values, but “true” is.
The true method always returns Boolean true.

Node Methods

Node methods allow you to manipulate node data. To display a complete list of available nodes methods, click Nodes and then click Settings.

TABLE 5 Nodes Methods

Description
The count method returns the number of nodes in the node-set contained in <i>node-set1</i> .
The get BPid method returns the business process instance ID.
The get current time method returns the current time in ISO 8601 format (for example, 2003-08-15T02:03:49.92Z).
The get GUID method returns a randomly generated globally unique ID.
The get local time method returns the local time in ISO 8601 format (for example, 2003-08-15T02:03:49.92Z).
The id method returns a node-set of elements selected by the unique ID specified in <i>object1</i> .
The last method returns the position number of the last node in the processed node list.
The local name method returns the local part of the first node in <i>node-set1</i> . A node usually consists of a prefix and a colon, followed by the local name.
The name method returns the name of the first node in <i>node-set1</i> .
The namespace url method returns the namespace URL of the first node in <i>node-set1</i> .
The position method returns the position in the node list of the node that is currently being processed

XSD Operation

The XSD Operation methods enable you to marshal and unmarshal messages. To display a complete list of available XSD operation methods, click XSDOperation and then click Settings.

TABLE 6 XSD Operation Methods

Description
The marshal method marshals the information in the OTD bean, and returns an XML document.
The unmarshal method unmarshals the information in an XML document, and returns an OTD bean.
The marshalToBytes method marshals the information in the OTD bean, and returns the data as bytes.
The unmarshalFromBytes method unmarshals the bytes input, and returns an OTD bean.

Conversion

There is one Conversion method, `convert`, that allows you to make conversions from various data types. To display the settings for the Conversion menu, click Conversion and then click Settings.

TABLE 7 Conversion Methods

Description
The convert method converts the input object to the output object. This method is only supported in version 5.0.X Projects.

Encoder Parameter Values

The text to bytes and bytes to text methods both take a parameter that specifies the type of encoding to use when performing the conversion. The possible values for the encoder parameter include the following:

- | | |
|---------------|---------------|
| ▪ Big5 | ▪ Big5-HKSCS |
| ▪ EUC-JP | ▪ EUC-KR |
| ▪ GB18030 | ▪ GBK |
| ▪ ISO-2022-JP | ▪ ISO-2022-KR |
| ▪ ISO-8859-1 | ▪ ISO-8859-13 |

-
- | | |
|-------------------|-----------------------|
| ▪ ISO-8859-15 | ▪ ISO-8859-2 |
| ▪ ISO-8859-3 | ▪ ISO-8859-4 |
| ▪ ISO-8859-5 | ▪ ISO-8859-6 |
| ▪ ISO-8859-7 | ▪ ISO-8859-8 |
| ▪ ISO-8859-9 | ▪ JIS_X0201 |
| ▪ JIS_X0212-1990 | ▪ KO18-R |
| ▪ Shift_JIS | ▪ TIS-620 |
| ▪ US-ASCII | ▪ UTF-16 |
| ▪ UTF-16BE | ▪ UTF-16LE |
| ▪ UTF-8 | ▪ windows-1250 |
| ▪ windows-1251 | ▪ windows-1252 |
| ▪ windows-1253 | ▪ windows-1254 |
| ▪ windows-1255 | ▪ windows-1256 |
| ▪ windows-1257 | ▪ windows-1258 |
| ▪ windows-31j | ▪ x-EUC-CN |
| ▪ x-euc-jp-linux | ▪ x-EUC-TW |
| ▪ x-eucJP-Open | ▪ x-ISC1191 |
| ▪ x-JIS0208 | ▪ x-JISAutoDetect |
| ▪ x-Johab | ▪ x-MS950-HKSCS |
| ▪ x-mswin-936 | ▪ x-windows-50220 |
| ▪ x-windows-50221 | ▪ x-windows-949 |
| ▪ x-windows-950 | ▪ x-windows-iso2022jp |
-