

# Oracle® Health Sciences Central Designer

## Rules Reference Guide



Release 7.0  
F56112-01



F56112-01

Copyright © 2019, 2023, Oracle and/or its affiliates.

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, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

Intel and Intel Inside 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, Epyc, and the AMD 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 about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

# Contents

## Preface

---

Documentation accessibility	x
Related resources	x
Diversity and Inclusion	x
Access to Oracle Support	x
Additional copyright information	xi

## 1 Rule expressions

---

About the rule expression language	1-1
Components of the rule expression language	1-2
Mappings in rules	1-4
Conversion to different units in rules	1-4
Rules with multiple conditions	1-5
Dynamic prompts in the Expression workspace	1-6
Selecting a rule model object from a dynamic expression prompt	1-8
Rule templates	1-8
Creating, modifying, and deleting a rule template?	1-9
Create a rule template	1-9
Modify a rule template	1-9
Delete a rule template	1-10
Other descriptions for the New Rule Template dialog box	1-10
Default rule templates	1-11
_CheckTextLength	1-12
_DateTimeRangeCheck	1-12
_FutureDateCheck	1-12
_PartialCompletenessCheck	1-13
_RangeCheck	1-13
_RangeCheckInclusive	1-14
ValidBPCheck	1-14

## 2 Data

---

Data tab	2-1
Using data in rules and rule templates	2-1
Icons used on the Data tab	2-2
Rule model properties for study events, forms, and sections	2-3
[Name of review stage]	2-4
[Name of review state]	2-4
CurrentIndex	2-4
Count	2-5
HasData	2-6
IsDeleted	2-7
RelatedData[]	2-8
ReviewStates	2-9
Rule model properties for items	2-10
[Alias or code of codelist item]	2-11
[Name of codelist]	2-11
Cloud Reference Topic Title	2-11
AllValues[]	2-13
Completed	2-15
CurrentAllObjectsIndex	2-16
CurrentAllValuesIndex	2-16
CurrentObjectsIndex	2-17
CurrentValuesIndex	2-17
Empty	2-18
EnteredUnit	2-18
EnteredValue	2-20
Objects[]	2-22
Selected[]	2-23
Value	2-24
Value[]	2-25
ValueLabel	2-25
Values[]	2-26
Rule model properties for DateTime items	2-28
Year	2-29
YearEmpty	2-29
YearUnknown	2-30
Month	2-30
MonthEmpty	2-30
MonthUnknown	2-31
Day	2-31

DayEmpty	2-31
DayUnknown	2-32
Hour	2-32
HourEmpty	2-32
HourUnknown	2-33
Minute	2-33
MinuteEmpty	2-33
MinuteUnknown	2-34
Second	2-34
SecondEmpty	2-34
SecondUnknown	2-35
Methods for repeating study objects	2-35
[] [Indexer]	2-35
Current()	2-36
Examples--Using the Current() method and IsValueInArray function	2-37
Current(Integer)	2-39
GetValue	2-39
Methods for non-repeating study objects	2-41
GetReviewStates	2-42
HasReviewStates	2-42
HasState(Integer)	2-43
Example--Methods for non-repeating study objects	2-43

## 3 Functions

---

Functions tab of the rule wizard	3-1
Functions in rules and rule templates	3-2
Dynamic prompts in the Expression workspace	3-2
Viewing and editing a function	3-3
Deleteing a function	3-3
About predefined functions	3-3
Date time processing	3-4
Exceptions	3-5
Predefined functions in the system library	3-5
_CalculateBMI	3-7
_CalculateBSA	3-8
_CalculateDateTime	3-9
_CalculateWaistHipRatio	3-11
_CheckPatientInitials	3-12
_CompareDates	3-12
_CompareDatesWithRange	3-13

_Count	3-15
_Count(String, Array, Boolean)	3-15
_Count(Date, Array)	3-16
_Count(Integer, Array)	3-16
_Count(Float, Array)	3-17
_GetCurrentDate	3-17
_GetDateDifference	3-17
GetScreeningNumber	3-19
GetSiteLocale	3-19
GetSiteMnemonic	3-20
GetSiteTime	3-20
GetTrialName	3-20
GetUserName	3-21
GetSubjectNumber	3-21
_IsValueGreaterThanArray	3-22
_IsValueGreaterThanArray (PFDateTime, Array)	3-22
_IsValueGreaterThanArray (Float, Array)	3-22
_IsValueGreaterThanArray (Integer, Array)	3-23
_IsValueGreaterThanOrEqualToArray	3-23
_IsValueGreaterThanOrEqualToArray (PFDateTime, Array)	3-24
_IsValueGreaterThanOrEqualToArray (Float, Array)	3-25
_IsValueGreaterThanOrEqualToArray (Integer, Array)	3-25
_IsValueInArray	3-26
_IsValueInArray (PFDateTime, Array)	3-26
_IsValueInArray (Float, Array)	3-26
_IsValueInArray (Integer, Array)	3-27
_IsValueInArray (Text, Array)	3-28
_IsValueLessThanArray	3-28
_IsValueLessThanArray (PFDateTime, Array)	3-29
_IsValueLessThanArray (Float, Array)	3-29
_IsValueLessThanArray (Integer, Array)	3-30
_IsValueLessThanOrEqualToArray	3-30
_IsValueLessThanOrEqualToArray (PFDateTime, Array)	3-30
_IsValueLessThanOrEqualToArray (Float, Array)	3-31
_IsValueLessThanOrEqualToArray (Integer, Array)	3-32
_NormalizeDate	3-32
_NormalizeDate (PFDateTime, PFDateTime)	3-32
_NormalizeDate (PFDateTime, PFDateTime, PFDateTime)	3-33
_NormalizeDateToMax	3-35
_NormalizeDateToMax (Date)	3-35
_NormalizeDateToMax (Array)	3-36

Randomize	3-37
Example—Using the Randomize function	3-39
_SaveToDb (String, String)	3-41
About user-defined functions	3-43
Function definition requirements	3-43
Recommendations for creating user-defined functions	3-44
Creating a user-defined function	3-45
Importing a user-defined function	3-45
Attributes of user-defined functions	3-46
DesignerFunctionClassification	3-46
DesignerFunction	3-47
DesignerParameter	3-47
Signing user-defined function assemblies	3-47
Securing user-defined functions	3-48
Sample function definition code	3-49

## 4 Constants

---

Constants tab of the Rule Wizard	4-1
Using constants in rules and rule templates	4-1
Predefined constants in the System Library	4-2
Creating a constant	4-3
Constants tab - Option descriptions	4-4
New Constant dialog box - Option descriptions	4-5
Deleting a constant	4-5

## 5 Data mappings

---

Data Mappings tab	5-1
Using data mappings in rules and rule templates	5-1
Icons used on the Data Mappings tab	5-2
Rule model properties for data series	5-4
Count	5-4
Values[]	5-5
Variables[]	5-6
Empty	5-7
Value	5-7
Methods for data sets	5-8
StudyEvent(StudyEvents)	5-9
StudyEvent(StudyEvents, Integer)	5-9
Forms(Forms)	5-10

Form(Forms, Integer)	5-10
Section(Sections)	5-11
Section(Sections, Integer)	5-11
Item(Items)	5-12
[NameOfCustomDataDimension	5-12
Examples—Data set methods in rule expressions	5-13
CurrentStudyEvent()	5-14
Additional study objects in the Data Mappings tab	5-14
Study events	5-15
Forms	5-16
Sections	5-17
Items	5-18

## 6 Methods, operators, and literals

---

Methods	6-1
Math methods	6-1
Abs	6-2
Ceiling	6-2
DivRem	6-2
Exp	6-3
Floor	6-3
IEEERemainder	6-4
Log	6-4
Log10	6-4
Max	6-4
Min	6-5
Pow	6-5
Round	6-6
Sqrt	6-6
Truncate	6-6
Examples—Math methods in rule expressions	6-7
Methods for study objects	6-7
Data set methods	6-7
Operators and literals	6-8
Frequently used operators	6-8
Frequently used literals	6-10

## 7 Sample expressions for data-entry rules

---

Sample expressions that use operators	7-1
---------------------------------------	-----



Sample data-entry rule that uses the Data tab	7-1
Sample data-entry rule that uses rule model properties	7-3
Sample data-entry rules that use methods	7-3
Sample data-entry rule that uses constants	7-6
Sample data-entry rules that use functions	7-8
Sample calculation rules	7-10
Sample data-entry rules that use mappings	7-12

## 8 Option descriptions

---

Rule expressions	8-1
New Rule Template dialog box—Option descriptions	8-1
Rule Templates tab—Option descriptions	8-2
Functions	8-3
Functions tab on the Study and Library information Explorer bars	8-3
Functions tab—Option descriptions	8-3
Edit Function dialog box—Option descriptions	8-4
Constants	8-5
Constants tab on the Study and Library Information Explorer bars	8-5
Constants tab—Option descriptions	8-6
New Constant dialog box—Option descriptions	8-7

# Preface

This preface contains the following sections:

- [Documentation accessibility](#)
- [Related resources](#)
- [Diversity and Inclusion](#)
- [Access to Oracle Support](#)
- [Additional copyright information](#)

## Documentation accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

## Related resources

All documentation and other supporting materials are available on the [Oracle Help Center](#).

## Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

## Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through Support Cloud.

Contact our Oracle Customer Support Services team by logging requests in one of the following locations:

- English interface of Oracle Health Sciences Customer Support Portal (<https://hsgbu.custhelp.com/>)

- Japanese interface of Oracle Health Sciences Customer Support Portal (<https://hsgbu-jp.custhelp.com/>)

You can also call our 24x7 help desk. For information, visit <http://www.oracle.com/us/support/contact/health-sciences-cloud-support/index.html> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Additional copyright information

This documentation may include references to materials, offerings, or products that were previously offered by Phase Forward Inc. Certain materials, offerings, services, or products may no longer be offered or provided. Oracle and its affiliates cannot be held responsible for any such references should they appear in the text provided.

# 1

## Rule expressions

In this chapter:

- [About the rule expression language](#)
- [Dynamic prompts in the Expression workspace](#)
- [Rule templates](#)

### About the rule expression language

Use the rule expression language to create the expression component of a data-entry rule, workflow rule, or global condition.

You create rule expressions in:

- The Expression tab of the Rule Wizard.
- The Expression workspace of the dialog boxes used to create or edit a workflow rule or global condition.

As you type, a list of the rule model components that you can use in the expression (study objects and their properties, functions, constants, and data mappings) appears in the Expression workspace. When you select a rule model component, a tooltip appears to indicate its usage (for example, the parameters and their data types required for using a function). The list changes dynamically to support the contents of the expression as you create it.

You can also use the following methods for creating a rule expression:

- Drag rule model components into the Expression workspace from the tabs that appear on the right side of the workspace.
- Type an expression directly in the Expression workspace.

A rule expression can include any valid C# expression that can appear on the right side of the equals sign (=), including:

- Operators and literals.
- Study objects and their rule model properties.
- Functions.
- Constants defined for the study.
- Data mapping study objects and their rule model properties.
- Methods. You can use any method, including:
  - Math methods.
  - Data set methods.
  - Methods for repeating study objects.

A rule expression cannot include:

- Complex structures, such as if statements (conditionals are allowed) and looping statements.  
Use functions to build more complex rule expressions. For more information, see [Functions tab of the rule wizard](#).
- Multi-line expressions.  
The rule expression does not allow flow-of-control operators, such as *like*, *if*, *then*, *for*, or *while*. You can use parentheses to provide better readability and grouping.
- Semi-colons in rule comments. If a semi-colon is included, deployment fails.

An expression must evaluate to one of the following types:

- Integer
- Float
- Boolean
- Text
- Date time



**Note:**

A rule expression is similar to a switch statement in C++, C#, or Java.

For more information, see:

- [Components of the rule expression language](#)
- [Mappings in rules](#)
- [Conversion to different units in rules](#)
- [Rules with multiple conditions](#)

## Components of the rule expression language

Component	Description
<a href="#">Operators and literals</a>	Operators and literals connect information in the rule expression. Use standard C# and Java operators and literals to create rule expressions.
<a href="#">Data</a>	Data includes the following: <ul style="list-style-type: none"> <li>• Values provided for study objects within the scope of the rule.</li> <li>• Values of rule model properties of study objects within the scope of the rule.</li> <li>• Methods for repeating study objects.</li> </ul>

---

Component	Description
Functions	<p>A function is a reusable piece of code that extends the behavior of a rule. Functions allow experienced programmers to build complex rule expressions and make them available in libraries or studies. A function can be part or all of a rule expression.</p> <p>If a function has parameters, you must specify their substitution values after you drag the function into the rule expression. You can use the following to specify parameter substitution values:</p> <ul style="list-style-type: none"><li>• Numeric values.</li><li>• Values of study objects and their rule model properties.</li><li>• Constants.</li><li>• Another function.</li><li>• Global study objects and their rule model properties.</li></ul>
Constants	<p>A constant is a value that is defined in a library or study and that can be referenced by any rule.</p>
Data mappings	<p>Study objects and properties that have a global scope for rule creation are called data mappings. The following data mappings can be used in any rule in the study in which you are working:</p> <ul style="list-style-type: none"><li>• RefNames of the following study objects in the study or library:<ul style="list-style-type: none"><li>– Mappings.</li><li>– Data sets.</li><li>– Data series.</li><li>– Items added to data series.</li></ul></li><li>• Methods for data sets. Methods are automatically available for all data sets.</li><li>• Rule model properties for data series. If the data series contains one or more items that collect more than one value, then the rule model properties for data series appear so you can specify the value to use.</li></ul>
Methods	<p>A method is a block of code that is called by a rule and that is used to manipulate data. You can use any method, including:</p> <ul style="list-style-type: none"><li>• Math methods.</li><li>• Methods for repeating study objects.</li><li>• Data set methods.</li></ul>

---

## Mappings in rules

### About data mappings

Study objects and properties that are related to mappings are called data mappings. You add data mappings to rule expressions in the Rule Wizard using the dynamic expression prompts in the Expression tab or by dragging them from the Expression tab > Data Mappings tab. The Data Mappings tab lists:

- RefNames of the data mappings, data sets, and data series in the study or library.
- Rule model properties for data series.  
A data series has the properties of the item that is mapped to it. If a data series contains an item that collects more than one value, the rule model properties for repeating study objects appear so you can access an array of all of the values of the item.
- Methods for data sets.  
A method appears if you select the corresponding standard data dimension of the data set. You can use data set methods to return a subset of the data in the data set.
- Study events, forms, sections, and items that are mapped to each data set.  
Study objects appear if you select the corresponding standard data dimension of the data set. The properties of the study objects are used as parameters of data set methods.

### Managing data mappings

You manage data mappings in the Project Explorer, where you create mappings, data sets, and data series and add items to data series. For more information, see [Data Mappings tab](#).

### Mappings and data-entry rules

Adding items to a data series in a mapping allows you to:

- Use vector arithmetic to examine a range of values.  
When an item appears on multiple forms or is part of a repeating form or repeating study event, an array of values is collected for the item.
- Use the items in any data-entry rule in the study, regardless of the level at which the rule is created or the scope of the study object.  
To reuse the data-entry rule in another study, all items must be in the mapping in the other study.

## Conversion to different units in rules

In the Oracle Health Sciences InForm application, when the base unit of an item is in one unit, such as kilograms, but the item allows users to enter different units, such as pounds, the value for the item is stored in the database in two ways—as the entered value (pounds) and the normalized value (pounds converted to kilograms).

For example, if a user types 154 and selects pounds, the value is stored as 154 pounds (for the entered value) and 70 kilograms (for the normalized value). The conversion process from the entered unit to the base unit is called normalization.

Some rules and functions require specific units for item values. For example, a BMI rule might require a weight value to be in kilograms. If a weight item uses pounds

instead of kilograms as the base unit, a conversion from pounds to kilograms has to be done in the rule expression.

If the BMI rule is created on a VitalSigns form with Height, Weight, and BMI items, the rule expression with the conversion information could appear as follows:

```
(this.Weight.Value * 0.45359) / ((this.Height.Value) * (this.Height.Value))
```

Alternately, you can define a constant that performs the conversion. For example, you can define a **LbToKg** constant that equals *0.45359* and use the constant in the expression:

```
(this.Weight.Value * LbtoKg) / ((this.Height.Value) * (this.Height.Value))
```

## Rules with multiple conditions

You can create a rule with multiple conditions using the **? :** operator. The operator allows you to mimic an if/then/else control structure and present multiple conditions to determine whether a rule passes or fails.

The following is an example of a rule that returns a Boolean value. However, you can create a rule to return any data type—for example, a Boolean value, an integer, or a string—for which you can provide a corresponding action.

For example, a rule is used to determine the following information:

- Is the subject pregnant? (True or False)
- Has a severe or life-threatening adverse event occurred? (True or False)

If both are false, the rule passes. If one is true and the other is false, the rule passes. If both are true, the rule checks that the correct termination code was entered when the subject was terminated from the study.

- If the correct termination code was entered, the rule passes.
- If an incorrect termination code was entered, the rule fails, and a query is issued, indicating that the correct termination code must be selected.

The rule checks data on the following forms:

- **Demographics form**—To determine if the subject is pregnant.
- **AE (Adverse Events) form**—To determine if the adverse event that occurred was severe or life threatening.
- **Termination form**—To check the termination code that was used when the subject was terminated from the study.

This rule is created at the study design level because the three forms are in the scope of the study design. The query is created on the Termination form.

### Example 1-1 A rule with multiple conditions

```
evaluate on Form Submission
value = this.InitialVisit.Pregnancy.Pregnant.Value ==
this.InitialVisit.Pregnancy.Pregnant.YesNoCodes.YesNoCode1 &&
(this.AECM.AdverseEvents.Current().Severity.Value ==
this.AECM.AdverseEvents.Current().Severity.SeverityCodes.SeverityCode3 ||
this.AECM.AdverseEvents.Current().Severity.Value ==
this.AECM.AdverseEvents.Current().Severity.SeverityCodes.SeverityCode4 ) ?
this.Termination.TerminationForm.TerminationReason.Value ==
this.Termination.TerminationForm.TerminationReason.TerminationCodes.Terminati
```



```
onCodePREGSEV : true
when value is false
issue query on this.Termination.TerminationForm.TerminationReason: If
the patient is pregnant and a severe or life-threatening AE occurs,
Termination Reason must be filled out with a value of "Severe AE
during Pregnancy"
```

In the previous example, the values collected for items (such as the Pregnancy item) are compared to the codelist item names, not the actual values of the codelist items. You can simplify the rule expression by providing the codelist item values in place of the codelist item names (shown in the following example). However, Oracle recommends that you provide RefNames when writing rules, as this eliminates the need for re-work if study object values change.

The following codes are used in the study:

- In the YesNo codelist:
  - Yes = 1.
- In the SeverityCodes codelist:
  - Severe = 3.
  - Life-Threatening = 4.
- In the TerminationCodes codelist:
  - Severe AE during Pregnancy = PREGSEV

### Example 1-2 A simplified rule expression in a rule with multiple conditions

Using these values, you could simplify the rule syntax:

```
evaluate on Form Submission
value = this.InitialVisit.Pregnancy.Pregnant.Value == 1 &&
(this.AECM.AdverseEvents.Current().Severity.Value == 3 ||
this.AECM.AdverseEvents.Current().Severity.Value == 4) ?
this.Termination.TerminationForm.TerminationReason.Value == PREGSEV :
true
when value is false
issue query on this.Termination.TerminationForm.TerminationReason: If
the patient is pregnant and a severe or life-threatening AE occurs,
Termination Reason must be filled out with a value of "Severe AE
during Pregnancy"
```

## Dynamic prompts in the Expression workspace

When you create an expression, you can type directly in the Expression workspace, and you can drag rule model objects from the tabs at the right side of the windows where you create a data-entry rule, workflow rule, or global condition.

When you type in the Expression workspace, prompts appear dynamically as you type, listing the rule model objects that are available for you to use based on the scope of the rule and the content of the expression. As an alternative to dragging rule model objects from the tabs, you can select the objects from the prompts.

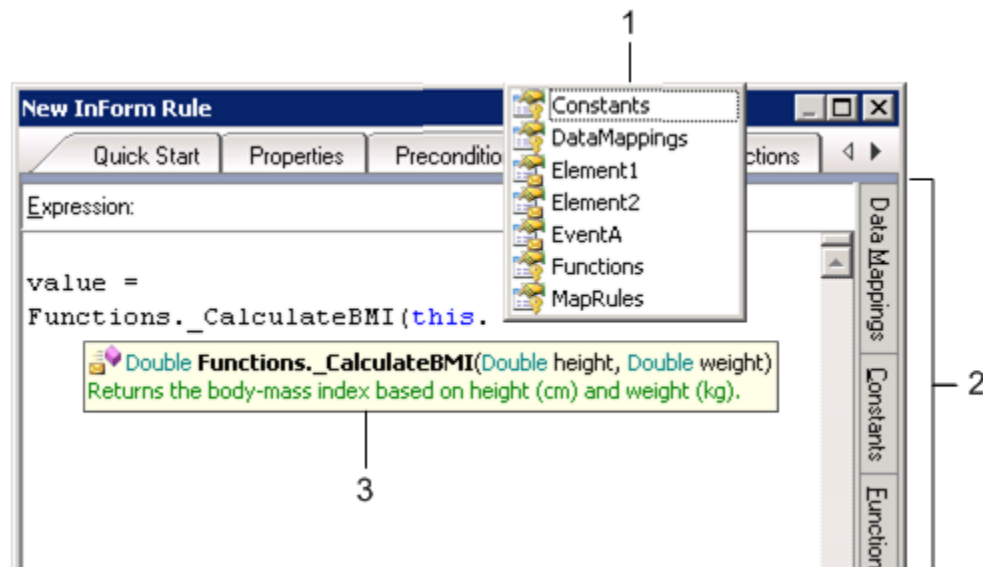
A dynamic expression prompt appears when you type a period. The prompt contains a list of rule model objects that you can select.

- To use the value of a study object or rule model property, type `this.`, and select the study object or property from the dynamic expression prompt.
- To use a function, type `Functions.`, and select the function from the dynamic expression prompt.
- To use a constant, type `Constants.`, and select the constant from the dynamic expression prompt.
- To use a data mapping study object or a rule model property, type `DataMappings.`, and select the study object or rule model property from the dynamic expression prompt.

When you select a rule model object that requires parameter values (for example, a function or method), an open parenthesis mark follows the name of the rule model object, and a tooltip indicates the parameters and their data types. The tooltip also appears when you point to the name of the rule model object.

The following illustration shows a dynamic expression prompt and tooltip that appear when you create a rule expression with a function on a study event that includes several forms. The Rule Wizard tabs containing the same options are visible on the right.

**Figure 1-1 Dynamic expression prompt and tooltip**



1—Dynamic expression prompt. Constants, functions, data mappings, and child study objects are available for selection.

2—Standard Rule Wizard tabs.

3—Tooltip showing format and data types of required parameters for the function.

For more information, see:

- [Selecting a rule model object from a dynamic expression prompt](#)

## Selecting a rule model object from a dynamic expression prompt

When a dynamic expression prompt appears, perform one of the following:

- Double-click a rule model object.
- Select a rule model object, and press the **Tab** or **Enter** key.

 **Note:**

You can navigate the list in the dynamic expression prompt with arrow keys or by typing the first letter of a rule model object.

## Rule templates

To simplify the process of creating rules, experienced programmers can create rule templates to define the most common rules. Non-programmers can then use the rule templates to create rules. A rule created using a rule template is called an intrinsic rule. Intrinsic rules are typically used for simple, item-level checks.

A rule template is a function that is defined on a study object, study object template, or study object type and can be used as the expression clause of a rule.

A rule template defines only the expression clause of an intrinsic rule. You must supply the precondition clause, action clause, and parameters for an intrinsic rule in the Rule Wizard to form a complete rule.

Rule templates are especially useful when they are part of a study object template or type because the rule template is part of all study objects created from the template or type. For example, a rule template that tests the value of an item against a minimum or maximum value can be part of the integer item type. Then, when you create an integer item, the rule template is part of the item. A user who works on the rule can create an intrinsic rule based on the rule template.

 **Note:**

If you modify a rule template, you must re-validate existing rules that use the template, and make necessary changes.

### Rule Templates tab

Rule templates are created and managed using the Rule Templates tab. The Rule Templates tab displays a list of all rule templates that are attached to the study object and a toolbar for adding, modifying, or deleting a rule template.

The Rule Templates tab appears in the Form Editor, Section Editor, and Item Editor. The tab provides a summary of all the rule templates that are attached to the selected study object. You can create rule templates for:

- Forms
- Form templates

- Sections
- Section templates
- Items
- Item templates
- Item types

For more information, see:

- [Creating, modifying, and deleting a rule template?](#)
- [Default rule templates](#)

## Creating, modifying, and deleting a rule template?

You cannot delete a [Rule templates](#) if an existing rule is based on the template.

For more information, see:

- [Create a rule template](#)
- [Modify a rule template](#)
- [Delete a rule template](#)
- [Other descriptions for the New Rule Template dialog box](#)

## Create a rule template

To create a rule template:

1. In the Project Explorer, select a form, section, or item.  
The editor for the study object appears in the workspace.
2. Select the **Rule Templates** tab.
3. Click **Add**.  
The New Rule Template dialog box appears.
4. Fill in the fields in the dialog box.
5. Click **OK**,

## Modify a rule template

To modify a rule template:

1. In the Project Explorer, select a form, section, or item.  
The editor for the study object appears in the workspace.
2. Select the **Rule Templates** tab.
3. In the grid, select the rule template to modify.
4. Click **Edit**.  
The Edit Rule Template dialog box appears.
5. Modify the template as necessary.

## Delete a rule template

To delete a rule template:

1. In the Project Explorer, select a form, section, or item.  
The editor for the study object appears in the workspace.
2. Select the **Rule Templates** tab.
3. In the grid, select the rule template to delete.
4. Click **Delete**.  
The template is removed from the form or item.

## Other descriptions for the New Rule Template dialog box

Option	Description
<b>Properties tab</b>	–
Name	Name of the rule template.
Classification	User-defined term used to organize rule templates.
Description	Description of the rule template.
Display Text	Text that appears in the Rule Summary section of the Rule wizard after the <b>When Value Is</b> information.  If this field is blank, the contents of the Expression workspace are used. If the expression contains parameters, the name of the parameter and the value of the parameter appear. For example, if the expression is <b>value must be between {a} and {b}</b> , and the value of <b>a</b> is 10 and the value of <b>b</b> is 100, the parameters appear as <b>a:10</b> and <b>b:100</b> .
<b>Definition tab</b>	–
Return Type drop-down list	Return type of the rule template; one of the following: Integer, Float, String, Boolean, Date/Time, or Array (A list of values, all of the same type).
Expression	Expression of the rule.
<b>Parameters (Optional)</b>	–
Parameter	Name of the parameter.
Data Type	Return type of the parameter; one of the following: Integer, Float, String, Boolean, Date/Time, or Array (A list of values, all of the same type).
Default Value	Specified value of the parameter.
<b>References</b>	–
Data tab	Lists study objects in the scope of the rule. Optionally, to view the rule model properties of all of the study objects, select <b>Show all</b> .

---

Option	Description
Functions tab	Lists functions registered in a study and the libraries that appear in the Libraries List in the Study Editor. Any rule in the study can reference a function.
Constants tab	Lists constants created in the study and the libraries that appear in the Libraries List in the Study Editor. Any rule in the study can reference a constant.
Data Mappings tab	Lists: <ul style="list-style-type: none"><li>• RefNames of the data mappings, data sets, and data series in the study or library.</li><li>• Rule model properties for data series. A data series has the properties of the item that is mapped to it. If a data series contains an item that collects more than one value, the rule model properties for repeating study objects appear so you can access an array of all of the values of the item.</li><li>• Methods for data sets. A method appears if you select the corresponding standard data dimension of the data set. You can use data set methods to return a subset of the data in the data set.</li><li>• Study events, forms, sections, and items that are mapped to each data set.</li><li>• Study objects appear if you select the corresponding standard data dimension of the data set. The properties of the study objects are used as parameters of data set methods.</li></ul>

---

## Default rule templates

In this section:

- [\\_CheckTextLength](#)
- [\\_DateTimeRangeCheck](#)
- [\\_FutureDateCheck](#)
- [\\_PartialCompletenessCheck](#)
- [\\_RangeCheck](#)
- [\\_RangeCheckInclusive](#)
- [ValidBPCheck](#)

## \_CheckTextLength

Characteristic	Description
<b>Availability</b>	Text items
<b>Description</b>	Checks whether the value entered in the item is less than the maximum length. The maximum length is 255 by default, but you can specify any value when you create a rule based on the template.
<b>Returns</b>	<ul style="list-style-type: none"> <li>• <b>True</b>—The value entered contains fewer than 255 characters.</li> <li>• <b>False</b>—The value entered contains 255 characters or more.</li> </ul>
<b>Display text</b>	<code>this.Value.Length &lt; MaxLength</code>

## \_DateTimeRangeCheck

Characteristic	Description
<b>Availability</b>	Date time items
<b>Description</b>	Checks whether a date falls within a specified range of another date.  When you create a rule based on the template, you specify the two date time items, the date part to compare (by default, days), and the minimum and maximum range.
<b>Returns</b>	<ul style="list-style-type: none"> <li>• <b>True</b>—Date1 is within the specified range of Date2.</li> <li>• <b>False</b>—Date1 is not within the specified range of Date2.</li> </ul>
<b>Display text</b>	<code>_CompareDatesWithRange(date1,date2,datePart,rangeMin,rangeMax)</code>

## \_FutureDateCheck

Characteristic	Description
<b>Availability</b>	Date time items
<b>Description</b>	Checks whether the specified date and time is in the future, as compared to the site date and time.  Site information is based upon the time zone of the data-entry site rather than the time zone of the InForm server.
<b>Returns</b>	<ul style="list-style-type: none"> <li>• <b>True</b>—The entered date and time are in the future, compared to the site date and time.</li> <li>• <b>False</b>—The entered date and time is empty or is in the past, compared to the site date and time.</li> </ul>

Characteristic	Description
Display text	!this.Empty && _CompareDates(Value, GetSiteTime()) > 0

## \_PartialCompletenessCheck

Characteristic	Description
Availability	Compound and blood pressure items
Description	Checks whether child items within a compound item are partially complete. A partially complete compound item consists of one or more empty child items and one or more complete child items.
Returns	<ul style="list-style-type: none"> <li>• <b>True</b>—Child items within a compound item are partially complete (one or more child items are empty, and one or more child items are complete).</li> <li>• <b>False</b>—Child items within a compound item are all either empty or complete.</li> </ul>
Display text	!this.Empty && !this.Complete

## \_RangeCheck

Characteristic	Description
Availability	Integer and float items
Description	<p>Checks whether the value for an integer or float item is greater than a minimum value and less than a maximum value.</p> <p>By default, the minimum value is:</p> <ul style="list-style-type: none"> <li>• 0 for an integer.</li> <li>• 0.0 for a float.</li> </ul> <p>By default, the maximum value is:</p> <ul style="list-style-type: none"> <li>• 100 for an integer.</li> <li>• 100.0 for a float item.</li> </ul> <p>You can change these values when you create a rule based on the template.</p>
Returns	<ul style="list-style-type: none"> <li>• <b>True</b>—The value of the item is greater than the minimum value and less than the maximum value.</li> <li>• <b>False</b>—The value of the item is less than the minimum value or greater than the maximum value.</li> </ul>
Display text	Value must be between {MinValue} and {MaxValue}



## \_RangeCheckInclusive

Characteristic	Description
<b>Availability</b>	Integer and float items
<b>Description</b>	<p>Checks whether the value of a float or integer item is greater than or equal to a minimum value and less than or equal to a maximum value.</p> <p>By default, the minimum value is:</p> <ul style="list-style-type: none"> <li>• 0 for an integer.</li> <li>• 0.0 for a float.</li> </ul> <p>By default, the maximum value is:</p> <ul style="list-style-type: none"> <li>• 100 for an integer.</li> <li>• 100.0 for a float item.</li> </ul> <p>You can change these values when you create a rule based on the template.</p>
<b>Returns</b>	<ul style="list-style-type: none"> <li>• <b>True</b>—The value of the item is greater than or equal to the minimum value and less than or equal to the maximum value.</li> <li>• <b>False</b>—The value of the item is less than the minimum value or greater than the maximum value.</li> </ul>
<b>Display text</b>	Value must be greater than or equal to {MinValue}, and less than or equal to {MaxValue}

## ValidBPCheck

Characteristic	Description
<b>Availability</b>	Blood pressure items
<b>Description</b>	Checks whether the entered systolic value is greater than the entered diastolic value.
<b>Returns</b>	<ul style="list-style-type: none"> <li>• <b>True</b>—The systolic value is greater than the diastolic value.</li> <li>• <b>False</b>—The systolic value is less than the diastolic value.</li> </ul>
<b>Display text</b>	this.SystolicVariable.Value > this.DiastolicVariable.Value

# 2

## Data

In this chapter:

- [Data tab](#)
- [Using data in rules and rule templates](#)
- [Icons used on the Data tab](#)
- [Rule model properties for study events, forms, and sections](#)
- [Rule model properties for items](#)
- [Rule model properties for DateTime items](#)
- [Methods for repeating study objects](#)
- [Methods for non-repeating study objects](#)

### Data tab

A rule expression can include data from a study, including:

- The values of study objects.
- The values of the rule model properties of study objects.  
Data collected in the InForm application has properties, such as Empty and Required, that you can select. These and other properties, collectively called rule model properties, are visible in the Rule Wizard and can be used in rule expressions.
- Review states and review stages.
- A method used to identify an instance of a repeating study object.

### Using data in rules and rule templates

To add data to rule expressions, use one of the following:

- **For workflow rules**—Workflow Expression Editor dialog box.
- **For rule templates**—Edit Rule Template dialog box > Definition tab.
- **For data-entry rules**—Rule Wizard Expression tab > Data tab.
- **For global conditions**—Edit Global Conditions dialog box > Data tab.

#### Note:

Rule expressions use RefNames, not titles. Therefore, RefNames appear in the Rule Wizard and in the rule expression.

In the Data tab (in the Expression tab of the Rule Wizard), the RefName of the study object to which you are adding a rule always appears at the top of the tree. The information that appears below the study object depends on whether **Show all** is selected.







- **Show all not selected.**  
The RefNames of all children of the study object are listed below the study object. Only study objects within the scope of the rule are listed.  
  
You can use the values of any of the study objects in the rule expression.
- **Show all selected.**  
The following information appears:
  - RefNames of all children of the study object.  
Only study objects within the scope of the rule are listed. You use the values of the study objects and properties in the rule expression.
  - Rule model properties of the children of the study object. Rule model properties are available for study events, forms, sections, and items.
  - Methods for repeating study objects.









To use the value of a study object or the value of a rule model property in a rule expression:

- Drag the study object or property from the Data tab to the Expression workspace.

## Icons used on the Data tab

The icons that appear depend on the level on which a rule is created and whether **Show all** is selected.

Icon	Description
	Study design.
	Study element.
	Study event.
	Form.
	<ul style="list-style-type: none"> <li>• DateTime item.</li> </ul> or <ul style="list-style-type: none"> <li>• Rule model property with a DateTime return type.</li> </ul>
	<ul style="list-style-type: none"> <li>• Float item.</li> </ul> or <ul style="list-style-type: none"> <li>• Rule model property with a Float return type.</li> </ul>

Icon	Description
	<ul style="list-style-type: none"> <li>Integer item.</li> </ul> or <ul style="list-style-type: none"> <li>Rule model property with a Integer return type.</li> </ul>
	<ul style="list-style-type: none"> <li>Text item.</li> </ul> or <ul style="list-style-type: none"> <li>Rule model property with a Text return type.</li> </ul>
	Compound item.
	Method.
	Used for some rule model properties for items.
	Rule model property with a Boolean return type.
	[Name of codelist] rule model property.
	[Name of codelist item] rule model property.

## Rule model properties for study events, forms, and sections

You add rule model properties to an expression in the Rule Wizard in the Expression tab > Data tab.



### Note:

The types for rule model properties are .NET base types.



### Note:


All arrays start at 0.

For more information, see:


- [\[Name of review stage\]](#)
- [\[Name of review state\]](#)

- [CurrentIndex](#)
- [Count](#)
- [HasData](#)
- [IsDeleted](#)
- [RelatedData\[\]](#)
- [ReviewStates](#)


## [Name of review stage]


Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Appears for forms when at least one review state has been defined in the study.
<b>Description</b>	Review stage in a custom review state. You can get the review stage of a form using the GetReviewState method. You can check for the review stage of a form using the HasReviewState method.

## [Name of review state]


Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Appears for forms. This property is a container for the stages defined for a custom review stage and cannot be dragged to the Expression workspace. You can drag the named review stages that appear below the property.


## CurrentIndex

Characteristic	Description
<b>Icon</b>	


Characteristic	Description
<b>Availability</b>	Repeating study events, repeating forms, and repeating sections.
	<div data-bbox="1084 365 1380 823" style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>Note:</b></p> <p>To view this property, you must select the parent study object. For example, to view the property on a study event, you must select its parent (either a study element or a study design).</p> </div>
<b>Return type</b>	Int32 (for Oracle Central Designer Integer and YesNo types).
<b>Description</b>	The index of the current study events, form, or section. The index starts at 0.
<b>Purpose</b>	Use this property to determine the section, form, or study event for which someone is currently entering or modifying data. For example, you might want to perform a rule action for the first form for which data was entered and another rule action for subsequent instances of the form.

## Count

Characteristic	Description
<b>Icon</b>	

Characteristic	Description
<b>Availability</b>	Repeating study events, repeating forms, and repeating sections.
	<div data-bbox="1122 365 1458 764" style="border: 1px solid #0070C0; padding: 10px;"> <p> <b>Note:</b></p> <p>To view this property, you must select the parent study object. For example, to view the property on a study event, you must select its parent (either a study element or a study design).</p> </div>
<b>Return type</b>	Int32 (for Oracle Central Designer Integer and YesNo types).
<b>Description</b>	The current number of instances of the repeating study event, form, or section.
<b>Purpose</b>	Use this property to determine the number of sections, forms, or study events in a study. For example, you might want to know the number of adverse events that exist for a subject.

## HasData


Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Non-repeating forms and sections.
<b>Return type</b>	Boolean.
<b>Description</b>	True or False. When True, a value for the item has been provided.

Characteristic	Description
<b>Purpose</b>	<p>Use this property to determine whether a form has data on it. For example, if an Adverse Events form indicates that a subject was hospitalized, you could check whether the hospitalization form has been filled out yet.</p> <pre> value = this.AE.Current().itmHospitalized == 'Y' ? this.Hospitalization.HasData : true when value is false   issue query on this.AE.Current().itmHospitalized:   If patient was hospitalized, Hospitalization form must be filled out. </pre>


**Notes** **Note:**

The functionality available with HasData is now available with HasState. If you are using HasData, consider converting it to HasState(Constants.FormStates.HasData).


## IsDeleted

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Repeating forms and sections.
<b>Return type</b>	Boolean.
<b>Description</b>	True or False. When True, the form or section has been deleted.




Characteristic	Description
<b>Purpose</b>	Use this property to determine whether an itemset (in the Oracle Central Designer application, a section) or a form has been deleted in the InForm application.
<b>Notes</b>	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>Note:</b></p> <p>The functionality available with IsDeleted is now available with HasState. If you are using HasData, consider converting it to HasState(Constants.FormStates.Deleted)</p> </div>

## RelatedData[]

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Repeating, associated form.
<b>Return type</b>	<ul style="list-style-type: none"> <li>Form instances for the associated form.</li> </ul> or <ul style="list-style-type: none"> <li>An array of item values from associated forms.</li> </ul>
<b>Description</b>	Returns instances of associated forms, sorted by form index, and allows access to arrays of item values from the associated forms.

Characteristic	Description
<b>Example</b>	<p>If the AE and CM forms are associated, and you are creating a rule on a study event that contains both of them:</p> <ul style="list-style-type: none"> <li>To return an array of associated AE forms:<code>this.CM.Current().RelatedData</code></li> <li>To return an array of associated CM forms:<code>this.AE.Current().RelatedData</code></li> <li>To return the value of the Verbatim item on the first AE form:<code>this.CM.Current().RelatedData[0].Verbatim.Value</code></li> <li>To return an array of values for the drugName item from all instances of the CM form that are associated with the current instance of the AE form:<code>this.AE.Current().RelatedData.drugName.Values</code></li> </ul>
<b>Notes</b>	<p>Trigger dependencies are created in the Oracle Health Sciences InForm application for a data-entry rule using associated forms. When the expression of a data-entry rule references the RelatedData rule model property, a trigger dependency is created for both the item that is explicitly referenced and the item that is referenced through the RelatedData property.</p> <ul style="list-style-type: none"> <li>When you submit either of the associated forms, the rule runs.</li> <li>When you associate two forms in the Oracle Health Sciences InForm application, a trigger dependency causes rules to run.</li> <li>When you remove a form association in the Oracle Health Sciences InForm application, rules do not run.</li> </ul>

## ReviewStates

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	<p>Appears for forms when at least one custom review state has been defined in the study.</p> <p>This property is a container for custom review states and cannot be dragged to the Expression workspace. You can drag the named review states that appear below the property.</p>

## Rule model properties for items

Rule model properties are different from standard and custom properties, which are properties of the study design and are visible in the Properties Browser.

You add rule model properties to an expression in the Rule Wizard in the Expression tab > Data tab. The properties in this section are available for items of all data types, except where noted.



**Note:**

The types for rule model properties are .NET base types.




**Note:**

All arrays start at 0.


For more information, see:

- [\[Alias or code of codelist item\]](#)
- [\[Name of codelist\]](#)
- [Cloud Reference Topic Title](#)
- [AllValues\[\]](#)
- [Completed](#)
- [CurrentAllObjectsIndex](#)
- [CurrentAllValuesIndex](#)
- [CurrentObjectsIndex](#)
- [CurrentValuesIndex](#)
- [Empty](#)
- [EnteredUnit](#)
- [EnteredValue](#)
- [Objects\[\]](#)
- [Selected\[\]](#)
- [Value](#)
- [Value\[\]](#)
- [ValueLabel](#)
- [Values\[\]](#)


## [Alias or code of codelist item]

Characteristic	Description
<b>Icon</b>	
<b>Availability</b> <b>Return type</b>	Items with a codelist. One of the following: <ul style="list-style-type: none"> <li>• String (for Oracle Central Designer string types).</li> <li>• Double (for Oracle Central Designer float types).</li> <li>• Int32 (for Oracle Central Designer Integer and YesNo types).</li> </ul>
<b>Description</b>	Codelist item in the codelist that is associated with the item.  When you use a codelist item in a rule expression, the value of the codelist item is used.
<b>Purpose</b>	The alias of a codelist item is useful because you can use the alias in place of the code in the codelist. The Rule Wizard allows you to drag the codelist item as a replacement for the code, so you do not have to memorize it or close the Rule Wizard to determine the value.


## [Name of codelist]

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Appears for items that have a codelist.  This property is a container for codelist items and cannot be dragged to the Expression workspace. You can drag the codelists that appear below the property.


## Cloud Reference Topic Title

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	An item that is a child of one of the following: <ul style="list-style-type: none"> <li>• Form that is part of a repeating study event.</li> <li>• Repeating form.</li> <li>• Repeating section.</li> </ul>


Characteristic	Description
<b>Return type</b>	object[]
<b>Description</b>	Returns an array of values from item instances that are part of a repeating study event, repeating form, or repeating section.
<b>Included in the array</b>	<ul style="list-style-type: none"> <li>• Values of undeleted items. An item is undeleted if someone has provided a value for it, and the form or section that contains the item has not been deleted.</li> <li>• Values of deleted items. An item is deleted if the form or section that contains it is deleted from a study.</li> <li>• Values of empty items. An item is empty if the form has been created but a value has not been provided for the item.</li> </ul>
<b>Example</b>	<p>A study contains the following repeating forms, which contain the AEDate item:</p> <ul style="list-style-type: none"> <li>• <b>First instance of the form</b>—Someone has entered a value for the AEDate item.</li> <li>• <b>Second instance</b>—Someone has filled in values for other items but not the AEDate item (the item is empty).</li> <li>• <b>Third instance</b>—Someone entered a value for the AEDate item and deleted the form (the item is deleted).</li> <li>• <b>Fourth instance</b>—Someone has entered a value for the AEDate item.</li> </ul> <p>When you use this property, the following instances are returned:</p> <ul style="list-style-type: none"> <li>• First instance</li> <li>• Second instance</li> <li>• Third instance</li> <li>• Fourth instance</li> </ul>

Characteristic	Description
Notes	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>Note:</b></p> <p>Values[], AllValues[], Objects[], and AllObjects[] all return an array of data. Consider the following scenarios when deciding upon the property to use:</p> <ul style="list-style-type: none"> <li>• To include deleted values, use either AllValues[] or AllObjects[].</li> <li>• To include null values, use Objects[] or AllObjects[].</li> </ul> </div> <p>Additionally, consider that the following CurrentIndex properties are also available:</p> <ul style="list-style-type: none"> <li>• <a href="#">CurrentAllObjectsIndex</a></li> <li>• <a href="#">CurrentAllValuesIndex</a></li> <li>• <a href="#">CurrentObjectsIndex</a></li> <li>• <a href="#">CurrentValuesIndex</a></li> </ul>


## AllValues[]

Characteristic	Description
Icon	
Availability	<p>An item that is a child of one of the following:</p> <ul style="list-style-type: none"> <li>• Form that is part of a repeating study event.</li> <li>• Repeating form.</li> <li>• Repeating section.</li> </ul>
Return type	<ul style="list-style-type: none"> <li>• <b>If an integer type</b>—Int[].</li> <li>• <b>If a float type</b>—Double[].</li> <li>• <b>If a text type</b>—String[].</li> <li>• <b>If a date time type</b>—PFDateTime[].</li> </ul>
Description	Returns an array of values from item instances that are part of a repeating study event, repeating form, or repeating section.


Characteristic	Description
<b>Included in the array</b>	<ul style="list-style-type: none"> <li>• Values of undeleted items. An item is undeleted if someone has provided a value for it, and the form or section that contains the item has not been deleted.</li> <li>• Values of deleted items. An item is deleted if the form or section that contains it is deleted from a study.</li> </ul>
<b>Not included in the array</b>	<p>Values of empty items. An item is empty if the form has been created but a value has not been provided for the item.</p>
<b>Example</b>	<p>A study contains the following repeating forms, which contain the AEDate item:</p> <ul style="list-style-type: none"> <li>• <b>First instance of the form</b>—Someone has entered a value for the AEDate item.</li> <li>• <b>Second instance</b>—Someone has filled in values for other items but not the AEDate item (the item is empty).</li> <li>• <b>Third instance</b>—Someone entered a value for the AEDate item and deleted the form (the item is deleted).</li> <li>• <b>Fourth instance</b>—Someone has entered a value for the AEDate item.</li> </ul> <p>When you use this property, the following instances are returned:</p> <ul style="list-style-type: none"> <li>• First instance</li> <li>• Third instance</li> <li>• Fourth instance</li> </ul>

Characteristic	Description
<b>Notes</b>	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>Note:</b></p> <p>Values[], AllValues[], Objects[], and AllObjects[] all return an array of data. Consider the following scenarios when deciding upon the property to use:</p> <ul style="list-style-type: none"> <li>• To include deleted values, use either AllValues[] or AllObjects[].</li> <li>• To include null values, use Objects[] or AllObjects[].</li> </ul> </div> <p>Additionally, consider that the following CurrentIndex properties are also available:</p> <ul style="list-style-type: none"> <li>• <a href="#">CurrentAllObjectsIndex</a></li> <li>• <a href="#">CurrentAllValuesIndex</a></li> <li>• <a href="#">CurrentObjectsIndex</a></li> <li>• <a href="#">CurrentValuesIndex</a></li> </ul>


## Completed

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Compound items.
<b>Return type</b>	Boolean.




Characteristic	Description
<b>Description</b>	<div style="border: 1px solid #0070C0; padding: 10px; margin-bottom: 10px;"> <p> <b>Note:</b></p> <p>Items that are conditional on child items of the compound item are not evaluated.</p> </div> <ul style="list-style-type: none"> <li>• <b>True</b>—All child items of the conditional item are not empty.</li> <li>• <b>False</b>—At least one child item of the conditional item is empty.</li> </ul>
<b>Purpose</b>	<p>This property allows you to simplify the expressions of data-entry rules that check to see if controls that have been started are completed. Typically these expressions also use the Empty rule model property.</p>

## CurrentAllObjectsIndex

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	<p>An item that is a child of one of the following:</p> <ul style="list-style-type: none"> <li>• Form that is part of a repeating study event.</li> <li>• Repeating form.</li> <li>• Repeating section.</li> </ul>
<b>Return type</b>	Int32 (for Central Designer Integer and YesNo types).
<b>Description</b>	Returns the current position in the array that is returned by the AllObjects[] rule model property. The index is 0 based.

## CurrentAllValuesIndex

Characteristic	Description
<b>Icon</b>	

Characteristic	Description
<b>Availability</b>	An item that is a child of one of the following: <ul style="list-style-type: none"> <li>• Form that is part of a repeating study event.</li> <li>• Repeating form.</li> <li>• Repeating section.</li> </ul>
<b>Return type</b>	Int32 (for Oracle Central Designer Integer and YesNo types).
<b>Description</b>	Returns the current position in the array that is returned by the AllValues[] rule model property. The index is 0 based.


## CurrentObjectsIndex

Characteristic	Description
<b>Icon</b>	<a href="#">123</a>
<b>Availability</b>	An item that is a child of one of the following: <ul style="list-style-type: none"> <li>• Form that is part of a repeating study event.</li> <li>• Repeating form.</li> <li>• Repeating section.</li> </ul>
<b>Return type</b>	Int32 (for Oracle Central Designer Integer and YesNo types).
<b>Description</b>	Returns the current position in the array that is returned by the Objects[] rule model property. The index is 0 based.


## CurrentValuesIndex

Characteristic	Description
<b>Icon</b>	<a href="#">123</a>
<b>Availability</b>	An item that is a child of one of the following: <ul style="list-style-type: none"> <li>• Form that is part of a repeating study event.</li> <li>• Repeating form.</li> <li>• Repeating section.</li> </ul>
<b>Return type</b>	Int32 (for Oracle Central Designer Integer and YesNo types).
<b>Description</b>	Returns the current position in the array that is returned by the Values[] rule model property. The index is 0 based.

## Empty

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	A data series that contains a single item that is used only once in a study. Available below the Variables[ ] property.
<b>Return type</b>	Boolean.
<b>Description</b>	True or False. When True, a value for the item has not been provided.



## EnteredUnit

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	An item with a base unit selected on the Design tab for the item.
<b>Return type</b>	String (for Oracle Central Designer string types).
<b>Description</b>	Returns the localized name of the unit, or null if a unit was not selected. For example, you can use this property in query or email text.

Characteristic	Description
<b>Purpose</b>	<p>If you allow Oracle Health Sciences InForm users to choose between metric and imperial units but want to require that the user is consistent within a form, use this property to check that the values are from the same measurement system. Consider that values entered in the Oracle Health Sciences InForm application are normalized to their base unit. For example, if kg is the base unit, a value of 150 lbs. is stored in the database as 68.0388555 kg.</p> <p>Some unit conversions are not absolute, which can make writing rules difficult. For example, converting mmol/L requires knowledge of a chemical's molecular weight.</p> <ul style="list-style-type: none"><li>• To convert mmol/L of glucose to mg/dL, you multiply by 18.</li><li>• To convert mmol/L of LDL cholesterol to mg/dL, you multiply by 39.</li></ul> <p>You have the following options for recording this information:</p> <ul style="list-style-type: none"><li>• If you use a single item, you must create separate unit definitions for each chemical that is measured.</li><li>• If you use a compound item, with one child to store the value and one child to store the unit, you must perform a conversion to normalize the values for reporting or CDD purposes.</li><li>• If you use EnteredValue and EnteredUnit, you can store the true normalized value in a hidden field, which you can use for reporting or CDD.</li></ul> <p>The following rule example uses EnteredUnit and EnteredValue to convert glucose from mmol/L to mg/dL, and stores that value in the itmNormalizedGlucose item. Consider that a conversion is not done if another unit is chosen; if the base unit is selected, EnteredValue and Value are the same.</p> <pre>value = this.itmGlucose.EnteredUnit == "mmol/L" ? this.itmGlucose.EnteredValue * 18 : this.itmGlucose.Value always     set this.itmNormalizedGlucose = value</pre> <p>You can write a similar rule to convert cholesterol, triglycerides, and other measurements. Additionally, you can replace numerical values, such as 18 and 39, with constants such as</p>


Characteristic	Description
	Constants.Conversions.Glucose and Constants.Conversions.Cholesterol. You define the constants in the study or a library.


## EnteredValue

Characteristic	Description
<b>Icon</b>	<p></p> <p>or</p> <p></p>
<b>Availability</b>	An item with a base unit selected on the Design tab for the item.
<b>Return type</b>	<p>One of the following:</p> <ul style="list-style-type: none"> <li>• Int32 (for Oracle Central Designer Integer and YesNo types).</li> <li>• Double (for Oracle Central Designer float types).</li> </ul>
<b>Description</b>	Returns the entered value for an item for which a base unit was selected.


Characteristic	Description
<b>Purpose</b>	<p>Some unit conversions are not absolute, which can make writing rules difficult. For example, converting mmol/L requires knowledge of a chemical's molecular weight.</p> <ul style="list-style-type: none"><li>• To convert mmol/L of glucose to mg/dL, you multiply by 18.</li><li>• To convert mmol/L of LDL cholesterol to mg/dL, you multiply by 39.</li></ul> <p>You have the following options for recording this information:</p> <ul style="list-style-type: none"><li>• If you use a single item, you must create separate unit definitions for each chemical that is measured.</li><li>• If you use a compound item, with one child to store the value and one child to store the unit, you must perform a conversion to normalize the values for reporting or CDD purposes.</li><li>• If you use EnteredValue and EnteredUnit, you can store the true normalized value in a hidden field, which you can use for reporting or CDD.</li></ul> <p>The following rule example uses EnteredUnit and EnteredValue to convert glucose from mmol/L to mg/dL, and stores that value in the itmNormalizedGlucose item. Consider that a conversion is not done if another unit is chosen; if the base unit is selected, EnteredValue and Value are the same.</p> <pre>value = this.itmGlucose.EnteredUnit == "mmol/L" ? this.itmGlucose.EnteredValue * 18 : this.itmGlucose.Value always     set this.itmNormalizedGlucose = value</pre> <p>You can write a similar rule to convert cholesterol, triglycerides, and other measurements. Additionally, you can replace numerical values, such as 18 and 39, with constants such as Constants.Conversions.Glucose and Constants.Conversions.Cholesterol. You define the constants in the study or a library.</p>

## Objects[]

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	An item that is a child of one of the following: <ul style="list-style-type: none"> <li>Form that is part of a repeating study event.</li> <li>Repeating form.</li> <li>Repeating section.</li> </ul>
<b>Return type</b>	object[]
<b>Description</b>	Returns an array of values from item instances that are part of a repeating study event, repeating form, or repeating section.
<b>Included in the array</b>	<ul style="list-style-type: none"> <li>Values of undeleted items. An item is undeleted if someone has provided a value for it, and the form or section that contains the item has not been deleted.</li> <li>Values of empty items. An item is empty if the form has been created but a value has not been provided for the item.</li> </ul>
<b>Not included in the array</b>	Values of deleted items. An item is deleted if the form or section that contains it is deleted from a study.
<b>Example</b>	<p>A study contains the following repeating forms, which contain the AEDate item:</p> <ul style="list-style-type: none"> <li><b>First instance of the form</b>—Someone has entered a value for the AEDate item.</li> <li><b>Second instance</b>—Someone has filled in values for other items but not the AEDate item (the item is empty).</li> <li><b>Third instance</b>—Someone entered a value for the AEDate item and deleted the form (the item is deleted).</li> <li><b>Fourth instance</b>—Someone has entered a value for the AEDate item.</li> </ul> <p>When you use this property, the following instances are returned:</p> <ul style="list-style-type: none"> <li>First instance</li> <li>Second instance</li> <li>Fourth instance</li> </ul>

Characteristic	Description
<b>Notes</b>	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>Note:</b></p> <p>Values[], AllValues[], Objects[], and AllObjects[] all return an array of data. Consider the following scenarios when deciding upon the property to use:</p> <ul style="list-style-type: none"> <li>• To include deleted values, use either AllValues[] or AllObjects[].</li> <li>• To include null values, use Objects[] or AllObjects[].</li> </ul> </div> <p>Additionally, consider that the following CurrentIndex properties are also available:</p> <ul style="list-style-type: none"> <li>• <a href="#">CurrentAllObjectsIndex</a></li> <li>• <a href="#">CurrentAllValuesIndex</a></li> <li>• <a href="#">CurrentObjectsIndex</a></li> <li>• <a href="#">CurrentValuesIndex</a></li> </ul>





## Selected[]

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Available for items with multi-select codelists.
<b>Return type</b>	Boolean.
<b>Description</b>	<p>For the codelist item that is specified in the brackets, returns a value that indicates if the checkbox for the codelist item is selected.</p> <p>You can also use this property to select or deselect the checkbox for a codelist item.</p>




Characteristic	Description
<b>Purpose</b>	<p>Use this property to determine whether a checkbox has been selected. For example, to disqualify subjects who are using aspirin, you can create a list of common OTC drugs with a checkbox next to each.</p> <pre> value = this.itmOTCMeds.Selected[this.itmOTCM eds.clOTCMeds.Aspirin] when value is true     issue query on this.itmOTCMeds:     Subjects on an aspirin regimen are not eligible for the study. </pre> <p>You can also use the property to programmatically select items. To select an item, use a SetValue action, setting the value to True for a selected checkbox or False for a deselected checkbox.</p>


## Value

Characteristic	Description
<b>Icon</b>	<p></p> <p>,</p> <p></p> <p>,</p> <p></p> <p>, or</p> <p></p>
<b>Availability</b>	A data series that contains a single item that is used only once in a study.
<b>Return type</b>	<p>One of the following:</p> <ul style="list-style-type: none"> <li>String (for Oracle Central Designer string types).</li> <li>DateTime (for Oracle Central Designer date time types).</li> <li>Double (for Oracle Central Designer float types).</li> <li>Int32 (for Oracle Central Designer Integer and YesNo types).</li> </ul>
<b>Description</b>	Value of the item in the data series.

## Value[]


Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	All items with multi-select codelists.
<b>Return type</b>	<ul style="list-style-type: none"> <li>• <b>If an integer type</b>—Int[].</li> <li>• <b>If a float type</b>—Double[].</li> <li>• <b>If a text type</b>—String[].</li> </ul>
<b>Description</b>	Returns an array of values for the item.
<b>Purpose</b>	<p>Use this property to process all of the boxes that are selected in a checkbox group. To use a single checkbox item, use <a href="#">Selected[]</a>.</p> <p>For example, if you have a codelist with checkboxes and want to make sure that no more than three checkboxes are selected, use this property to return an array of the selected checkboxes, and use the Length property of the array to determine the number of selected checkboxes.</p> <pre>value = this.itmConcomMeds.Value.Length when value &gt; 3   issue query on this.itmConcomMeds:   Subject should not be on more than   three   concomitant meds at time of   enrollment</pre>

## ValueLabel


Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	All items with a codelist
<b>Return type</b>	String (for Oracle Central Designer string types).
<b>Description</b>	<p>Returns the text value of the selected radio button or item in a drop-down list.</p> <p>If a value is not selected, the rule stops running.</p>

Characteristic	Description
<b>Purpose</b>	<p>This property prevents you from needing to memorize codes for codelist items and drag codelist item aliases. For example, if you have a Route of Administration drop-down list for medications and you are using the FDA codes, Spinal corresponds to 356. You can write the expression in the following way:</p> <pre>value = (this.itmRouteOfAdministration.Value == 356)</pre> <p>However, this expression requires that you know the value of the code. This property allows you to write the expression in the following way:</p> <pre>value = (this.itmRouteOfAdministration.ValueL abel == "Spinal")</pre> <p>Alternatively, you can perform multiple rule actions:</p> <pre>value = this.itmRouteOfAdministration.ValueLa bel when value is "Spinal"... when value is "Oral"... when value is "Intravenous"...</pre>

## Values[]

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	<p>An item that is a child of one of the following:</p> <ul style="list-style-type: none"> <li>Form that is part of a repeating study event.</li> <li>Repeating form.</li> <li>Repeating section.</li> </ul>
<b>Return type</b>	<ul style="list-style-type: none"> <li><b>If an integer type</b></li> <li><b>If a float type</b>—Double[].</li> <li><b>If a text type</b></li> <li><b>If a date time type</b></li> </ul>
<b>Description</b>	Returns an array of values from item instances that are part of a repeating study event, repeating form, or repeating section.

Characteristic	Description
<b>Included in the array</b>	Values of undeleted items. An item is undeleted if someone has provided a value for it, and the form or section that contains the item has not been deleted.
<b>Not included in the array</b>	<ul style="list-style-type: none"><li>• Values of deleted items. An item is deleted if the form or section that contains it is deleted from a study.</li><li>• Values of empty items. An item is empty if the form has been created but a value has not been provided for the item.</li></ul>
<b>Example</b>	<p>A study contains the following repeating forms, which contain the AEDate item:</p> <ul style="list-style-type: none"><li>• <b>First instance of the form</b>—Someone has entered a value for the AEDate item.</li><li>• <b>Second instance</b>—Someone has filled in values for other items but not the AEDate item (the item is empty).</li><li>• <b>Third instance</b>—Someone entered a value for the AEDate item and deleted the form (the item is deleted).</li><li>• <b>Fourth instance</b>—Someone has entered a value for the AEDate item.</li></ul> <p>When you use this property, the following instances are returned:</p> <ul style="list-style-type: none"><li>• First instance</li><li>• Fourth instance</li></ul>

Characteristic	Description
Notes	<div data-bbox="1084 302 1378 1083" style="border: 1px solid #0070C0; padding: 10px;"><p> <b>Note:</b></p><p>Values[], AllValues[], Objects[], and AllObjects[] all return an array of data. Consider the following scenarios when deciding upon the property to use:</p><ul style="list-style-type: none"><li>• To include deleted values, use either AllValues[] or AllObjects[].</li><li>• To include null values, use Objects[] or AllObjects[].</li></ul></div> <p>Additionally, consider that the following CurrentIndex properties are also available:</p> <ul style="list-style-type: none"><li>• <a href="#">CurrentAllObjectsIndex</a></li><li>• <a href="#">CurrentAllValuesIndex</a></li><li>• <a href="#">CurrentObjectsIndex</a></li><li>• <a href="#">CurrentValuesIndex</a></li></ul>


## Rule model properties for DateTime items

In this section:


- [Year](#)
- [YearEmpty](#)
- [YearUnknown](#)
- [Month](#)
- [MonthEmpty](#)
- [MonthUnknown](#)
- [Day](#)
- [DayEmpty](#)
- [DayUnknown](#)

- [Hour](#)
- [HourEmpty](#)
- [HourUnknown](#)
- [Minute](#)
- [MinuteEmpty](#)
- [MinuteUnknown](#)
- [Second](#)
- [SecondEmpty](#)
- [SecondUnknown](#)


## Year

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Available under the <b>Value</b> rule model property for DateTime items. This property appears only when <b>Allow</b> is selected for the <b>Year</b> property of the item.
<b>Return type</b>	Int32 (for Oracle Central Designer Integer and YesNo types).
<b>Description</b>	Value of the <b>Year</b> component of the DateTime item.


## YearEmpty

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Available under the <b>Value</b> rule model property for DateTime items. This property appears only when <b>Required</b> is not selected for the <b>Year</b> property of the item.
<b>Return type</b>	Boolean.
<b>Description</b>	True or False. When True, a value for the part of the date time item has not been provided.


## YearUnknown

Characteristic	Description
Icon	
Availability	Available under the <b>Value</b> rule model property for DateTime. This property appears only when <b>Allow unknown</b> is selected for the Year property of the item.
Return type	Boolean.
Description	True or False. When True, the Unknown value was selected for the date time part.


## Month

Characteristic	Description
Icon	
Availability	Available under the <b>Value</b> rule model property for DateTime items. This property appears only when <b>Allow</b> is selected for the <b>Month</b> property of the item.
Return type	Int32 (for Oracle Central Designer Integer and YesNo types).
Description	Value of the <b>Month</b> component of the DateTime item.


## MonthEmpty

Characteristic	Description
Icon	
Availability	Available under the <b>Value</b> rule model property for DateTime items. This property appears only when <b>Required</b> is not selected for the <b>Month</b> property of the item.
Return type	Boolean.
Description	True or False. When True, a value for the part of the date time item has not been provided.

## MonthUnknown


Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Available under the <b>Value</b> rule model property for DateTime. This property appears only when <b>Allow unknown</b> is selected for the <b>Month</b> property of the item.
<b>Return type</b>	Boolean.
<b>Description</b>	True or False. When True, the Unknown value was selected for the date time part.

## Day

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Available under the <b>Value</b> rule model property for DateTime items. This property appears only when <b>Allow</b> is selected for the <b>Day</b> property of the item.
<b>Return type</b>	Int32 (for Oracle Central Designer Integer and YesNo types).
<b>Description</b>	Value of the <b>Day</b> component of the DateTime item.


(Optional) Enter reference information in this section.

## DayEmpty


Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Available under the <b>Value</b> rule model property for DateTime items. This property appears only when <b>Required</b> is not selected for the <b>Day</b> property of the item.
<b>Return type</b>	Boolean.
<b>Description</b>	True or False. When True, a value for the part of the date time item has not been provided.




## DayUnknown

Characteristic	Description
Icon	
Availability	Available under the <b>Value</b> rule model property for DateTime. This property appears only when <b>Allow unknown</b> is selected for the <b>Day</b> property of the item.
Return type	Boolean.
Description	True or False. When True, the Unknown value was selected for the date time part.


## Hour

Characteristic	Description
Icon	
Availability	Available under the <b>Value</b> rule model property for DateTime items. This property appears only when <b>Allow</b> is selected for the <b>Hour</b> property of the item.
Return type	Int32 (for Oracle Central Designer Integer and YesNo types).
Description	Value of the <b>Hour</b> component of the DateTime item.


## HourEmpty

Characteristic	Description
Icon	
Availability	Available under the <b>Value</b> rule model property for DateTime items. This property appears only when <b>Required</b> is not selected for the <b>Hour</b> property of the item.
Return type	Boolean.
Description	True or False. When True, a value for the part of the date time item has not been provided.


## HourUnknown

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Available under the <b>Value</b> rule model property for DateTime. This property appears only when <b>Allow unknown</b> is selected for the <b>Hour</b> property of the item.
<b>Return type</b>	Boolean.
<b>Description</b>	True or False. When True, the Unknown value was selected for the date time part.


## Minute

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Available under the <b>Value</b> rule model property for DateTime items. This property appears only when <b>Allow</b> is selected for the <b>Minute</b> property of the item.
<b>Return type</b>	Int32 (for Oracle Central Designer Integer and YesNo types).
<b>Description</b>	Value of the <b>Minute</b> component of the DateTime item.


## MinuteEmpty

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Available under the <b>Value</b> rule model property for DateTime items. This property appears only when <b>Required</b> is not selected for the <b>Minute</b> property of the item.
<b>Return type</b>	Boolean.
<b>Description</b>	True or False. When True, a value for the part of the date time item has not been provided.


## MinuteUnknown

Characteristic	Description
Icon	
Availability	Available under the <b>Value</b> rule model property for DateTime. This property appears only when <b>Allow unknown</b> is selected for the <b>Minute</b> property of the item.
Return type	Boolean.
Description	True or False. When True, the Unknown value was selected for the date time part.


## Second

Characteristic	Description
Icon	
Availability	Available under the <b>Value</b> rule model property for DateTime items. This property appears only when <b>Allow</b> is selected for the <b>Second</b> property of the item.
Return type	Int32 (for Oracle Central Designer Integer and YesNo types).
Description	Value of the <b>Second</b> component of the DateTime item.

## SecondEmpty

Characteristic	Description
Icon	
Availability	Available under the <b>Value</b> rule model property for DateTime items. This property appears only when <b>Required</b> is not selected for the <b>Second</b> property of the item.
Return type	Boolean.
Description	True or False. When True, a value for the part of the date time item has not been provided.

## SecondUnknown

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Available under the <b>Value</b> rule model property for DateTime. This property appears only when <b>Allow unknown</b> is selected for the <b>Second</b> property of the item.
<b>Return type</b>	Boolean.
<b>Description</b>	True or False. When True, the Unknown value was selected for the date time part.

## Methods for repeating study objects

You use methods for rules that refer to repeating study events, repeating forms, and repeating sections. The data collected for repeating study objects comprise an array. To include information about a repeating study object in a rule, you must use a method to specify the instance in the array.

Methods for repeating study objects appear in the Rule Wizard in the Expression tab > Data tab.


To use a method in a rule expression:

1. Double-click the method.  
If the method has one or more parameters, the Invoke Function dialog box appears.
2. Drag a study object from the **References** tab to the **Values** field to define a value for each parameter.

For more information, see:


- [\[\] \[Indexer\]](#)
- [Current\(\)](#)
- [Current\(Integer\)](#)
- [GetValue](#)

## [] [Indexer]

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Repeating study events, repeating forms, and repeating sections.
<b>Return type and description</b>	Indexer method. Returns an instance of a repeating study object.
<b>Syntax</b>	<b>NameOfStudyObject[index]</b>

Characteristic	Description
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• <b>Parameter—index.</b></li> <li>• <b>Definition</b>—An integer from 0 to the length of the array minus 1 (Count - 1).</li> <li>• <b>Data type</b>—Integer.</li> </ul>
<b>Example</b>	<p>If form RefName is <b>ConMed</b>, the following example returns the first instance of the repeating form:</p> <p><b>ConMed[0]</b></p>
<b>Purpose</b>	<p>This method is useful when you are working with an array of values. For example, use the following expression to determine the date on which the first adverse event was entered:</p> <pre>this.frmAE[0].itmDateOfOnset.Value</pre> <p>To determine the date on which the last adverse event was entered, use the following expression:</p> <pre>this.frmAE[this.frmAE.Count - 1].itmDateOfOnset.Value</pre>

## Current()

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Repeating study events, repeating forms, and repeating sections.
<b>Return type and description</b>	Returns the current instance of the repeating study object.
<b>Syntax</b>	<b>NameOfStudyObject.Current( )</b>
<b>Parameters</b>	No parameters.
<b>Example</b>	<p>If form RefName is <b>ConMed</b>:</p> <p><b>ConMed.Current( )</b></p> <p>For more information, see <a href="#">Examples--Using the Current() method and IsValueInArray function.</a></p>

Characteristic	Description
<b>Best practices for rules</b>	<ul style="list-style-type: none"> <li>To create a rule that uses a repeating study object's current instance to determine if the conditions of the rule are satisfied, you need to determine whether all instances or a single instance of the repeating study object should be considered in the rule logic.</li> <li>If the rule logic considers more than one instance of the repeating study object, Oracle recommends that you use an array of the repeating study object, using the <code>IsValueInArray</code> function. In the expression, Oracle recommends checking whether the array has any values before using it for comparison.</li> <li>The target of the rule action must be a child study object of the study object referenced by the <code>Current()</code> method. In addition, validation considers the lowest level of <code>Current()</code> that you use. For example, for <code>Visit1</code> and <code>Form1</code>: <ul style="list-style-type: none"> <li>The following expression is valid only if you apply the rule to a child study object of <code>Visit1</code>. <b><code>this.Visit1.Current().Form1.I1 == 1</code></b></li> <li>The following expression is valid only if you apply the rule to a child study object of <code>Section1</code>. <b><code>this.Visit1.Current().Form1.Current().Section1.Current()I1 == 1</code></b></li> <li>The following expression is valid only if you apply the rule to a child study object of <code>Form1</code>. <b><code>this.Visit1.Current().Form1.Current().I1 == 1</code></b></li> </ul> </li> </ul>

For more information, see:

- [Examples--Using the `Current\(\)` method and `IsValueInArray` function](#)

## Examples--Using the `Current()` method and `IsValueInArray` function

The following examples show the following:

- Rule expressions that use the `Current()` rule method.
- For each correct rule expression that uses the `Current()` method, an alternative rule expression that uses the `IsValueInArray` function. Each example that uses the `IsValueInArray` function is correct, and passes validation.

The examples use the following abbreviations for study objects:

- evt—Study event
- frm—Form
- sct—Section
- int—Integer item

- float—Float item
- yn—Yes/No codelist

**Example—Data-entry rule:**

Rule expression	Correct—Rule passes validation	Incorrect—Validation error occurs
Data-entry rule	–	–
this.frm1a.Current().int1.Value == 1	query on this.frm1a.Current().sctbc.float 1: query text	query on this.frm1.float1: query text
Data-entry rule using an array	–	–
this.frm1a.int1.Values.Count>0 && Functions._IsValueInArray(1,this.frm1a.int1.Values) == true	query on this.frm1a.Current().sctbc.float 1: query text	N/A
–	query on this.frm1.float1: query text	N/A

**Example—Workflow rule:**


Rule expression	Correct—Rule passes validation	Incorrect—Validation error occurs
Workflow rule	–	–
this.evt6.Current().frm2a.sct2b.yn2.Value == this.evt6.Current().frm2a.sct2b.yn2.YesNoCodelist2.YesNoCodeListItem2	You can use this rule expression to trigger a form in study event <b>evt6</b> .	You cannot use this rule expression to trigger a form in a study event other than <b>evt6</b> .
Workflow rule using an array	–	–
this.evt6.frm2a.sct2b.yn2.AllValues.Count>0 && Functions._IsValueInArray(this.evt6[0].frm2a[0].sct2b[0].yn2.YesNoCodelist2.YesNoCodeListItem2,this.evt6.frm2a.sct2b.yn2.AllValues)==true	You can use this rule expression to trigger any study event.	N/A

**Example—Global condition:**

Rule expression	Correct—Rule passes validation	Incorrect—Validation error occurs
Global condition	–	–
this.evt1a.Current().frm3.yn1.Value == this.evt1a.Current().frm3.yn1.YesNoCodelist.YesNoCodeListItem	You can use this rule expression to trigger form <b>frm1</b> in study event <b>evt1a</b> .	You cannot use this rule expression to trigger form <b>frm1</b> in a study event other than <b>evt1a</b> .
Global condition using an array	–	–

Rule expression	Correct—Rule passes validation	Incorrect—Validation error occurs
<code>this.evt1a.frm3.yn1.Values.Count&gt;0 &amp;&amp; Functions._IsValueInArray(this.evt1a[0].frm3.yn1.YesNoCodeList.YesNoCodeListItem,this.evt1a.frm3.yn1.Values)== true</code>	You can use this rule to trigger: <ul style="list-style-type: none"> <li>Form <b>frm1</b> in study event <b>evt1a</b></li> <li>Form <b>frm1</b> in study event <b>evt2</b></li> </ul>	N/A

## Current(Integer)


Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	Repeating study events, repeating forms, and repeating sections.
<b>Return type and description</b>	Allows you to choose the instance relative to the current instance of the repeating study object. This method skips deleted instances.
<b>Syntax</b>	<code>NameOfStudyObject.Current(relativeIndex)</code>
<b>Parameters</b>	<ul style="list-style-type: none"> <li><b>Parameter—relativeIndex.</b></li> <li><b>Definition—(Optional)</b> An integer with a range of <code>(-CurrentIndex)</code> to <code>(Count - CurrentIndex - 1)</code>.</li> <li><b>Data type—Integer.</b></li> </ul>
<b>Example</b>	If form RefName is <b>ConMed</b> : <code>ConMed.Current(-1)</code>
<b>Purpose</b>	Use <code>Current</code> with the relative index to compare values across repeating study events and forms. For example, to confirm that the DOV for each unexpected visit is after the DOV for the previous visit, use the following expression: <pre>value = _CompareDates(this.Current().frmDOV.itmDOV.Value, this.Current(-1).frmDOV.itmDOV.Value) when value = -1   issue query on this.Current().frmDOV.itmDOV: DOV   for this visit must be later than the previous visit.</pre>

## GetValue

Appears as one of the following, depending on the type of the study object:



- GetValue(Integer)
- GetValue(Date)
- GetValue(String)
- GetValue(Float)

Characteristic	Description
<b>Icon</b>	
<b>Availability</b>	All items.
<b>Return type and description</b>	<p><b>If the Empty property of the item is True</b>—Returns the replacement value (a parameter of the method).</p> <p><b>If the Empty property of the item is False</b>—Returns the value of the item.</p>
<b>Syntax</b>	<p>(If rule is created on a study design)</p> <pre>StudyEventRefName.FormRefName.ItemRefName.GetValue(replacementValue)</pre>

 **Note:**

If the item is a child of a repeating study object, method information is included in the rule expression to indicate the item instance to which you are referring.

**Parameters**

- **Parameter**—**replacementValue**.
- **Definition**—The replacement value for the item.
- **Data type**—Data type of the item.

**Example**

- If:
- Study event RefName is **Death**
  - Form RefName is **DeathForm**
  - Item RefName is **RelatedToDevice**
  - Codelist item RefName that you specify for the parameter is **RelatedCode3**
- ```
Death.DeathForm.RelatedToDevice.GetValue(Death.DeathForm.RelatedToDevice.RelatedCodes.RelatedCode3)
```


| Characteristic | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | <p>This method is useful for replacing a null value with a value that is usable for calculations, or for causing a rule to run even if a value has not been entered. For example, a BMI calculation typically does not run until the InForm user enters both a Height and Weight value for a subject. However, you can use this method to store a value of 0 in the BMI field if either Height or Weight is missing.</p> <p>Below is a typical rule expression for calculating BMI:</p> <pre>value = this.itmHeight.Value !=<br/>0 ?<br/>_CalculateBMI(this.itmWeight.Value<br/>, this.itmHeight.Value) : 0</pre> <p>This rule expression does not calculate the BMI until both values are present.</p> <p>Below is the same expression, except it uses GetValue</p> <pre>value =<br/>this.itmHeight.GetValue(0) == 0 ?<br/>_CalculateBMI(this.itmWeight.GetVa<br/>lue(0), this.itmHeight.Value) : 0</pre> <p>This expression calculates BMI upon submission of the form, regardless of whether the values for Height and Weight were entered.</p> <p>In both cases, the expression checks whether Height is 0. If Height is 0, a divide-by-zero error occurs.</p> |

## Methods for non-repeating study objects


In this section:

- [GetReviewStates](#)
- [HasReviewStates](#)
- [HasState\(Integer\)](#)
- [Example--Methods for non-repeating study objects](#)


## GetReviewStates

| Characteristic                     | Description                                                                                                                                                             |
|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icon</b>                        |                                                                                        |
| <b>Availability</b>                | All non-deleted forms.                                                                                                                                                  |
| <b>Return type and description</b> | Integer.<br>Returns the review stage that a form is in within a specified review state.                                                                                 |
| <b>Syntax</b>                      | <code>GetReviewState(Name of review state.Name of review stage)</code>                                                                                                  |
| <b>Parameters</b>                  | <ul style="list-style-type: none"> <li>• <b>Parameter—state.</b></li> <li>• <b>Definition—</b>Review state of the form.</li> <li>• <b>Data type—</b>Integer.</li> </ul> |
| <b>Example</b>                     | <p>The following example returns the review stage that the form is in within the QA Review review state:</p> <pre>GetReviewState(this.ReviewStates.QAR review)</pre>    |
| <b>Purpose</b>                     | Use this method to obtain the review stage that a form is in within a specified review state.                                                                           |

## HasReviewStates

| Characteristic                     | Description                                                                                                                                                             |
|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icon</b>                        |                                                                                      |
| <b>Availability</b>                | All non-deleted forms.                                                                                                                                                  |
| <b>Return type and description</b> | Boolean.<br>Returns true if the form is in the review stage specified in the parameter. Review stages appear in the folder for the form.                                |
| <b>Syntax</b>                      | <code>HasReviewState(Name of review stage)</code>                                                                                                                       |
| <b>Parameters</b>                  | <ul style="list-style-type: none"> <li>• <b>Parameter—stage.</b></li> <li>• <b>Definition—</b>Review stage of the form.</li> <li>• <b>Data type—</b>Integer.</li> </ul> |
| <b>Example</b>                     | <p>The following example returns True if the form is in the NeedsReview stage of the QA Review review state:</p> <pre>HasReviewState(this.QAReview.NeedsRe view)</pre>  |
| <b>Purpose</b>                     | Use this method to check whether a form is in a particular review stage.                                                                                                |

## HasState(Integer)

| Characteristic                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icon</b>                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Availability</b>                | All non-deleted forms.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Return type and description</b> | <p>Boolean.</p> <p>Returns True if the form is in the state that is specified in the parameter. Specify the state using one of the following constants. The constants appear on the Constants tab and have a data type of integer.</p> <ul style="list-style-type: none"> <li>• <b>Deleted</b>—(Available only for repeating forms) The repeating form instance has been deleted.</li> <li>• <b>Frozen</b>—The form is frozen.</li> <li>• <b>HasComment</b>—The form has a comment at the form level.</li> <li>• <b>HasData</b>—The form has data.</li> <li>• <b>HasMissingData</b>—The form has missing data.</li> <li>• <b>HasQueries</b>—The form has queries.</li> <li>• <b>Locked</b>—The form is locked.</li> <li>• <b>SdvComplete</b>—The form has been marked source verified.</li> <li>• <b>SdvPartial</b>—The form has been partially source verified.</li> <li>• <b>SdvReady</b>—The form has been marked ready for source verification.</li> <li>• <b>Signed</b>—The form has been signed.</li> <li>• <b>Skipped</b>—The form has been marked not completed with a form-level comment.</li> <li>• <b>Started</b>—The form was started with patient data, comments, or queries.</li> </ul> |
| <b>Syntax</b>                      | <code>HasState (NameOfConstant)</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Parameters</b>                  | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<b>formState</b>.</li> <li>• <b>Definition</b>—State of the form.</li> <li>• <b>Data type</b>—Integer.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Example</b>                     | <p>The following example returns True if the form is locked:</p> <pre>HasState(Constants.FormStates.Locked)</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Purpose</b>                     | Use this method to return the status of InForm forms.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

## Example--Methods for non-repeating study objects

The following example is for a rule named `rulDemoReviewDM` that runs when the Demographics form is submitted. If data is entered or modified for the Date of Birth item on the Demographics form, the rule sets the Data Management Review stage to the following:

| Current value | Value after the rule runs |
|---------------|---------------------------|
| Initial       | Follow Up                 |
| Follow Up     | Final                     |
| Final         | Follow Up                 |

### Rule summary

```

evaluate on Form Submission
value = !this.sctDM.itmDOB.Empty ?
this.HasReviewState(this.DataManagmentReview.InitialReview)
?
0
:
this.HasReviewState(this.DataManagmentReview.FollowUp)
?
1
:
this.HasReviewState(this.DataManagmentReview.FinalReview)
?
2
:
99
:
99
when value == 0
this.SetReviewState(DataManagmentReview.FollowUp, "")
when value == 1
this.SetReviewState(DataManagmentReview.FinalReview, "")
when value == 2
this.SetReviewState(DataManagmentReview.FollowUp, "")

```

# 3

## Functions

In this chapter:

- [Functions tab of the rule wizard](#)
- [Functions in rules and rule templates](#)
- [Dynamic prompts in the Expression workspace](#)
- [Viewing and editing a function](#)
- [Deleteing a function](#)
- [About predefined functions](#)
- [About user-defined functions](#)

### Functions tab of the rule wizard

A function is a reusable piece of code that extends the behavior of a rule. A function provides part or all of a rule expression. Without functions, a rule can support simple expressions with no control-flow statements or compound statements. Functions make possible or greatly simplify complex calculations or decision-making functionality in rule expressions. Functions are global. They do not have a scope and can be used in any rule created in a study or library as long as they are part of the study or library.

Functions allow users with different levels of programming experience to participate in rule-building:

- Experienced programmers with the ability to program in a .NET language (for example, C#) and create .NET assemblies build complex function code outside of the Oracle Central Designer application and import it into libraries or studies.
- Users who are not familiar with programming languages can create rules or rule templates that use functions.

The Oracle Central Designer application includes predefined functions (for example, several functions handle checking the components of date time fields).

You can use functions in a library or study:

- **In a library**—User-defined functions that you imported into a library are available after you publish them. You can use a function in a library when:
  - You are working in the library.
  - You are working in a study for which the library appears in the Library List.

#### Note:

If you are working in a study, you can use predefined functions that were created in the System Library when the System Library appears in the Library List.

- **In a study**—You can use user-defined functions that you imported into a study.

## Functions in rules and rule templates

To add functions to rule expressions, use one of the following:

- **For workflow rules**—Workflow Expression Editor dialog box.
- **For rule templates**—Edit Rule Template dialog box > Definition tab.
- **For data-entry rules**—Rule Wizard Expression tab > Functions tab.
- **For global conditions**—Edit Global Conditions dialog box > Functions tab.

You can use the functions that are registered in a study and in the libraries that appear in the Libraries List in the Study Editor. Both predefined and user-defined functions appear in these locations:

- In the Expression workplace. As you type, a list of the rule model components that you can use in the expression appears dynamically in the Expression workspace. When you select Functions and type a period, an alphabetical list of functions appears and is available for selection.
- In the Functions tab. Predefined functions are listed by category. For more information, see [Predefined functions in the system library](#).

You can incorporate functions in an expression using either location or by combining function information from both locations. Both dynamic expression prompts and the Functions tab appear in:

- Expression tab of the Rule Wizard.
- Workflow Expression Editor dialog box.
- Edit Global Conditions dialog box.
- Definition tab of the Edit Rule Template dialog box.

The list of functions is ordered in the following way:

1. Functions in the study.
2. Functions published in the first library on the Library List.
3. Functions published in the second library on the Library List (and so on).

## Dynamic prompts in the Expression workspace

Use either or both of the following methods:

- Select the function using the dynamic prompts that appear in the Expression workspace as you type.  
For more information, see [Dynamic prompts in the Expression workspace](#).
- Drag the function from the Functions tab to the Expression workspace. If parameters are associated with the function, the Invoke Function dialog box appears.

 **Note:**

The expression contains the name of the function, followed by parameters, which are enclosed by parentheses. Optionally, you can type the name of the function and its parameters.

## Viewing and editing a function

1. Perform one of the following:
  - In a study, select the **Study Information** Explorer bar.
  - In a library, select the **Library Information** Explorer bar.
2. Select **InForm**.
3. Select the **Functions** tab.
4. Perform one of the following:
  - Select or right-click the function, and click **Edit**.
  - Double-click the function.

The Edit Function dialog box appears.
5. Edit the function as necessary, and click **OK** when you are finished.

## Deleting a function

1. Perform one of the following:
  - In a study, select the **Study Information** Explorer bar.
  - In a library, select the **Library Information** Explorer bar.
2. Select **InForm**.
3. Select the **Functions** tab.
4. Select or right-click the function, and click **Delete**.

A confirmation dialog box appears.
5. Click **OK**.

## About predefined functions

A function is a reusable piece of code that extends the behavior of a rule. Many predefined functions are available by default in the System Library.

This section describes the predefined functions in the Oracle Central Designer application. The description of each function includes:

- Purpose of the function.
- Syntax, including parameters.
- Name, description, and data type of each parameter.
- Description of what the function returns.



- Notes about how the function works.

For more information, see:

- [Date time processing](#)
- [Exceptions](#)
- [Predefined functions in the system library](#)
- [\\_CalculateBMI](#)
- [\\_CalculateBSA](#)
- [\\_CalculateDateTime](#)
- [\\_CalculateWaistHipRatio](#)
- [\\_CheckPatientInitials](#)
- [\\_CompareDates](#)
- [\\_CompareDatesWithRange](#)
- [\\_Count](#)
- [\\_GetCurrentDate](#)
- [\\_GetDateDifference](#)
- [GetScreeningNumber](#)
- [GetSiteLocale](#)
- [GetSiteMnemonic](#)
- [GetSiteTime](#)
- [GetTrialName](#)
- [GetUserName](#)
- [GetSubjectNumber](#)
- [\\_IsValueGreaterThanArray](#)
- [\\_IsValueGreaterThanOrEqualToArray](#)
- [\\_IsValueInArray](#)
- [\\_IsValueLessThanArray](#)
- [\\_IsValueLessThanOrEqualToArray](#)
- [\\_NormalizeDate](#)
- [\\_NormalizeDateToMax](#)
- [Randomize](#)
- [\\_SaveToDb \(String, String\)](#)

## Date time processing

Functions that process date time items use a customized version of the .NET `DateTime` data type to enable the handling of incomplete date time fields. In the descriptions of functions, this customized data type is called `PFDatetime`.

To handle incomplete date time fields (where at least one part of the date time field has the value `UNK`, for `UNKNOWN`), date time fields are normalized. Normalization

makes sure that UNK date parts have no effect on date computations or comparisons by setting the unknown part of all dates that are involved in the computation to a neutral value. Normalization is performed by all functions that perform date time comparisons or calculations.

During normalization:

- The appropriate date time part value from a template is substituted for each UNK date time part or for a date time field that does not allow entry in all date time parts (for example, a date time field that accepts only dates and no times). The default template used in date time functions is 2000-01-01-12:00:00 (Year-Month-Day-Hour-Minute-Second). If you need to perform date time calculations that use a different template for normalization, you can use the `_NormalizeDate` functions to specify a custom template.
- When using date comparison functions, an additional normalization step occurs to account for the possibility that different date time parts could be marked UNK in the two date time fields. In this case, if a date time part in either date time field has the value UNK, the date time parts in both fields are assigned values from the template.

The following examples use the default template of 2000-01-01-12:00:00.

| Date time type                                          | Before normalization       | After normalization                        |
|---------------------------------------------------------|----------------------------|--------------------------------------------|
| One date time field                                     | 2007-04-UNK-16:20:UNK      | 2007-04-01-16:20:00                        |
| Two date time fields with same UNK date time part       | 07:22:UNK<br>15:14:UNK     | 07:22:00<br>15:14:00                       |
| Two date time fields with different UNK date time parts | 2006-12-UNK<br>2007-UNK-15 | 2006-01-01-12:00:00<br>2007-01-01-12:00:00 |

## Exceptions

When the execution of a function results in an exception:

- Processing of the rule stops.
- An Oracle Health Sciences InForm Server Error occurs in the Oracle Health Sciences InForm client application.
- An entry is posted in the Event log on the Oracle Health Sciences InForm server computer.

## Predefined functions in the system library

(Optional) Use sections to add and organize related content if another section heading is needed.

| Classification     | Function                                | Purpose                                                     |
|--------------------|-----------------------------------------|-------------------------------------------------------------|
| Clinical Functions | <a href="#">_CalculateBMI</a>           | Calculates the Body-Mass Index, based on height and weight. |
| –                  | <a href="#">_CalculateBSA</a>           | Calculates the body surface area.                           |
| –                  | <a href="#">_CalculateWaistHipRatio</a> | Calculates the waist-to-hip ratio.                          |

| Classification                                                                                                                                                                                         | Function                                          | Purpose                                                                                                                                                                             |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Clinical Validation Functions                                                                                                                                                                          | <a href="#">_CheckPatientInitials</a>             | Checks the format of the subject initials.                                                                                                                                          |
| Date Manipulation Functions                                                                                                                                                                            | <a href="#">_CalculateDateTime</a>                | Calculates a new date and time by adding an interval to or subtracting an interval from another date and time. The interval is based on a date part, for example, a number of days. |
| –                                                                                                                                                                                                      | <a href="#">_CompareDates</a>                     | Compares two date times to determine whether the first date time is less than, equal to, or greater than the second date time.                                                      |
| –                                                                                                                                                                                                      | <a href="#">_CompareDatesWithRange</a>            | Compares two date times to determine whether they are within a specified range of each other.                                                                                       |
| –                                                                                                                                                                                                      | <a href="#">_GetCurrentDate</a>                   | Returns the current date and time on the Oracle Health Sciences InForm server.                                                                                                      |
| –                                                                                                                                                                                                      | <a href="#">_GetDateDifference</a>                | Returns the number of units between two specified dates, based on the requested date part.                                                                                          |
| –                                                                                                                                                                                                      | <a href="#">_NormalizeDate</a>                    | Normalizes a date based on the specified template.                                                                                                                                  |
| –                                                                                                                                                                                                      | <a href="#">_NormalizeDateToMax</a>               | Normalizes UNKNOWN or EMPTY date time parts to the maximum values for those parts.                                                                                                  |
| Date/Time, Float, Integer, or Text Array Comparison Functions<br>These functions take different parameters depending on the data type of the value to which you are comparing the values in the array. | <a href="#">_Count</a>                            | Counts the number of occurrences of a specified value in an array.                                                                                                                  |
| –                                                                                                                                                                                                      | <a href="#">_IsValueGreaterThanArray</a>          | Checks whether a specified value is greater than all values in an array.                                                                                                            |
| –                                                                                                                                                                                                      | <a href="#">_IsValueGreaterThanOrEqualToArray</a> | Checks whether a specified value is greater than or equal to at least one of the values in an array.                                                                                |
| –                                                                                                                                                                                                      | <a href="#">_IsValueInArray</a>                   | Checks whether a specified value is equal to one of the values in an array.                                                                                                         |
| –                                                                                                                                                                                                      | <a href="#">_IsValueLessThanArray</a>             | Checks whether a specified value is less than all values in an array.                                                                                                               |

| Classification       | Function                                       | Purpose                                                                                                                                                                                         |
|----------------------|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| –                    | <a href="#">_IsValueLessThanOrEqualToArray</a> | Checks whether a specified value is less than or equal to at least one of the values in an array.                                                                                               |
| EDC                  | <a href="#">GetScreeningNumber</a>             | Returns the screening number of the subject in the Oracle Health Sciences InForm application.                                                                                                   |
| –                    | <a href="#">GetSiteLocale</a>                  | Returns the locale of the study site, such as en-US or ja-JP.                                                                                                                                   |
| –                    | <a href="#">GetSiteMnemonic</a>                | Returns the mnemonic (abbreviated site name, specified in the Admin area of the Oracle Health Sciences InForm application) for the site of the current subject.                                 |
| –                    | <a href="#">GetSiteTime</a>                    | Returns the current time at the data-entry site.                                                                                                                                                |
| –                    | <a href="#">GetSubjectNumber</a>               | Returns the number of the subject in the Oracle Health Sciences InForm application.                                                                                                             |
| –                    | <a href="#">GetTrialName</a>                   | Returns the name of the trial in the Oracle Health Sciences InForm application.                                                                                                                 |
| –                    | <a href="#">GetUserName</a>                    | Returns the name of user who is currently logged in to the Oracle Health Sciences InForm application.                                                                                           |
| –                    | <a href="#">Randomize</a>                      | Returns a drug kit number based on the randomization type and the sequence and drug kit numbers stored in the randomization source database on the Oracle Health Sciences InForm host computer. |
| Rule Event Functions | <a href="#">_SaveToDb (String, String)</a>     | Determines whether to send data to the Oracle Health Sciences InForm Publisher queue.                                                                                                           |

## \_CalculateBMI

Calculates the Body-Mass Index, based on height and weight.

### Syntax

`_CalculateBMI(height,weight)`

### Parameters

| Parameter | Definition                             | Data type |
|-----------|----------------------------------------|-----------|
| height    | Height of the subject, in centimeters. | Float     |

| Parameter | Definition                           | Data type |
|-----------|--------------------------------------|-----------|
| weight    | Weight of the subject, in kilograms. | Float     |

**Returns**

BMI measurement (Float).

**Notes**

- The Body-Mass index is calculated using the following formula:

$$\text{weight} / (\text{height} * \text{height})$$

- Height is assumed to be in centimeters, and weight is assumed to be in kilograms.

**Exceptions**

An exception of type `Argument Exception` is returned if the height measurement equals zero. This exception results in:

- Failure of the rule execution.
- An Oracle Health Sciences InForm Server Error in the Oracle Health Sciences InForm client application.
- An entry in the Event log on the Oracle Health Sciences InForm server computer.

**Example**

The following rule is created at the form level on the Vital Signs form, which has items Weight and BMI. The rule also refers to the item Ht in a mapping called RulesLS.

**Note:**

Oracle recommends that you include rule logic to clear the calculated value if a referenced item value is cleared. For example, if a rule calculates BMI based on the values entered in the Height and Weight items, the rule should clear the BMI value if the data in the Height item is deleted. For an example of the appropriate rule logic, see [Sample data-entry rules that use methods](#).

```
evaluate on Form Submission
    value = _CalculateBMI (RulesLS.DSRules.Ht.Value, this.Weight.Value)
always
    set this.BMI.Value = value
```

## \_CalculateBSA

Calculates the body surface area.

**Syntax**

`_CalculateBSA(height,weight)`

**Parameters**

| Parameter | Definition                             | Data type |
|-----------|----------------------------------------|-----------|
| height    | Height of the subject, in centimeters. | Float     |
| weight    | Weight of the subject, in kilograms.   | Float     |

**Returns**

Body surface area (Float).

**Notes**

- Calculates the BSA using the following formula:

```
(Float) (0.007184 * Math.Pow(height,0.725) * Math.Pow(weight,0.425));
```

(Math.Pow is a method that raises a value to the specified power.)

- Height is assumed to be in centimeters, and weight is assumed to be in kilograms.
- Oracle recommends that you include rule logic to clear the calculated value if a referenced item value is cleared. For example, if a rule calculates BMI based on the values entered in the Height and Weight items, the rule should clear the BMI value if the data in the Height item is deleted. For more information on appropriate rule logic, see [Sample data-entry rules that use methods](#).

## \_CalculateDateTime

Calculates a new date and time by adding an interval to or subtracting an interval from another date and time. The interval is based on a date part, for example, a number of days.

**Syntax**

```
_CalculateDateTime(date,interval,datePart)
```

**Parameters**

| Parameter | Definition                                                                 | Data type  |
|-----------|----------------------------------------------------------------------------|------------|
| date      | Original date and time.                                                    | PFDatetime |
| interval  | Time interval to add to or subtract from the original date time.           | Integer    |
| datePart  | Date part to which the value specified in the interval parameter is added. | Integer    |

**Returns**

New date time (PFDatetime).

**Notes**

- Date intervals are calculated in the following way:

- Years are measured in 365- or 366-day increments, depending on the number of calendar days in the year:

 **Note:**

If the time period includes a February month with 29 days, then 366 days equal 1 year. Otherwise, 365 days equal 1 year.

- \* Fewer than 365 (or, as mentioned above, 366 days)—0 years.
  - \* 365 (or 366) to 729 (or 730) days—1 year
  - \* 729 (or 730) to 1094 (or 1095) days—2 years.
- Months are measured in 28-, 29-, 30-, or 31-day increments, depending on the number of days in the month.  
For example, if a start date is in a month with 31 days, then 31 days from the start date is one month. If the date range goes from February until September, then each month's number of days is used for the calculation. For example, 28 (or 29, if February has 29 days) days from the start date in February is one month; 31 days from the date in March is another month; 30 days from the date in April is another month; and so on.  
For increments that are fewer than 28, 29, 30, or 31 days, depending on the month, the range is 0 months.
  - Days are measured in 24-hour increments. For example:
    - \* Fewer than 24 hours—0 days.
    - \* 24 to 47 hours—1 day.
    - \* 48 to 71 hours—2 days.
  - The new date time is based on the sum of the original date time and the interval.
  - Before the interval is added, the date is normalized. For more information, see [Date time processing](#).
  - Valid dateParts are taken from the DateTimeParts enumerator, which returns an integer constant. (An enumerator is a variable type that represents a restricted list of values.) To reference a date part, type **DateTimeParts.datePart**, where *datePart* is one of the following:
    - Years
    - Months
    - Days
    - Hours
    - Minutes
    - Seconds

### Example

This example sets a date for follow-up that is four weeks after the initial date entered. Both items are on the same form, so the rule is created at the form level. There is no enumeration for weeks in DateTimeParts, so the rule uses twenty-eight days instead of four weeks.

 **Note:**

Oracle recommends that you include rule logic to clear the calculated value if a referenced item value is cleared. For example, if a rule calculates BMI based on the values entered in the Height and Weight items, the rule should clear the BMI value if the data in the Height item is deleted. For an example of the appropriate rule logic, see [Sample data-entry rules that use methods](#).

```
evaluate on Form Submission
    value = _CalculateDateTime(this.InitialExamDate.Value, 28,
    DateTimeParts.Days)
always
    set this.FollowupExamDate.Value = value
```

## \_CalculateWaistHipRatio

Calculates the waist-to-hip ratio.

### Syntax

```
_CalculateWaistHipRatio(waist,hip)
```

### Parameters

| Parameter | Definition                        | Data type |
|-----------|-----------------------------------|-----------|
| waist     | Waist measurement of the subject. | Float     |
| hip       | Hip measurement of the subject.   | Float     |

### Returns

Waist-to-hip ratio (Float).

### Notes

- Waist-to-hip ratio is calculated as *waist/hip*.
- Both measurements must be in the same units. For example, if one measurement is in centimeters, the other must be, as well.

### Exceptions

An exception of type Argument Exception is returned if the hip measurement equals zero. This exception results in:

- Failure of the rule execution.
- An Oracle Health Sciences InForm Server Error in the Oracle Health Sciences InForm client application.
- An entry in the Event log on the Oracle Health Sciences InForm server computer.

### Example

The following rule is attached to the Vital Signs form and does not need any objects from outside the form. The form includes a waistCircumference item to record the measurement of



waist circumference and a hipCircumference item to record the measurement of hip circumference. Both allow the value to be recorded in centimeters or inches, but both values are normalized to centimeters.

 **Note:**

Oracle recommends that you include rule logic to clear the calculated value if a referenced item value is cleared. For example, if a rule calculates BMI based on the values entered in the Height and Weight items, the rule should clear the BMI value if the data in the Height item is deleted. For an example of the appropriate rule logic, see [Sample data-entry rules that use methods](#).

```
evaluate on Form Submission
    value = _CalculateWaistHipRatio(this.waistCircumference.Value,
    this.hipCircumference.Value)
always
    set this.whRatio.Value = value
```

## \_CheckPatientInitials

Checks the format of the subject initials.

### Syntax

`_CheckPatientInitials(initials)`

### Parameters

| Parameter | Definition        | Data type |
|-----------|-------------------|-----------|
| initials  | Subject initials. | Text      |

### Returns

True or False (Boolean), indicating whether the subject initials are in the correct format.

### Notes

The entered string is validated to make sure that it consists of three letters or two letters with a dash in the middle to indicate that no middle name was provided.

## \_CompareDates

Compares two date times to determine whether the first date time is less than, equal to, or greater than the second date time.

### Syntax

`_CompareDates(date1,date2)`

### Parameters

| Parameter | Definition          | Data type   |
|-----------|---------------------|-------------|
| date1     | First date to use.  | PFDDateTime |
| date2     | Second date to use. | PFDDateTime |

**Returns**

One of the following (Integer):

- **-1**—If date1 is less than date2.
- **0**—If date1 equals date2.
- **1**—If date1 is greater than date2.

**Notes**

Both dates are normalized before comparison. For more information, see [Date time processing](#).

**Example**

This example of a constraint rule uses both the `GetSiteTime` and `_CompareDates` functions to check whether the date entered is in the future. It is attached at the item level to a DOV item.

```
evaluate on Form Submission
  value = _CompareDates(this.Value, GetSiteTime())
when value == 1
  issue query: Date cannot be in the future compared to System Date
```

You could also write the rule to evaluate to true or false instead of to a numeric value:

```
evaluate on Form Submission
  value = _CompareDates(this.Value, GetSiteTime()) == 1
when value is true
  issue query: Date cannot be in the future compared to System Date
```

## `_CompareDatesWithRange`

Compares two date times to determine whether they are within a specified range of each other.

**Syntax**

```
_CompareDatesWithRange(date1,date2,datePart,rangeMin,rangeMax)
```

**Parameters**

| Parameter | Definition          | Data type   |
|-----------|---------------------|-------------|
| date1     | First date to use.  | PFDDateTime |
| date2     | Second date to use. | PFDDateTime |

| Parameter | Definition                                                                        | Data type |
|-----------|-----------------------------------------------------------------------------------|-----------|
| datePart  | Date part used to calculate the interval between date1 and date2.                 | Integer   |
| rangeMin  | Minimum number in the range to compare with the interval between date1 and date2. | Integer   |
| rangeMax  | Maximum number in the range to compare with the interval between date1 and date2. | Integer   |

### Returns

True or False (Boolean), indicating whether the first date time is within the specified range of the second date time.

### Notes

- Date intervals are calculated in the following way:
  - Years are measured in 365- or 366-day increments, depending on the number of calendar days in the year.

#### Note:

If the time period includes a February month with 29 days, then 366 days equal 1 year. Otherwise, 365 days equal 1 year.

- \* Fewer than 365 (or, as mentioned above, 366 days)—0 years.
- \* 365 (or 366) to 729 (or 730) days—1 year.
- \* 729 (or 730) to 1094 (or 1095) days—2 years.
- Months are measured in 28-, 29-, 30-, or 31-day increments, depending on the number of days in the month.  
For example, if a start date is in a month with 31 days, then 31 days from the start date is one month. If the date range goes from February until September, then each month's number of days is used for the calculation. For example, 28 (or 29, if February has 29 days) days from the start date in February is one month; 31 days from the date in March is another month; 30 days from the date in April is another month; and so on.  
For increments that are fewer than 28, 29, 30, or 31 days, depending on the month, the range is 0 months.
- Days are measured in 24-hour increments. For example:
  - \* Fewer than 24 hours—0 days.
  - \* 24 to 47 hours—1 day.
  - \* 48 to 71 hours—2 days.
- The method compares the interval between the specified datePart in each date time and determines whether the interval is within the range indicated by the rangeMin and rangeMax parameters. This method is useful when determining

whether the age of a subject is within a specific age range. The minimum and maximum ranges are inclusive.

- Both dates are normalized before comparison. For more information, see [Date time processing](#).
- Valid dateParts are taken from the DateTimeParts enumerator, which returns an integer constant. (An enumerator is a variable type that represents a restricted list of values.) To reference a date part, type **DateTimeParts.datePart**, where *datePart* is one of the following:
  - Years
  - Months
  - Days
  - Hours
  - Minutes
  - Seconds

### Example

This example checks whether a reported adverse event lasts for more than six months. The rule is defined at the form level and checks the StartDate and EndDate items in that form. This rule creates a query when an adverse event lasts more than six months.

```
evaluate on Form Submission
  value = _CompareDatesWithRange(this.StartDate.Value, this.EndDate.Value,
    DateTimeParts.Months, 0, 6)
when value is false
  issue query on this.EndDate: Adverse Event lasted more than six months.
```

## \_Count

Counts the number of occurrences of a specified value in an array. This function takes different parameters depending on the data type of the value you are counting.

For more information on these different parameters, see:

- [\\_Count\(String, Array, Boolean\)](#)
- [\\_Count\(Date, Array\)](#)
- [\\_Count\(Integer, Array\)](#)
- [\\_Count\(Float, Array\)](#)

### \_Count(String, Array, Boolean)

Counts the number of occurrences of a string value in an array.

#### Syntax

```
_Count(valueToCount,valueList,caseSensitive)
```

#### Parameters

| Parameter     | Definition                                                                                                                                                                                                                                                          | Data type              |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| valueToCount  | Value to count                                                                                                                                                                                                                                                      | String                 |
| valueList     | Array of values to search                                                                                                                                                                                                                                           | Array of String values |
| caseSensitive | Case sensitive search flag, True or False. If the value is True, the function takes case into consideration when searching for strings that match the value to count. For example, if caseSensitive is True, ABC is not counted as an occurrence of the string abc. | Boolean                |

**Returns**

The number of occurrences of a string value in an array.

**\_Count(Date, Array)**

Counts the number of occurrences of a date time value in an array.

**Syntax**

*\_Count(valueToCount,valueList)*

**Parameters**

| Parameter    | Definition                | Data type                  |
|--------------|---------------------------|----------------------------|
| valueToCount | Value to count            | PFDatetime                 |
| valueList    | Array of values to search | Array of PFDatetime values |

**Returns**

The number of occurrences of a given date time value in an array.

**\_Count(Integer, Array)**

Counts the number of occurrences of an integer value in an array.

**Syntax**

*\_Count(valueToCount,valueList)*

**Parameters**

| Parameter    | Definition                | Data type               |
|--------------|---------------------------|-------------------------|
| valueToCount | Value to count            | Integer                 |
| valueList    | Array of values to search | Array of Integer values |

**Returns**

The number of occurrences of a given integer value in an array.

## `_Count`(Float, Array)

Counts the number of occurrences of a float value in an array.

### Syntax

```
_Count(valueToCount,valueList)
```

### Parameters

| Parameter    | Definition                | Data type             |
|--------------|---------------------------|-----------------------|
| valueToCount | Value to count            | Float                 |
| valueList    | Array of values to search | Array of Float values |

### Returns

The number of occurrences of a given float value in an array.

## `_GetCurrentDate`

Returns the current date and time on the InForm server.

### Syntax

```
_GetCurrentDate()
```

### Returns

Current date and time at the location of the InForm server computer.

### Notes

To get the current time in the time zone of the data-entry site, use [\\_GetCurrentDate](#).

## `_GetDateDifference`

Returns the number of units between two specified dates, based on the requested date part.

### Syntax

```
_GetDateDifference(date1,date2,units)
```

### Parameters

| Parameter | Definition                                                                                                                      | Data type   |
|-----------|---------------------------------------------------------------------------------------------------------------------------------|-------------|
| date1     | First date to use. date1 is subtracted from date2.                                                                              | PFDDateTime |
| date2     | Second date to use.                                                                                                             | PFDDateTime |
| units     | Unit to use when computing the difference. Units are taken from DateTimeParts:<br>Years, Months, Days, Hours, Minutes, Seconds. | Integer     |

## Returns

A positive, zero, or negative interval length depending on whether the first date is earlier than, equal to, or later than the second date. Unknown date parts are normalized. Results are rounded down.

## Notes

- `GetDateDifference` returns positive value if `date1` is before `date2` and returns a negative value when `date 2` is before `date 1`.
- Date intervals are calculated in the following way:
  - Years are measured in 365- or 366-day increments, depending on the number of calendar days in the year.

 **Note:**

If the time period includes a February month with 29 days, then 366 days equal 1 year. Otherwise, 365 days equal 1 year.

- \* Fewer than 365 (or, as mentioned above, 366 days)—0 years.
- \* 365 (or 366) to 729 (or 730) days—1 year.
- \* 729 (or 730) to 1094 (or 1095) days—2 years.
- Months are measured in 28-, 29-, 30-, or 31-day increments, depending on the number of days in the month.  
For example, if a start date is in a month with 31 days, then 31 days from the start date is one month. If the date range goes from February until September, then each month's number of days is used for the calculation. For example, 28 (or 29, if February has 29 days) days from the start date in February is one month; 31 days from the date in March is another month; 30 days from the date in April is another month; and so on.  
For increments that are fewer than 28, 29, 30, or 31 days, depending on the month, the range is 0 months.
- Days are measured in 24-hour increments. For example:
  - \* Fewer than 24 hours—0 days.
  - \* 24 to 47 hours—1 day.
  - \* 48 to 71 hours—2 days.
- Valid `dateParts` are taken from the `DateTimeParts` enumerator, which returns an integer constant. (An enumerator is a variable type that represents a restricted list of values.) To reference a date part, type **`DateTimeParts.datePart`**, where `datePart` is one of the following:
  - Years
  - Months
  - Days
  - Hours
  - Minutes
  - Seconds

- Dates are normalized before the number of units is calculated. For more information, see [Date time processing](#).
- The entire date is used in the calculation, not only the specified date part. Results are rounded down to the nearest integer. For example, if `_GetDateDifference` runs with the following data, it returns 0, not 1:  
`_GetDateDifference (new DateTime(2007,1,10,0,0,0), new DateTime (2007,2,1,0,0,0), DateTimeParts.Months)`

### Example

The `_GetDateDifference` function is used to determine how long an adverse event lasts and to store the result in an item. This rule is attached to the AE form, which contains the `OnsetDate` and `EndDate` items, as well as an `AEDuration` item to store the calculated value.

```
evaluate on Form Submission
    value = _GetDateDifference(this.OnsetDate.Value, this.EndDate.Value,
    DateTimeParts.Days)
always
    set this.AEDuration.Value = value
```

If `OnsetDate` is January 1, 2008, and `EndDate` is January 6, 2008, the function returns 5.

## GetScreeningNumber

Returns the screening number of the subject in the Oracle Health Sciences InForm application.

### Syntax

```
GetScreeningNumber()
```

### Returns

Screening number of the subject in the Oracle Health Sciences InForm application.

## GetSiteLocale

Returns the locale of the study site, such as en-US or ja-JP. You select locales in the Study Editor.

### Syntax

```
GetSiteLocale()
```

### Returns

Locale of the study site.

### Notes

You can use the site locale to format a date time or floating point number. For example, you can write a user-defined function that uses the locale that is returned by this function and the date time value that a user enters in the Oracle Health Sciences InForm application to format the date time value according to the standard formatting of the locale.



## GetSiteMnemonic

Returns the mnemonic (abbreviated site name, specified in the Admin area of the Oracle Health Sciences InForm application) for the site of the current subject.

### Syntax

```
GetSiteMnemonic()
```

### Returns

Mnemonic of the site of the current patient.

## GetSiteTime

Returns the current time at the data-entry site.

### Syntax

```
GetSiteTime()
```

### Returns

Current time at the data-entry site.

### Notes

The function returns the current time in the time zone of the data-entry site, not the time zone of the location of the Oracle Health Sciences InForm server. To get the current time at the Oracle Health Sciences InForm server computer, use [\\_GetCurrentDate](#).

### Example

This example of a constraint rule uses both the `GetSiteTime` and `_CompareDates` functions to check whether the date entered is in the future. It is attached at the item level to a DOV item.

```
evaluate on Form Submission
  value = _CompareDates(this.Value, GetSiteTime())
when value == 1
  issue query: Date cannot be in the future compared to System Date
```

You could also write the rule to evaluate to true or false instead of to a numeric value:

```
evaluate on Form Submission
  value = _CompareDates(this.Value, GetSiteTime()) == 1
when value is true
  issue query: Date cannot be in the future compared to System Date
```

## GetTrialName

Returns the name of the trial in the Oracle Health Sciences InForm application.

### Syntax

GetTrialName()

**Returns**

Name of the trial in the Oracle Health Sciences InForm application.

**Notes**

This function is particularly useful for populating email parameters, writing query text, and writing user-defined functions. For example, you can display the study name in the query text of a rule or in the email subject and body.

## GetUserName

Returns the name of user who is currently logged in to the Oracle Health Sciences InForm application.

**Syntax**

GetUserName()

**Returns**

Name of the currently logged-on user in the Oracle Health Sciences InForm application.

**Notes**

The **GetUserName()** function is supported in the following Oracle Health Sciences InForm releases:

- Release 4.6 SP2 and above, except release 4.7.
- Release 5.0 and above.



**Note:**

If a rule containing the function runs in an unsupported release, the rule returns an empty string.

## GetSubjectNumber

Returns the number of the subject in the Oracle Health Sciences InForm application.

**Syntax**

GetSubjectNumber()

**Returns**

Number of the subject in the Oracle Health Sciences InForm application.

**Notes**

- This function calls the GetPatientNumber() method in the Oracle Health Sciences InForm application. The number is assigned to the subject during enrollment.
- In the Rule Test Cases dialog box, rules that use this function return TestSubjectNumber.

## `_IsValueGreaterThanArray`

Checks whether a specified value is greater than all values in an array. This function takes different parameters depending on the data type of the value to which you are comparing the values in the array.

For more information on these different data type parameters, see:

- [\\_IsValueGreaterThanArray \(PFDatetime, Array\)](#)
- [\\_IsValueGreaterThanArray \(Float, Array\)](#)
- [\\_IsValueGreaterThanArray \(Integer, Array\)](#)

### `_IsValueGreaterThanArray (PFDatetime, Array)`

Checks if a date time value is greater than all of the date times in a list.

#### Syntax

```
_IsValueGreaterThanArray(valueToTest,valueList)
```

#### Parameters

| Parameter                | Definition                                                                                                        | Data type                  |
|--------------------------|-------------------------------------------------------------------------------------------------------------------|----------------------------|
| <code>valueToTest</code> | Date time value with which to compare the values in the list.                                                     | PFDatetime                 |
| <code>valueList</code>   | Array of date time values against which to compare the value specified in the <code>valueToTest</code> parameter. | Array of PFDatetime values |

#### Returns

True or False (Boolean), indicating whether the date time to test is greater than all of the date times in the array.

#### Notes

- The `valueToTest` value is compared to each element of the array. If the value is greater than every element of the array, the method returns True. If at least one element is less than or equal to the value, the method returns False.
- All comparisons are done using the `_CompareDates` method and are normalized. For more information, see [Date time processing](#).

### `_IsValueGreaterThanArray (Float, Array)`

Checks if a float value is greater than all of the date times in an array.

#### Syntax

```
_IsValueGreaterThanArray(valueToTest,valueList)
```

#### Parameters

| Parameter   | Definition                                                                                       | Data type             |
|-------------|--------------------------------------------------------------------------------------------------|-----------------------|
| valueToTest | Float value with which to compare the values in the array.                                       | Float                 |
| valueList   | Array of float values against which to compare the value specified in the valueToTest parameter. | Array of Float values |

**Returns**

True or False (Boolean), indicating whether the float value to test is greater than all of the elements in the array.

**Notes**

The valueToTest value is compared to each element of the array. If the value is greater than every element of the array, the method returns True. If at least one element is less than or equal to the value, the method returns False.

## \_IsValueGreaterThanArray (Integer, Array)

Checks if an integer value is greater than all of the date times in an array.

**Syntax**

`_IsValueGreaterThanArray(valueToTest,valueList)`

**Parameters**

| Parameter   | Definition                                                                                         | Data type               |
|-------------|----------------------------------------------------------------------------------------------------|-------------------------|
| valueToTest | Integer value with which to compare the values in the array.                                       | Integer                 |
| valueList   | Array of integer values against which to compare the value specified in the valueToTest parameter. | Array of Integer values |

**Returns**

True or False (Boolean), indicating whether the integer value to test is greater than all of the elements in the list.

**Notes**

The valueToTest value is compared to each element of the array. If the value is greater than every element of the array, the method returns True. If at least one element is less than or equal to the value, the method returns False.

## \_IsValueGreaterThanOrEqualToArray

Checks whether a specified value is greater than or equal to at least one of the values in an array. This function takes different parameters depending on the data type of the value to which you are comparing the values in the array.

For more information on the different data type parameters, see:

- [\\_IsValueGreaterThanOrEqualToArray \(PFDateTime, Array\)](#)
- [\\_IsValueGreaterThanOrEqualToArray \(Float, Array\)](#)
- [\\_IsValueGreaterThanOrEqualToArray \(Integer, Array\)](#)

## [\\_IsValueGreaterThanOrEqualToArray \(PFDateTime, Array\)](#)

Checks if a date time value is greater than or equal to all of the date times in an array.

### Syntax

```
_IsValueGreaterThanOrEqualToArray(valueToTest,valueList)
```

### Parameters

| Parameter   | Definition                                                                                           | Data type                  |
|-------------|------------------------------------------------------------------------------------------------------|----------------------------|
| valueToTest | Date time value with which to compare the values in the array.                                       | PFDateTime                 |
| valueList   | Array of date time values against which to compare the value specified in the valueToTest parameter. | Array of PFDateTime values |

### Returns

True or False (Boolean), indicating whether the date time to test is greater than or equal to all of the date times in the array.

#### Note:

- The valueToTest value is compared to each element of the array. If the value is greater than or equal to every element of the array, the method returns True. If at least one element is less than the value, the method returns False.
- All comparisons are done using the `_CompareDates` method and are normalized. For more information, see [Date time processing](#).

### Example

This example checks whether the date of termination is greater than or equal to all visit dates for the patient and issues a query if the date of termination is less than any of the patient visit dates. The rule is created at the item level and uses the mapping "RulesLS" and the data set and series within that mapping. All of the visit dates in a data series called VisitDates within the RulesLS mapping.

```
evaluate on Form Submission
  value = _IsValueGreaterThanOrEqualToArray(this.Value,
RulesLS.DateCollection.VisitDates.Values)
when value is false
  issue query: Termination date must be later than any visit date.
```

## `_IsValueGreaterThanOrEqualToArray (Float, Array)`

Checks if a float value is greater than or equal to at least one of the values in an array.

### Syntax

```
_IsValueGreaterThanOrEqualToArray(valueToTest,valueList)
```

### Parameters

| Parameter   | Definition                                                                                       | Data type             |
|-------------|--------------------------------------------------------------------------------------------------|-----------------------|
| valueToTest | Float value with which to compare the values in the array.                                       | Float                 |
| valueList   | Array of float values against which to compare the value specified in the valueToTest parameter. | Array of Float values |

### Returns

True or False (Boolean), indicating whether the float value to test is greater than all of the elements in the array.

### Notes

The valueToTest value is compared to each element of the array. If the value is greater than or equal to every element of the array, the method returns True. If at least one element is less than the value, the method returns False.

## `_IsValueGreaterThanOrEqualToArray (Integer, Array)`

Checks if a specified integer value is greater than or equal to at least one of the integer values in an array.

### Syntax

```
_IsValueGreaterThanOrEqualToArray(valueToTest,valueList)
```

### Parameters

| Parameter   | Definition                                                                                         | Data type               |
|-------------|----------------------------------------------------------------------------------------------------|-------------------------|
| valueToTest | Integer value with which to compare the values in the array.                                       | Integer                 |
| valueList   | Array of integer values against which to compare the value specified in the valueToTest parameter. | Array of Integer values |

### Returns

True or False (Boolean), indicating whether the integer value to test is greater than all of the elements in the array.

### Notes

The `valueToTest` value is compared to each element of the array. If the value is greater than or equal to every element of the array, the method returns `True`. If at least one element is less than the value, the method returns `False`.

## `_IsValueInArray`

Checks whether a specified value is equal to one of the values in an array. This function takes different parameters depending on the data type of the value to which you are comparing the values in the array.

For more information on the different data type parameters, see:

- [\\_IsValueInArray \(PFDatetime, Array\)](#)
- [\\_IsValueInArray \(Float, Array\)](#)
- [\\_IsValueInArray \(Integer, Array\)](#)
- [\\_IsValueInArray \(Text, Array\)](#)

### `_IsValueInArray (PFDatetime, Array)`

Checks if a specified date time value is equal to one of the date time values in an array.

#### Syntax

```
_IsValueInArray(valueToTest,valueList)
```

#### Parameters

| Parameter                | Definition                                                                                                        | Data type                               |
|--------------------------|-------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
| <code>valueToTest</code> | Date time value with which to compare the values in the array.                                                    | <code>PFDatetime</code>                 |
| <code>valueList</code>   | Array of date time values against which to compare the value specified in the <code>valueToTest</code> parameter. | Array of <code>PFDatetime</code> values |

#### Returns

`True` or `False` (Boolean), indicating whether the date time value to test is in the array of date times.

#### Notes

- The `valueToTest` value is compared to each element of the array. If the value is equal to at least one element of the array, the method returns `True`. Otherwise, the method returns `False`.
- All comparisons are done using the `_CompareDates` method and are normalized. For more information, see [Date time processing](#).

### `_IsValueInArray (Float, Array)`

Checks if a specified float value is equal to one of the float values in an array.

**Syntax**

```
_IsValueInArray(valueToTest,valueList)
```

**Parameters**

| Parameter   | Definition                                                                                       | Data type             |
|-------------|--------------------------------------------------------------------------------------------------|-----------------------|
| valueToTest | Float value with which to compare the values in the array.                                       | Float                 |
| valueList   | Array of float values against which to compare the value specified in the valueToTest parameter. | Array of Float values |

**Returns**

True or False (Boolean), indicating whether the float value to test is in the array of float values.

**Notes**

The valueToTest value is compared to each element of the array. If the value is equal to at least one element of the array, the method returns True. Otherwise, the method returns False.

**\_IsValueInArray (Integer, Array)**

Checks if a specified integer value is equal to one of the integer values in an array.

**Syntax**

```
_IsValueInArray(valueToTest,valueList)
```

**Parameters**

| Parameter   | Definition                                                                                         | Data type               |
|-------------|----------------------------------------------------------------------------------------------------|-------------------------|
| valueToTest | Integer value with which to compare the values in the array.                                       | Integer                 |
| valueList   | Array of integer values against which to compare the value specified in the valueToTest parameter. | Array of Integer values |

**Returns**

True or False (Boolean), indicating whether the integer value to test is in the array of integer values.

**Notes**

The valueToTest value is compared to each element of the array. If the value is equal to at least one element of the array, the method returns True. Otherwise, the method returns False.

**Example**



In this example of a global condition, when a user indicates in the AE form that a patient died as the result of an adverse event, a form called Death is enabled. Because the AE form is a common form that does not have reference to specific visits, the Outcome field from the AE form is collected in a mapping.

```
_IsValueInArray("Death", RulesLS.DSRules.Outcomes.Values)
```

This condition returns true or false. When the condition is applied to the Death form, the Death form appears when the expression is true.

## `_IsValueInArray` (Text, Array)

Checks if a specified text value is equal to one of the text values in an array.

### Syntax

```
_IsValueInArray(valueToTest,valueList)
```

### Parameters

| Parameter   | Definition                                                                                      | Data type            |
|-------------|-------------------------------------------------------------------------------------------------|----------------------|
| valueToTest | Text value with which to compare the values in the array.                                       | Text                 |
| valueList   | Array of text values against which to compare the value specified in the valueToTest parameter. | Array of Text values |

### Returns

True or False (Boolean), indicating whether the text value to test is in the array of text values.

### Notes

The valueToTest value is compared to each element of the array. If the value is equal to at least one element of the array, the method returns True. Otherwise, the method returns False.

## `_IsValueLessThanArray`

Checks whether a specified value is less than all values in an array. This function takes different parameters depending on the data type of the value to which you are comparing the values in the array.

For more information on the different data type parameters, see:

- [\\_IsValueLessThanArray \(PFDateTime, Array\)](#)
- [\\_IsValueLessThanArray \(Float, Array\)](#)
- [\\_IsValueLessThanArray \(Integer, Array\)](#)

## `_IsValueLessThanArray` (PFDateTime, Array)

Checks if a specified date time value is less than all date time values in an array.

### Syntax

```
_IsValueLessThanArray(valueToTest,valueList)
```

### Parameters

| Parameter                | Definition                                                                                                        | Data type                  |
|--------------------------|-------------------------------------------------------------------------------------------------------------------|----------------------------|
| <code>valueToTest</code> | Date time value with which to compare the values in the array.                                                    | PFDateTime                 |
| <code>valueList</code>   | Array of date time values against which to compare the value specified in the <code>valueToTest</code> parameter. | Array of PFDateTime values |

### Returns

True or False (Boolean), indicating whether the specified date time value is less than all date time values in an array.

### Notes

- The `valueToTest` value is compared to each element of the array. If the value is less than every element of the array, the method returns True. If at least one element is greater than or equal to the value, the method returns False.
- All comparisons are done using the `_CompareDates` method and are normalized. For more information, see [Date time processing](#).

## `_IsValueLessThanArray` (Float, Array)

Checks if a specified float value is less than all float values in an array.

### Syntax

```
_IsValueLessThanArray(valueToTest,valueList)
```

### Parameters

| Parameter                | Definition                                                                                                    | Data type             |
|--------------------------|---------------------------------------------------------------------------------------------------------------|-----------------------|
| <code>valueToTest</code> | Float value with which to compare the values in the array.                                                    | Float                 |
| <code>valueList</code>   | Array of float values against which to compare the value specified in the <code>valueToTest</code> parameter. | Array of Float values |

### Returns

True or False (Boolean), indicating whether the specified float value is less than all float values in an array.

### Notes

The `valueToTest` value is compared to each element of the array. If the value is less than every element of the array, the method returns `True`. If at least one element is greater than or equal to the value, the method returns `False`.

## `_IsValueLessThanArray` (Integer, Array)

Checks if a specified integer value is less than all integer values in an array.

### Syntax

```
_IsValueLessThanArray(valueToTest,valueList)
```

### Parameters

| Parameter                | Definition                                                                                                      | Data type               |
|--------------------------|-----------------------------------------------------------------------------------------------------------------|-------------------------|
| <code>valueToTest</code> | Integer value with which to compare the values in the array.                                                    | Integer                 |
| <code>valueList</code>   | Array of integer values against which to compare the value specified in the <code>valueToTest</code> parameter. | Array of Integer values |

### Returns

`True` or `False` (Boolean), indicating whether the specified integer value is less than all integer values in an array.

### Notes

The `valueToTest` value is compared to each element of the array. If the value is less than every element of the array, the method returns `True`. If at least one element is greater than or equal to the value, the method returns `False`.

## `_IsValueLessThanOrEqualToArray`

Checks whether a specified value is less than or equal to at least one of the values in an array. This function takes different parameters depending on the data type of the value to which you are comparing the values in the array.

For more information on different data type parameters, see:

- [\\_IsValueLessThanOrEqualToArray \(PFDateTime, Array\)](#)
- [\\_IsValueLessThanOrEqualToArray \(Float, Array\)](#)
- [\\_IsValueLessThanOrEqualToArray \(Integer, Array\)](#)

## `_IsValueLessThanOrEqualToArray` (PFDateTime, Array)

Checks if a specified date time value is less than or equal to at least one of the date time values in an array.

### Syntax

```
_IsValueLessThanOrEqualToArray(valueToTest,valueList)
```

### Parameters

| Parameter   | Definition                                                                                           | Data type                  |
|-------------|------------------------------------------------------------------------------------------------------|----------------------------|
| valueToTest | Date time value with which to compare the values in the array.                                       | PFDateTime                 |
| valueList   | Array of date time values against which to compare the value specified in the valueToTest parameter. | Array of PFDateTime values |

**Returns**

True or False (Boolean), indicating whether the specified date time value is less than or equal to at least one of the date time values in an array.

**Notes**

- The valueToTest value is compared to each element of the array. If the value is less than or equal to every element of the array, the method returns True. If at least one element is greater than the value, the method returns False.
- All comparisons are done using the `_CompareDates` method and are normalized. For more information, see [Date time processing](#).

**`_IsValueLessThanOrEqualToArray` (Float, Array)**

Checks if a specified float value is less than or equal to at least one of the float values in an array.

**Syntax**

```
_IsValueLessThanOrEqualToArray(valueToTest,valueList)
```

**Parameters**

| Parameter   | Definition                                                                                       | Data type             |
|-------------|--------------------------------------------------------------------------------------------------|-----------------------|
| valueToTest | Float value with which to compare the values in the array.                                       | Float                 |
| valueList   | Array of Float values against which to compare the value specified in the valueToTest parameter. | Array of Float values |

**Returns**

True or False (Boolean), indicating whether the specified float value is less than or equal to at least one of the float values in an array.

**Notes**

The valueToTest value is compared to each element of the array. If the value is less than or equal to every element of the array, the method returns True. If at least one element is greater than the value, the method returns False.

## `_IsValueLessThanOrEqualToArray (Integer, Array)`

Checks if a specified integer value is less than or equal to at least one of the integer values in an array.

### Syntax

```
_IsValueLessThanOrEqualToArray(valueToTest,valueList)
```

### Parameters

| Parameter   | Definition                                                                                         | Data type               |
|-------------|----------------------------------------------------------------------------------------------------|-------------------------|
| valueToTest | Integer value with which to compare the values in the array.                                       | Integer                 |
| valueList   | Array of integer values against which to compare the value specified in the valueToTest parameter. | Array of Integer values |

### Returns

True or False (Boolean), indicating whether the specified integer value is less than or equal to at least one of the integer values in an array.

### Notes

The valueToTest value is compared to each element of the array. If the value is less than or equal to every element of the array, the method returns True. If at least one element is greater than the value, the method returns False.

## `_NormalizeDate`

Normalizes a date based on the specified template. For more information, see [Date time processing](#).

For more information on these specific templates, see:

- [\\_NormalizeDate \(PFDateTime, PFDateTime\)](#)
- [\\_NormalizeDate \(PFDateTime, PFDateTime, PFDateTime\)](#)

## `_NormalizeDate (PFDateTime, PFDateTime)`

Normalizes a single date by replacing the parts that are entered as UNK with the date part values in the specified template.

### Syntax

```
_NormalizeDate(dateToNormalize,template)
```

### Parameters

| Parameter       | Definition                                                                                                                                       | Data type  |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| dateToNormalize | Date time value to normalize by replacing UNK date time parts from a template.                                                                   | PFDateTime |
| template        | Date time value containing the date and time part values with which to replace UNK date and time parts. This template contains no unknown parts. | PFDateTime |

**Returns**

A normalized date time in which all UNK parts are replaced with the corresponding part from the template.

**Notes**

During normalization, the appropriate date time part value from the specified template, in Year-Month-Day-Hour-Minute-Second format, is substituted for each UNK date time part. Any date time parts that are not allowed in the definition of the date time field are replaced with values from the template. For example, if the value of the dateToNormalize is 1987-04-UNK and the value of the template is 1980-01-01, the normalized value is 1987-04-01.

**Example**

In this example, the Start Date and End Date of an adverse event are compared to make sure that the End Date is after the Start Date. However, the patient may not know the exact start and end dates of the adverse event, so the Day part of the date time fields is set to Allow Unknown. A valid comparison requires that the dates first be normalized. Then the rule uses the `_CompareDates` function to compare the dates and creates a query if StartDate is later than EndDate. When normalizing, the function substitutes 1 when the day is omitted.

```
evaluate on Form Submission
  value = _CompareDates(
    _NormalizeDate(this.EndDate.Value, new DateTime(2007, 1, 1, 0, 0, 0)),
    _NormalizeDate(this.StartDate.Value, new DateTime(2007, 1, 1, 0, 0, 0))
  )
when value == -1
  issue query on this.EndDate: End Date cannot be prior to Start Date
```

**`_NormalizeDate` (PFDateTime, PFDateTime, PFDateTime)**

Normalizes a specified date and a reference date by replacing the parts that are entered as UNK with the date part values in the specified template. This function is used to normalize two dates that are being compared. It makes sure that the unknown parts in both dates are considered when determining which parts to normalize.

**Syntax**

```
_NormalizeDate(dateToNormalize,referenceDate,template)
```

**Parameters**

| Parameter       | Definition                                                                                                                                                              | Data type  |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| dateToNormalize | Date time value to normalize by replacing UNK date time parts from a template.                                                                                          | PFDatetime |
| referenceDate   | Second date time value to normalize by replacing UNK date time parts from a template. This date can be compared to the dateToNormalize after both dates are normalized. | PFDatetime |
| template        | Date time value containing the date and time part values with which to replace UNK date and time parts.                                                                 | PFDatetime |

### Returns

A new date with all UNK parts replaced with the corresponding part from the template.

### Notes

This function uses a second date as a reference. If either the date to normalize or the reference date has any UNK parts, those parts will be replaced with the corresponding part from the template. For example:

- **dateToNormalize**—2007/2/5.
- **referenceDate**—2006/5/UNK.
- **template**—2005/1/1.
- **Normalized result for dateToNormalize**—2007/2/1, even though its Day part is not unknown.

Since the reference date has an unknown Day part, both dates are normalized. For more information, see [Date time processing](#).

### Example

In the following example, the `_NormalizeDate` function is used within the `_CompareDates` function to normalize two dates that have unknown parts:

- **date1**—1987-0-UNK-14:45.
- **date2**—1987-2-7-10:UNK.
- **template**—1981-10-1-0:0:0.

```
_CompareDates (_NormalizeDate (date1, date2, new DateTime (1981,10,1,0,0,0)),
_NormalizeDate (date2, date1, new DateTime (1981,10,1,0,0,0))).
```

| Step                                                                                                                 | Result                                     |
|----------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| The first date, <i>date1</i> , is normalized against the template 1981,10,1,0,0,0 with <i>date 2</i> as a reference. | normalized <i>date1</i> = 1987-0-1-14:0:0. |

| Step                                                                                                                  | Result                                           |
|-----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| The second date, <i>date2</i> , is normalized against the template 1981,10,1,0,0,0 with <i>date 1</i> as a reference. | normalized <i>date2</i> = 1987-2-1-10:0:0.       |
| The two normalized results are compared.                                                                              | 1 ( <i>date2</i> is greater than <i>date1</i> ). |

## `_NormalizeDateToMax`

Normalizes UNKNOWN or EMPTY date time parts to the maximum values for those parts. This function takes different parameters for an array of date times and a single date time.

For more information on these different parameters, see

- [\\_NormalizeDateToMax \(Date\)](#)
- [\\_NormalizeDateToMax \(Array\)](#)

## `_NormalizeDateToMax (Date)`

Normalizes a single date time value by replacing any parts that are UNKNOWN or EMPTY with the maximum value for that part.

### Syntax

```
_NormalizeDateToMax(dateToNormalize)
```

### Parameters

| Parameter                    | Definition                  | Data type  |
|------------------------------|-----------------------------|------------|
| <code>dateToNormalize</code> | The date time to normalize. | PFDatetime |

### Returns

A new date time with all UNKNOWN or EMPTY parts replaced with their respective maximum values:

- **Year**—9999
- **Month**—12
- **Day**—(Last day of the month that year)
- **Hour**—23
- **Minute**—59
- **Second**—59

### Notes

The `_NormalizeDateToMax` function accounts for leap years and returns the correct number of days in February of any year.

### Example

The requirement in this example is to find the duration of a previous disease by subtracting the disease start date from the disease end date in the medical history. An unknown month and day are allowed for both start date and end date. An additional requirement is to



substitute unknown date parts with the earliest number for start date and with the latest number for end date so that the longest possible duration of disease is assumed.

To accomplish this:

- Normalize the start date to January 1 of any year.
- Normalize the end date to one of the following:
  - December 31 of any year.
  - The last day of the given month if the month is known.

```
_GetDateDifference(
    _NormalizeDateToMax (this.EVTSTART.Value, new
    DateTime(1,1,1)),
    _NormalizeDate(this.EVTSTART.Value, new DateTime(1,1,1)),
    DateTimeParts.Days)
```

## \_NormalizeDateToMax (Array)

Normalizes an array of date times by replacing any parts that are UNKNOWN or EMPTY with the maximum value for that part.

### Syntax

`_NormalizeDateToMax(datesToNormalize)`

### Parameters

| Parameter                     | Definition                            | Data type                  |
|-------------------------------|---------------------------------------|----------------------------|
| <code>datesToNormalize</code> | The array of date times to normalize. | Array of PFDateTime values |

### Returns

A new array of date times, with all UNKNOWN or EMPTY parts replaced with their respective maximum values:

- **Year**—9999
- **Month**—12
- **Day**—(Last day of the month that year)
- **Hour**—23
- **Minute**—59
- **Second**—59

### Notes

The `_NormalizeDateToMax` function accounts for leap years and returns the correct number of days in February of any year.

You can use arrays called by either the `Values()` or `Objects()` property.

### Example

The requirement in this example is to find the latest date among all dates in a repeating form, using the `MaxValueInArrayDate` function. If the date on the repeating form allows users to enter UNKNOWN for days or months, comparing dates without normalizing them can be unpredictable and depends on the sequence of dates in the array. Therefore, use the `_NormalizeDateToMax(Array)` function to return normalized dates with predictable UNKNOWN parts.

Consider the following arrays of dates, in which the only difference in content is the value of the Day part, which causes the order of the dates to be different in each array:

**Table 3-1** ArrayOfDate1

|       | Day | Month | Year |
|-------|-----|-------|------|
| Date1 | 18  | 10    | 2008 |
| Date2 | UNK | 10    | 2008 |

**Table 3-2** ArrayOfDates2

|       | Day | Month | Year |
|-------|-----|-------|------|
| Date1 | UNK | 10    | 2008 |
| Date2 | 18  | 10    | 2008 |

**Table 3-3** Sample function calls

| Function call                                                                        | Returned value |
|--------------------------------------------------------------------------------------|----------------|
| <code>MaxValueInArrayDate(ArrayOfDates1)</code>                                      | Date1          |
| <code>MaxValueInArrayDate(ArrayOfDates2)</code>                                      | Date2          |
| <code>MaxValueInArrayDate(_NormalizeDateToMax(ArrayOfDates1))</code>                 | Date2          |
| <code>MaxValueInArrayDate(_NormalizeDateToMax(ArrayOfDates2))</code>                 | Date2          |
| <code>MaxValueInArrayDate(_NormalizeDate(ArrayOfDates1, new DateTime(1,1,1)))</code> | Date1          |
| <code>MaxValueInArrayDate(_NormalizeDate(ArrayOfDates2, new DateTime(1,1,1)))</code> | Date1          |

## Randomize

Returns a drug kit number based on the randomization type and the sequence and drug kit numbers stored in the randomization source database on the InForm host computer.

### Syntax

`Randomize(source,type)`

### Parameters

| Parameter | Definition                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Data type |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| source    | Randomization source list name, or stratification.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | String    |
| type      | Randomization type: <ul style="list-style-type: none"> <li>• <b>Simple Central</b>—The study uses one list of drug kits. Each new patient is assigned the next sequential drug kit number on the list.</li> <li>• <b>Central Stratified</b>—The study has multiple lists of drug kits. Each new subject is assigned to a drug kit list based on entered subject data. Then, the subject is assigned the next sequential drug kit number on that list.</li> <li>• <b>Simple Site</b>—Each site has a different drug kit list. Each new subject is assigned the next sequential drug kit number on the list for the site of the subject.</li> <li>• <b>Stratified by Site</b>—Each site has multiple lists of drug kits. Each new subject is first assigned to the set of lists for the site of the subject. Then, the subject is assigned to one of the site drug kit lists based on entered subject data. Finally, the subject is assigned the next sequential drug kit number on that list.</li> </ul> | String    |

### Returns

Drug kit number (String), in the format *sequence/drug\_kit*

### Notes

Using the Randomize function requires special configuration in the study design and on the computer hosting the Oracle Health Sciences InForm study:

- The rule using the Randomization function must calculate the value of an item with the Special Fields custom property value of **Randomization field (Randomization)**.
- The following configuration is required on the computer hosting the Oracle Health Sciences InForm study:

- Define a randomization sequence for each different list of drug kits used in the study and install it in the study database using the MedML Installer tool.
- Configure the randomization data source manager.
- Configure the format of each randomization sequence.
- Set up a randomization source database.
- Create an ODBC connection for the randomization source database.

For more information, see the Oracle Health Sciences InForm documentation.

- [Example—Using the Randomize function](#)

## Example—Using the Randomize function

The randomization feature in the Oracle Health Sciences InForm application allows you to assign a drug kit to a subject based on a randomization scheme that has been chosen for a study. After randomization configuration is complete, one of the forms in the study contains a Drug Kit section. When a user clicks the Randomize button, the Oracle Health Sciences InForm application returns a drug kit number, along with associated information about the drug kit, in the Drug Kit section of the form.

For the randomization feature to work, you must create a randomization item. The randomization item must be on a form with at least one other item that will be completed before the treatment information can be populated. The following example illustrates creating study objects for randomization and using the Randomize function.

1. Create the following form on the Baseline study event.
  - **Title**—Randomization
  - **RefName**—frmRandomization
  - **Short Title**—RAND
2. Create the following items on the frmRandomization form.
  - **Title**—Ready to randomize
  - **RefName**—itmRandReady
  - **Type**—Integer
  - **Question**—Are you ready to dispense treatment to this patient?
  - **Codelist**—clYes
  - **Codelist item**—citmYes
  - **Display override**—Read Only
  - **Title**—Treatment Assignment
  - **RefName**—itmRandTreatment
  - **Question**—Treatment Assignment
  - **Type**—Text
  - **Display override**—Hidden
  - **Special Field Property**—Randomization Field (Randomization)
3. Generate form layout.  
For example:

| <b>RandTest: Randomization (RAND)</b>                                                                                                                                                                                                                                               |                                                              |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| <b>Randomization</b>                                                                                                                                                                                                                                                                |                                                              |
| Answer the question 'Are you ready to dispense treatment to this patient?' and submit the form.                                                                                                                                                                                     |                                                              |
| 1. Are you ready to dispense treatment to this patient?<br><i>[read-only]</i>                                                                                                                                                                                                       | <b>[Ready to randomize]</b><br>[1] <input type="radio"/> Yes |
| 2. Treatment Assignment<br>Once the form has been submitted<br><br><ul style="list-style-type: none"> <li>• Select 'Randomize' from the 'Select Action' list at the base of the screen.</li> <li>• Choose 'Apply' to obtain the Treatment Allocation.</li> </ul><br><i>[hidden]</i> | <b>[Treatment Assignment]</b><br>A128                        |

4. Create a rule on itmRandTreatment.  
The rule expression for the Randomization function uses the following format:

**Randomize([listname], [randomization type])**

 **Note:**

The value of [randomization type] can be one of the following (use the spacing that is specified):

- “Simple Central”
- “Central Stratified”
- “Simple Site”
- “Stratified by Site”

See the following example for more details.

A Simple Central randomization with one list called "SimpleList" uses the following expression.

| <b>Description</b> | <b>Rule expression</b>                               |
|--------------------|------------------------------------------------------|
| –                  | evaluate on Form Submission                          |
| –                  | value = Randomize("SimpleList",<br>"Simple Central") |

---

| Description                                                                                                                                            | Rule expression                                                                   |
|--------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| The following action sets the value of the itmRandTreatment item to the Description and Sequence Number that is returned by the randomization process. | <pre>always set this.Value = value.Description + " " + value.SequenceNumber</pre> |
| The following action sets the value of the itmRandTreatment item to the Description that is returned by the randomization process.                     | <pre>always set this.Value = value.Description</pre>                              |

---

## **\_SaveToDb (String, String)**

Determines whether to send data to the Oracle Health Sciences InForm Publisher queue.

### **Syntax**

`_SaveToDB(<Message>,<Trialname>)`

### **Parameters**

| Parameter | Definition                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Data type |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| message   | <p>Parameter that triggers the Oracle Health Sciences InForm Publisher application to do one of the actions listed below. The rule expression loads the message parameter with a value based on the value of specific items in the form.</p> <p>The text of the message parameter is fixed. The library in the Oracle Health Sciences InForm and Oracle Argus Safety integration package includes the following constants defining enumerations for the required strings. Using these SafetyConstants enumerations to specify the required strings is strongly recommended.</p> <ul style="list-style-type: none"> <li>• IsReadyToSend—Sends safety event data to the Oracle Argus Safety application immediately</li> <li>• IsReportableOrSerious—Marks the safety event as serious and sends it to the Oracle Argus Safety application after a time interval that is configured in the Oracle Health Sciences InForm Publisher application</li> <li>• IsCancelled—Does one of the following: <ul style="list-style-type: none"> <li>– Cancels a pending submission</li> <li>– Sends a nullification submission to the Oracle Argus Safety application if the safety event was submitted previously</li> <li>– Does nothing if the safety event was not marked serious previously</li> </ul> </li> </ul> | String    |
| trialname | Name of the study, populated by the GetTrialName() predefined function.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | String    |

**Example**

```
_SaveToDB('SafetyConstants.IsReportableOrSerious',GetTrialName());
```

## About user-defined functions

An experienced programmer can create a user-defined function using a programming language such as C# and make the function available to Oracle Health Sciences Central Designer users for use in rules. In rule processing, a user-defined function is treated as a local method on the study object class.

Assemblies that contain user-defined functions that are used in a study are included in the deployment package.

To associate user-defined functions with libraries or studies, you must import the functions in the Functions tab of a study or library. For more information, see [Importing a user-defined function](#).

For more information, see:

- [Function definition requirements](#)
- [Recommendations for creating user-defined functions](#)
- [Creating a user-defined function](#)
- [Importing a user-defined function](#)
- [Attributes of user-defined functions](#)
- [Signing user-defined function assemblies](#)
- [Securing user-defined functions](#)
- [Sample function definition code](#)

## Function definition requirements

All standard .NET classes and facilities are available to a function definition. The following requirements apply:

- The function must be defined as a public static method in a .NET class, compiled into a .NET assembly, and saved as a DLL.
- The function must reference the full paths to the following files in the compiler settings:
  - ExternalFunctions.dll (required)
  - PrebuiltFunctions.dll (optional)

The files appear in the Oracle Health Sciences Central Designer client installation folder.

- The assembly containing user-defined function definitions must be self-contained. It cannot reference any other assemblies other than standard .NET framework assemblies.
- The method must return a value of one of the following types:
  - Boolean.
  - Double.
  - Int32.
  - String.
  - PFDateTime. PFDateTime is a customized version of the .NET DateTime data type that allows date time functions to handle incomplete date time fields. For more



information, see [Date time processing](#). Note that user-defined functions cannot use `System.DateTime` as a parameter.

- An array of any of the listed types.
- Each parameter for the method must be of one of the listed types.
- The method cannot directly address any study objects or any other global value. It can operate only on its parameters.
- The name of the method must be the same as the name of the user-defined function.
- The function must return a value. Functions returning void are not permitted.
- The function signature must be unique for each function. (More than one function can have the same name as long as the signature is unique.) The signature of a function consists of the:
  - Function name.
  - Return type and order of the function parameters.
- Two functions that differ only in return type are not permitted because the function signature does not include the return type.
- If a study contains a user-defined function that performs a task such as reading from or writing to a file, accessing the database or the registry, making web service calls, running an external application, sending an email, or using the event log directly, the assembly for the user-defined function must be signed with a strong named signature that is valid and trusted in order for the function to work in the Oracle Health Sciences InForm application. For more information, see [Securing user-defined functions](#).
- Function overloading is supported in accordance with C# overloading rules. The C# compiler determines the function to call based on the best match of actual parameters with the formal parameters of the overloaded function definitions. Automatic type promotion and conversion is used when resolving a function call to the appropriate definition.
- For user-defined functions that are used in studies deployed to Oracle Health Sciences InForm release 6.0 or later and require database access, apply the Read Only User Solution, which includes:
  - Read Only User Manager—Forms and command line utility to set up a read only user and save encrypted connection information to a file and registry.
  - Read Only User Connection Provider—Assembly called `Oracle.HSGBU.ServicesSolutions.ROConnectionProvider.dll` that contains a method to provide a database connection object for a read only user based on study name. The assembly includes the function `DummyFunction()` that allows the assembly to be imported into the Oracle Health Sciences Central Designer application and deployed with the study. Do not use this function in rules.

For more information, contact your Oracle Services representative.

## Recommendations for creating user-defined functions

- Try to make user-defined functions as generic as possible to increase the likelihood that the function will be reused. For example, if you must create a very

- complex function, consider splitting it up into two or three small functions that are more likely to be reused.
- Do not create user-defined functions in a production study or library. Instead, create them in a test study or development library.
  - If you create a user-defined function, you must add the assembly as a reference in your function project and add a namespace for the assembly. The namespace is: `Designer.UserDefinedFunctions`
  - Use C# attributes to categorize user-defined functions. Attributes are already defined in predefined functions.
  - Oracle recommends testing the following parts of a user-defined function:
    - The user-defined function must encapsulate the logic that is required.
    - The correct data points must have been used as parameters.
    - The function must be deployed to the Oracle Health Sciences InForm application and tested with real values.
  - Consider creating a repository of user-defined functions that all users can access. A central repository is especially useful if users are working in different locales or if any part of the study is being outsourced.

## Creating a user-defined function

You can create user-defined functions using a process design application such as the Microsoft Visio application.

1. Create a .NET Class Library project.
2. Specify the Target Framework as .NET Framework 4.0.
3. Rename the default class and the class file for the function.
4. Remove references that are not required.
5. Provide a unique name such as `<Study name>.<Function name>` for the function.
6. Include a reference to the `Oracle.Designer.ExternalFunctions.dll` assembly.
  - a. Locate the assembly in `< Installation Directory >\Bin\ExternalFunctions.`
  - b. Copy the assembly to your project.
  - c. Add a code statement to declare the namespace.
7. If the function performs a task such as reading from or writing to a file, accessing the database or the registry, making web service calls, running an external application, sending an email, or using the event log directly, sign the function with a strong named signature that is valid and trusted.

For more information, see [Signing user-defined function assemblies](#).

8. Import the function into a library or study.

For more information, see [Importing a user-defined function](#).

## Importing a user-defined function

Before you can use a function that is created outside of the Oracle Health Sciences Central Designer application, you must import it to a study or library.

1. Perform one of the following:
  - In a study, select the **Study Information** Explorer bar.
  - In a library, select the **Library Information** Explorer bar.
2. Select **InForm**.
3. Select the **Functions** tab.
4. Click **Import Function**.

The Function File dialog box appears.
5. Locate the .NET assembly (a DLL file) containing the functions to import, and click **Open**.

The functions included in the .NET assembly are stored in the Oracle Health Sciences Central Designer database and are added to the list of functions in the grid.

## Attributes of user-defined functions

The following Oracle Health Sciences Central Designer function attributes are defined in the Oracle Health Sciences Central Designer assembly `Oracle.Designer.ExternalFunctions.dll`, which is required in the project references for user-defined function definitions.

For more information, see [Sample function definition code](#).

See also:

- [DesignerFunctionClassification](#)
- [DesignerFunction](#)
- [DesignerParameter](#)

## DesignerFunctionClassification

**Required:** No.

**Valid on:** The static method of the function on the class containing a collection of function definitions.

**Purpose:** Defines the classification of the function, which is used to group functions in the:

- Functions tab of the Study Editor.
- Function tab of the References section of the Rule Wizard and related dialog boxes.

If you do not include a `DesignerFunctionClassification` attribute, the classification of a function is either:

- The classification defined for the class that contains the function.
- The name of the class, if no classification is defined for the class that contains the function,

**Parameter:**

| Name        | Type   | Description         |
|-------------|--------|---------------------|
| Description | String | Classification name |

## DesignerFunction

**Required:** Yes.

**Valid on:** The static method that defines a Central Designer function.

**Purpose:** Identifies the static method as a user-defined Central Designer function and supplies the descriptive text that appears with the function in the Central Designer application. Only public static functions with this attribute are treated as user-defined function definitions.

**Parameter:**

| Name        | Type   | Description                                                                                                                                                                                                                                                       |
|-------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Description | String | <ul style="list-style-type: none"> <li>• Description of the function. This text appears in the:               <ul style="list-style-type: none"> <li>– Functions tab of the Study Editor.</li> <li>– Rule Wizard and related dialog boxes.</li> </ul> </li> </ul> |

## DesignerParameter

**Required:** No.

**Valid on:** Function parameters.

**Purpose:** Provides the parameter description that is displayed in the Invoke Function dialog box.

**Parameter:**

| Name        | Type   | Description                                                                        |
|-------------|--------|------------------------------------------------------------------------------------|
| Description | String | Description of the parameter. This text appears in the Invoke Function dialog box. |

## Signing user-defined function assemblies

If a study contains a user-defined function that performs a task such as reading from or writing to a file, accessing the database or the registry, making web service calls, running an external application, sending an email, or using the event log directly, the assembly for the user-defined function must be signed with a strong named signature that is valid and trusted in order for the function to work in the Oracle Health Sciences InForm application. For more information, see [Securing user-defined functions](#).

 **Note:**

For Oracle Health Sciences InForm studies hosted by Oracle, all user-defined function assemblies that require signing must be signed by Oracle Services prior to deployment.

When you create a user-defined function with an assembly that you want to secure and sign, on each machine in your environment that builds assemblies, use the following procedure to install a PFX signing file to a crypto service (CSP) certificate container. The PFX file is then used to secure and sign the assembly for the user-defined function.

1. Add the following to the AssemblyInfo.cs file for your project:

```
// Tell compiler to use whatever key pair is stored in <container
name> CSP container. Ignore
// warning that this can be done via command-line switch
#pragma warning disable 1699
[assembly: AssemblyKeyName("<container name>")]
#pragma warning restore 1699
```

where:

- **container name**—CSP container name.
2. Open a Command Prompt window.
  3. Navigate to the directory where your PFX file is located.
  4. Run the following command:

```
sn -i <PFX file name> <container name>.
```

where:

- **PFX file name**—PFX signing file name.
  - **container name**—CSP container name.
5. Build the user-defined function assembly.

## Securing user-defined functions

If a study contains a user-defined function that performs a task such as reading from or writing to a file, accessing the database or the registry, making web service calls, running an external application, sending an email, or using the event log directly, the assembly for the user-defined function must be signed with a strong named signature that is valid and trusted in order for the function to work in the Oracle Health Sciences InForm application.

To ensure that the user-defined functions and assemblies in your study projects and library projects are secure, Oracle recommends that you sign user-defined function assemblies using a strong named, valid and trusted signature. You must create a key pair, extract the public key, and place the key file:

- On the Oracle Health Sciences Central Designer application server. If you are using the Oracle Health Sciences Central Designer application in a web farm environment, place the key on all the application servers in the web farm.

- On all the Oracle Health Sciences InForm application servers where studies using the assembly will be deployed.

**Note:**

For Oracle Health Sciences InForm studies hosted by Oracle, all user-defined function assemblies that require signing must be signed by Oracle Services prior to deployment.

For more information, see [Signing user-defined function assemblies](#).

In addition, user-defined functions that use the Log4Net application must use the latest version of Log4Net or the Oracle Health Sciences Central Designer Log4Net wrapper. The Log4Net wrapper allows untrusted custom functions to use Log4Net logging, and shields users from future Log4Net upgrades. In addition, the wrapper elevates permissions so that the following loggers can run properly:

- Console
- ADO.NET
- EventLog

The wrapper is available in the Oracle.Designer.ExternalFunctions.dll assembly, and contains the following classes and interface:

- PhaseForward.Designer.Shared.Functions.Log4Net.LogManager
- PhaseForward.Designer.Shared.Functions.Log4Net.Log
- PhaseForward.Designer.Shared.Functions.Log4Net.ILog

The methods provided in the classes and interface match those in the Log4Net documentation.

## Sample function definition code

The following code examples illustrate the definition of user-defined functions.

### Body Mass Index (BMI) function

The following code sample illustrates the definition of a function that returns the body-mass index (BMI) based on the values of height and weight. The function has a classification of Clinical. It takes two parameters, height and weight. Descriptive text for the function and its parameters appears in the Rule Wizard and the Invoke Function dialog box.

**Note:**

Oracle Health Sciences Central Designer attributes are shown in bold>.

```
using System;
using Oracle.Designer.ExternalFunctions;
namespace Customer.Designer.ClinicalData.Functions
{
    /// <summary>
```

```

/// Summary description of Clinical
/// </summary>
[DesignerFunctionClassification("Clinical")]
public class Clinical
{
    private Clinical()
    {
    }
    /// <summary>
    /// Returns the body-mass index based on height and weight
    /// </summary>
    /// <param name="height">Height of the patient</param>
    /// <param name="weight">Weight of the patient</param>
    /// <returns></returns>
    [DesignerFunction ("Returns the body-mass index based on height
and weight")]
    public static float BMI (
        [DesignerParameter ("Patient height")]
        float height,
        [DesignerParameter ("Patient weight")]
        float weight)
    {
        return weight / (height * height);
    }
}

```

### Date Time difference function

The following code sample illustrates the definition of a function that returns the length of the interval between any two date time parts. If there are any unknown date time parts, the dates are normalized, and the results are rounded down. This function has a classification of Date Manipulation Function. It takes two dates and a date part as parameters. Descriptive text for the function and its parameters appears in the Rule Wizard and the Invoke Function dialog box.

#### Note:

Oracle Health Sciences Central Designer attributes are shown in bold.

```

using System;
using Oracle.Designer.ExternalFunctions;
namespace Customer.Designer.DateTime.Functions
{
    /// <summary>
    /// Returns the number of units between two dates, based on the
    requested interval.
    /// The units sign will be negative if date1 is before date2
    /// </summary>
    /// <param name="date1">The earlier date</param>
    /// <param name="date2">The later date</param>
    /// <param name="units">The interval to calculate</param>

```

```

    /// <returns>The integer number of interval units. The value will be
    negative if date1 is before date2.</returns>
    /// <remarks>Valid units are taken from the DateTimeParts enumerator,
    and include:
    /// <list type="bullet">
    /// <item>Years</item>
    /// <item>Months</item>
    /// <item>Days</item>
    /// <item>Hours</item>
    /// <item>Minutes</item>
    /// <item>Seconds</item>
    /// </list>
    /// Dates are normalized if either date has any UNKNOWN parts. Results
    are rounded down
    /// to the nearest integer. For example, <code>_GetDateDifference (new
    DateTime(2007,1,10,0,0,0),
    /// new DateTime (2007,2,1,0,0,0), DateTimeParts.Months)</code> will
    return 0, not one</remarks>
    [DesignerFunctionClassification("Date Manipulation Functions")]
    [DesignerFunction("Returns the length of the interval between two
    dates. The dates are normalized if there are any unknown parts. Results are
    rounded down.")]
    public static int _GetDateDifference (
        [DesignerParameter("Date 1 for compare")]
        PFDatetime date1,
        [DesignerParameter("Date 2 for compare")]
        PFDatetime date2,
        [DesignerParameter(". Valid intervals are taken from DateTimeParts
    enum: Years, Months, Days, Hours, Minutes, Seconds")]
        int units)
    {
        if (date1 == null || date2 == null)
        {
            throw new ArgumentException("null parameter to GetDateDifference");
        }

        double difference = 0.0;

        DateTime normalizedEarlierDate = (DateTime)_NormalizeDate(date1,
        date2, new DateTime(2000,1,1,12,0,0));
        DateTime normalizedLaterDate = (DateTime)_NormalizeDate(date2,
        date1, new DateTime(2000,1,1,12,0,0));
        if (normalizedEarlierDate > normalizedLaterDate)
        {
            DateTime temp = normalizedEarlierDate;
            normalizedEarlierDate = normalizedLaterDate;
            normalizedLaterDate = temp;
            sign = -1;
        }
        TimeSpan ts = (DateTime) normalizedLaterDate - (DateTime)
        normalizedEarlierDate;
        switch (units)
        {
            case DateTimeParts.Hours:
                difference = ts.TotalHours;

```



```
        break;
    case DateTimeParts.Minutes:
        difference = ts.TotalMinutes;
        break;
    case DateTimeParts.Seconds:
        difference = ts.TotalSeconds;
        break;
    case DateTimeParts.Days:
        difference = ts.TotalDays;
        break;
        // Rounds down to nearest month
    case DateTimeParts.Months:
    case DateTimeParts.Years:
        difference = (normalizedLaterDate.Year -
normalizedEarlierDate.Year)*12 + (normalizedLaterDate.Month -
normalizedEarlierDate.Month);
        if (normalizedLaterDate.Day < normalizedEarlierDate.Day)
        {
            difference -= 1;
        }
        break;
        }
        if (units == DateTimeParts.Years)
        {
            difference /= 12;
        }
        int ret = ( (int)difference * sign);
        return ret;
    }
}
```

# 4

## Constants

In this chapter:

- [Constants tab of the Rule Wizard](#)
- [Using constants in rules and rule templates](#)
- [Predefined constants in the System Library](#)
- [Creating a constant](#)
- [Deleting a constant](#)

### Constants tab of the Rule Wizard

A constant is a value that is defined in a library or study and that can be referenced by any rule.

You can define a constant in a library and change the value as needed for each study where you use it. For example, you might have a red blood cell count rule on a Hematology form. Because the value for the red blood cell count would be different for a Phase I arthritis trial and a Phase II leukemia trial, you could create a constant (for example, rbcMinCount) and change the value of the constant for each study.

User-defined constants are global. They do not have a scope and can be used in any rule created in a study or library.

You add constants to the Oracle Health Sciences Central Designer application:

- **In a library project**—In the Constants tab of the Library Editor. Constants created in a library are available after they have been published. You can use constants from the library in a study if the library appears on the Library List for the study.
- **In a study project**—In the Constants tab of the Study Editor. Constants created in a study are available only in that study.

### Using constants in rules and rule templates

To add constants to rule expressions, use one of the following:

- **For workflow rules**—Workflow Expression Editor dialog box.
- **For rule templates**—Edit Rule Template dialog box > **Definitions** tab.
- **For data-entry rules**—**Rule Wizard Expression** tab > **Constants** tab.
- **For global conditions**—Edit Global Conditions dialog box > **Constants** tab.

 **Note:**

RefNames for constants used in both a study and a library are not updated in both places if you update the RefName of the constant. If you update a RefName in one location, you must manually update it in the other.

The Constants tab (on the Expression tab of the Rule Wizard) lists all constants created in the study and in the libraries that appear in the Libraries List in the Study Editor. Any rule on any study object can reference any constant that appears.

Constants are organized in folders according to their classification. A classification is a user-defined value provided when a constant is created.

The list of constants is ordered in the following way:

1. Constants defined in the study.
2. Constants published in the first library on the Library List.
3. Constants published in the second library on the Library List (and so on).

You can publish special versions of constants from different libraries and allow the order of libraries in the Library List to determine which constant appears first.

For example, if you have two constants with matching names and data types, you see only the constant from the higher library in the Library List (or, if you have a local copy, in the study).

To include a constant in a rule expression:

- Drag the constant from the Constants tab to the Expression workspace.

## Predefined constants in the System Library

Predefined constants appear in:

- The Constants tab of the System Library.
- The Rule Wizard > Expression tab > Constants tab in the System Library or in a study. Predefined constants do not appear in this location for user-created libraries.

When your Libraries Browser searches include the System Library, the predefined constants are included in the search.

In the Rule Wizard, when you drag a function to the Expression workspace, the Invoke Function dialog box appears. In the dialog box, you can expand the Constants tab, and drag the appropriate constant to the Value field for the parameter.

**DateTimeParts constants**—Use the following constants to specify a parameter in a predefined function that requires a date part. For example, the following functions include date part parameters:

- `_CalculateDateTime`
- `_CompareDatesWithRange`
- `_GetDateDifference`

| Name                                         | Description                                                         | Data type |
|----------------------------------------------|---------------------------------------------------------------------|-----------|
| Years, Months, Days, Hours, Minutes, Seconds | Date time part for years, months, days, hours, minutes, or seconds. | Integer   |

**FormStates constants**—Use the following constants to specify a form state using [HasState\(Integer\)](#).

| Name           | Description                                                                        | Data type |
|----------------|------------------------------------------------------------------------------------|-----------|
| Deleted        | (Available only for repeating forms) The repeating form instance has been deleted. | Integer   |
| Frozen         | The form is frozen.                                                                | Integer   |
| HasComment     | The form has a comment at the form level.                                          | Integer   |
| HasData        | The form has data.                                                                 | Integer   |
| HasMissingData | The form has missing data.                                                         | Integer   |
| HasQueries     | The form has queries.                                                              | Integer   |
| Locked         | The form is locked.                                                                | Integer   |
| SdvComplete    | The form has been marked source verified.                                          | Integer   |
| SdvPartial     | The form has been partially source verified.                                       | Integer   |
| SdvReady       | The form has been marked ready for source verification.                            | Integer   |
| Signed         | The form has been signed.                                                          | Integer   |
| Skipped        | The form has been marked not completed with a form-level comment.                  | Integer   |
| Started        | The form was started with subject data, comments, or queries.                      | Integer   |

## Creating a constant


1. In the upper-left corner of the **Project Explorer**, click the **View Project/View Groups** icon so that you see the Project view. (When the Project view appears, the tooltip for the icon is **View Groups**.)
2. Select a study or library.  
The Study or Library Editor appears.
3. Select the **Constants** tab.
4. Click **New Constant**.  
The New Constant dialog box appears.
5. Fill in the fields of the dialog box, and click **OK**.

For more information, see:

- [Constants tab - Option descriptions](#)

- [New Constant dialog box - Option descriptions](#)

## Constants tab - Option descriptions

| Option                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Fields</b>                 | <div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>Note:</b></p> <p>Not all fields appear in the default view. Optionally, you can add the other fields to the browser view, and you can rearrange the order of fields. For more information, see Showing and hiding a field in the <i>User Guide</i>.</p> </div> |
| Classification                | User-defined text used to group, sort, and filter constants.                                                                                                                                                                                                                                                                                                                                                                                |
| Data Type                     | Return type of the constant: <ul style="list-style-type: none"> <li>• <b>Integer</b>—Contains only numbers.</li> <li>• <b>Float</b>—Contains numbers and a decimal point.</li> <li>• <b>String</b>—Contains alphanumeric characters.</li> <li>• <b>Boolean</b>—True or False.</li> <li>• <b>Date/Time</b>—Contains date and time information.</li> </ul>                                                                                    |
| Description                   | Description of the constant.                                                                                                                                                                                                                                                                                                                                                                                                                |
| Name                          | Name of the constant. The name must: <ul style="list-style-type: none"> <li>• Begin with a letter.</li> <li>• Contain only letters and digits.</li> <li>• Be unique.</li> </ul>                                                                                                                                                                                                                                                             |
| Published (only in libraries) | Indicates that the study object has been published.                                                                                                                                                                                                                                                                                                                                                                                         |
| Value                         | Value of the constant.                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Toolbar buttons</b>        | —                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| New Constant                  | Create and delete a constant.                                                                                                                                                                                                                                                                                                                                                                                                               |
| Delete                        |                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Publish (only in libraries)   | Publish the selected constant. This button is enabled for library objects that have been saved.                                                                                                                                                                                                                                                                                                                                             |

## New Constant dialog box - Option descriptions

| Option         | Description                                                                       |
|----------------|-----------------------------------------------------------------------------------|
| Name           | Name of the constant.                                                             |
| Data Type      | Data type of the constant, such as Integer, Float, String, Boolean, or Date/Time. |
| Classification | User-defined classification of the constant.                                      |
| Description    | Description of the constant.                                                      |
| Value          | Value of the constant. The value must be compatible with its data type.           |

## Deleting a constant

1. In the upper-left corner of the **Project Explorer**, click the **View Project/View Groups** icon so that you see the Project view. (When the Project view appears, the tooltip for the icon is **View Groups**.)
2. Select a study or library.  
The Study or Library Editor appears.
3. Select the **Constants** tab.
4. Select the constant to delete.
5. Perform one of the following:
  - Click **Delete**.
  - Right-click, and select **Delete**.

# 5

## Data mappings

In this chapter:

- [Data Mappings tab](#)
- [Using data mappings in rules and rule templates](#)
- [Icons used on the Data Mappings tab](#)
- [Rule model properties for data series](#)
- [Methods for data sets](#)
- [Additional study objects in the Data Mappings tab](#)

### Data Mappings tab

Study objects and properties that are related to mappings are called data mappings. You add data mappings to rule expressions in the Rule Wizard using the dynamic expression prompts in the Expression tab or by dragging them from the Expression tab > Data Mappings tab. The Data Mappings tab lists:

- RefNames of the data mappings, data sets, and data series in the study or library.
- Rule model properties for data series.  
A data series has the properties of the item that is mapped to it. If a data series contains an item that collects more than one value, the rule model properties for repeating study objects appear so you can access an array of all of the values of the item.
- Methods for data sets.  
A method appears if you select the corresponding standard data dimension of the data set. You can use data set methods to return a subset of the data in the data set.
- Study events, forms, sections, and items that are mapped to each data set.
- Study objects appear if you select the corresponding standard data dimension of the data set. The properties of the study objects are used as parameters of data set methods.

You manage data mappings in the Project Explorer, where you create mappings, data sets, and data series and add items to data series.

### Using data mappings in rules and rule templates

To add data mapping study objects to rule expressions, use one of the following:

- **For workflow rules**—Workflow Expression Editor dialog box.
- **For rule templates**—Edit Rule Template dialog box > **Definition** tab.
- **For data-entry rules**—Rule Wizard Expression tab > **Data Mappings** tab.
- **For global conditions**—Edit Global Conditions dialog box > **Data Mappings** tab.

 **Note:**

Rule expressions use RefNames, not titles. Therefore, RefNames appear in the Rule Wizard and in the rule expression.



The Data Mappings tab (on the Expression tab of the Rule Wizard) lists:

- RefNames of the data mappings, data sets, and data series in the study or library.
- Rule model properties for data series.  
A data series has the properties of the item that is mapped to it. If a data series contains an item that collects more than one value, the rule model properties for repeating study objects appear so you can access an array of all of the values of the item.
- Methods for data sets.  
A method appears if you select the corresponding standard data dimension of the data set. You can use data set methods to return a subset of the data in the data set.
- Study events, forms, sections, and items that are mapped to each data set.
- Study objects appear if you select the corresponding standard data dimension of the data set. The properties of the study objects are used as parameters of data set methods.












To include a data mapping study object or property in a rule expression:

- Drag the data mapping study object or study object property from the Data Mappings tab to the Expression workspace.

## Icons used on the Data Mappings tab

| Icon                                                                                | Description |
|-------------------------------------------------------------------------------------|-------------|
|  | Mapping.    |
|  | Data set.   |



| Icon                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Description                                         |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
|                                                                                                                                                                                                                                                                                                                                                                                                      | Data series.                                        |
| <div style="border-left: 2px solid #0070C0; border-right: 2px solid #0070C0; border-bottom: 2px solid #0070C0; padding: 10px;"> <p> <b>Note:</b></p> <p>If only one item is mapped to a data series, and that item occurs in the study in only one form and study event, the icon that appears is the rule model properties icon for the return type that matches the item's data type.</p> </div> |                                                     |
|                                                                                                                                                                                                                                                                                                                                                                                                      | Methods.                                            |
|                                                                                                                                                                                                                                                                                                                                                                                                    | Rule model properties with a DateTime return type.  |
|                                                                                                                                                                                                                                                                                                                                                                                                    | Rule model properties with a Float return type.     |
|                                                                                                                                                                                                                                                                                                                                                                                                    | Rule model properties with an Integer return type.  |
|                                                                                                                                                                                                                                                                                                                                                                                                    | Rule model properties with a Text return type.      |
|                                                                                                                                                                                                                                                                                                                                                                                                    | Rule model properties with a Boolean return type.   |
|                                                                                                                                                                                                                                                                                                                                                                                                    | Values[ ] and Variables[ ] rule model property.     |
|                                                                                                                                                                                                                                                                                                                                                                                                    | Groups of study events, forms, or items.            |
|                                                                                                                                                                                                                                                                                                                                                                                                    | Individual study events, forms, or items in groups. |



## Rule model properties for data series

Rule model properties are available for a data series in the Expression tab > Data Mappings tab in the Rule Wizard.


For more information, see:


- [Count](#)
- [Values\[\]](#)
- [Variables\[\]](#)
- [Empty](#)
- [Value](#)

### Count


| Characteristic      | Description                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icon</b>         |                                                                                                                                                                                                                                                                                                                      |
| <b>Availability</b> | Repeating study events, repeating forms, and repeating sections.                                                                                                                                                                                                                                                                                                                                      |
|                     | <div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>Note:</b></p> <p>To view this property, you must select the parent study object. For example, to view the property on a study event, you must select its parent (either a study element or a study design).</p> </div> |
| <b>Return type</b>  | Int32 (for Oracle Central Designer Integer and YesNo types).                                                                                                                                                                                                                                                                                                                                          |
| <b>Description</b>  | The current number of instances of the repeating study event, form, or section.                                                                                                                                                                                                                                                                                                                       |
| <b>Purpose</b>      | Use this property to determine the number of sections, forms, or study events in a study. For example, you might want to know the number of adverse events that exist for a subject.                                                                                                                                                                                                                  |

## Values[]


| Characteristic                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icon</b>                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Availability</b>              | An item that is a child of one of the following: <ul style="list-style-type: none"> <li>Form that is part of a repeating study event.</li> <li>Repeating form.</li> <li>Repeating section.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Return type</b>               | <ul style="list-style-type: none"> <li><b>If an integer type</b></li> <li><b>If a float type</b>—Double[].</li> <li><b>If a text type</b></li> <li><b>If a date time type</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Description</b>               | Returns an array of values from item instances that are part of a repeating study event, repeating form, or repeating section.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Included in the array</b>     | Values of undeleted items. An item is undeleted if someone has provided a value for it, and the form or section that contains the item has not been deleted.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Not included in the array</b> | <ul style="list-style-type: none"> <li>Values of deleted items. An item is deleted if the form or section that contains it is deleted from a study.</li> <li>Values of empty items. An item is empty if the form has been created but a value has not been provided for the item.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Example</b>                   | <p>A study contains the following repeating forms, which contain the AEDate item:</p> <ul style="list-style-type: none"> <li><b>First instance of the form</b>—Someone has entered a value for the AEDate item.</li> <li><b>Second instance</b>—Someone has filled in values for other items but not the AEDate item (the item is empty).</li> <li><b>Third instance</b>—Someone entered a value for the AEDate item and deleted the form (the item is deleted).</li> <li><b>Fourth instance</b>—Someone has entered a value for the AEDate item.</li> </ul> <p>When you use this property, the following instances are returned:</p> <ul style="list-style-type: none"> <li>First instance</li> <li>Fourth instance</li> </ul> |

| Characteristic | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Notes</b>   | <div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>Note:</b></p> <p>Values[], AllValues[], Objects[], and AllObjects[] all return an array of data. Consider the following scenarios when deciding upon the property to use:</p> <ul style="list-style-type: none"> <li>• To include deleted values, use either AllValues[] or AllObjects[].</li> <li>• To include null values, use Objects[] or AllObjects[].</li> </ul> </div> <p>Additionally, consider that the following CurrentIndex properties are also available:</p> <ul style="list-style-type: none"> <li>• <a href="#">CurrentAllObjectsIndex</a></li> <li>• <a href="#">CurrentAllValuesIndex</a></li> <li>• <a href="#">CurrentObjectsIndex</a></li> <li>• <a href="#">CurrentValuesIndex</a></li> </ul> |


## Variables[]

| Characteristic      | Description                                                                                                                                                                 |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icon</b>         |                                                                                          |
| <b>Availability</b> | <p>A data series that contains:</p> <ul style="list-style-type: none"> <li>• Two or more items<br/>or</li> <li>• An item that is used more than once in a study.</li> </ul> |
| <b>Return type</b>  | Array.                                                                                                                                                                      |





| Characteristic     | Description                                        |
|--------------------|----------------------------------------------------|
| <b>Description</b> | Array of items that are mapped to the data series. |

 **Note:**  
To indicate the first value in the array, use 0.

## Empty

| Characteristic      | Description                                                                                                                |
|---------------------|----------------------------------------------------------------------------------------------------------------------------|
| <b>Icon</b>         |                                           |
| <b>Availability</b> | A data series that contains a single item that is used only once in a study.<br>Available below the Variables[ ] property. |
| <b>Return type</b>  | Boolean.                                                                                                                   |
| <b>Description</b>  | True or False.<br>When True, a value for the item has not been provided.                                                   |

## Value

| Characteristic      | Description                                                                                                                                                                                                                                                                                                                                                                |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icon</b>         | <br>,<br><br>,<br><br>, or<br> |
| <b>Availability</b> | A data series that contains a single item that is used only once in a study.                                                                                                                                                                                                                                                                                               |

| Characteristic     | Description                                                                                                                                                                                                                                                                                                                          |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Return type</b> | One of the following: <ul style="list-style-type: none"> <li>• String (for Oracle Central Designer string types).</li> <li>• DateTime (for Oracle Central Designer date time types).</li> <li>• Double (for Oracle Central Designer float types).</li> <li>• Int32 (for Oracle Central Designer Integer and YesNo types).</li> </ul> |
| <b>Description</b> | Value of the item in the data series.                                                                                                                                                                                                                                                                                                |

## Methods for data sets

Data sets can contain multiple data series, which can contain multiple items. You use data set methods to return a data set that is a subset of the original data set. Data set methods always return a data set, not a single value.

Methods for data sets appear in the Rule Wizard in the Expression tab > Data Mappings tab. The list of methods that appear is based on the selection of standard data dimensions and the creation of custom data dimensions in the definition of the data set. In addition to the data set methods listed in the following table, one method appears for each custom dimension that is created for the data set.


To use a method in a rule expression:

1. Double-click the method.  
If the method has one or more parameters, the Invoke Function dialog box appears.
2. Drag a study object from the **References** tab to the **Values** field to define a value for each parameter.



For information on these standard data dimensions, see

- [StudyEvent\(StudyEvents\)](#)
- [StudyEvent\(StudyEvents, Integer\)](#)
- [Forms\(Forms\)](#)
- [Form\(Forms, Integer\)](#)
- [Section\(Sections\)](#)
- [Section\(Sections, Integer\)](#)
- [Item\(Items\)](#)
- [\[NameOfCustomDataDimension\]](#)
- [Examples—Data set methods in rule expressions](#)
- [CurrentStudyEvent\(\)](#)

## StudyEvent(StudyEvents)


| Characteristic      | Description                                                                                                                                                                               |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icon</b>         |                                                                                                          |
| <b>Availability</b> | When the <b>Event</b> standard data dimension is selected for the data set.                                                                                                               |
| <b>Returns</b>      | A data set subset with data from only the study event.                                                                                                                                    |
| <b>Syntax</b>       | NameOfDataSet.StudyEvent(StudyEvents.studyEvent)                                                                                                                                          |
| <b>Parameters</b>   | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>studyEvent</i>.</li> <li>• <b>Definition</b>—Study event on which to filter.</li> <li>• <b>Data type</b>—Integer.</li> </ul> |

## StudyEvent(StudyEvents, Integer)



| Characteristic      | Description                                                                                                                                                                                                                                                                                                                                        |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icon</b>         |                                                                                                                                                                                                                                                                  |
| <b>Availability</b> | When the <b>Event</b> and <b>Event Index</b> standard data dimensions are selected for the data set.                                                                                                                                                                                                                                               |
|                     | <div data-bbox="1122 1165 1458 1482" style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>Note:</b></p> <p>The Event Index standard data dimension is necessary for data sets that contain repeating study events.</p> </div> |
| <b>Returns</b>      | A data set subset with data from only the study event.                                                                                                                                                                                                                                                                                             |
| <b>Syntax</b>       | NameOfDataSet.StudyEvent(StudyEvents.studyEvent, studyEventIndex)                                                                                                                                                                                                                                                                                  |
| <b>Parameters</b>   | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>studyEvent</i>.</li> <li>• <b>Definition</b>—Study event on which to filter.</li> <li>• <b>Data type</b>—Integer.</li> </ul>                                                                                                                                                          |

| Characteristic | Description                                                                                                                                                                                                                                                                                  |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| –              | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>studyEventIndex</i>.</li> <li>• <b>Definition</b>—Index of the repeating study event, from 0 to a maximum value, which equals the count of the repeating study event instances.</li> <li>• <b>Data type</b>—Integer.</li> </ul> |

## Forms(Forms)

| Characteristic      | Description                                                                                                                                                                  |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icon</b>         |                                                                                             |
| <b>Availability</b> | When the <b>Form</b> standard data dimension is selected for the data set.                                                                                                   |
| <b>Returns</b>      | A data set subset with data from only the form.                                                                                                                              |
| <b>Syntax</b>       | NameOfDataSet.Form(Forms.form)                                                                                                                                               |
| <b>Parameters</b>   | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>form</i>.</li> <li>• <b>Definition</b>—Form on which to filter.</li> <li>• <b>Data type</b>—Integer.</li> </ul> |


## Form(Forms, Integer)

| Characteristic      | Description                                                                                                                                                                                                                                                                                                                                       |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icon</b>         |                                                                                                                                                                                                                                                                |
| <b>Availability</b> | When the <b>Form</b> and <b>Form Index</b> standard data dimensions are selected for the data set.                                                                                                                                                                                                                                                |
|                     | <div data-bbox="1084 1392 1380 1707" style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>Note:</b></p> <p>The Form Index standard data dimension is necessary for data sets that contain repeating study events.</p> </div> |
| <b>Returns</b>      | A data set subset with data from only the form.                                                                                                                                                                                                                                                                                                   |
| <b>Syntax</b>       | NameOfDataSet.Form(Forms.form, formIndex)                                                                                                                                                                                                                                                                                                         |





| Parameter        | Definition                                                                                                      | Data type |
|------------------|-----------------------------------------------------------------------------------------------------------------|-----------|
| <b>form</b>      | Form on which to filter.                                                                                        | Integer   |
| <b>formIndex</b> | Index of the repeating form, from 0 to a maximum value, which equals the count of the repeating form instances. | Integer   |

## Section(Sections)


| Characteristic      | Description                                                                                                                                            |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icon</b>         |                                                                       |
| <b>Availability</b> | When the <b>Section</b> standard data dimension is selected for the data set.                                                                          |
| <b>Returns</b>      | A data set subset with data from only the section.                                                                                                     |
| <b>Syntax</b>       | NameOfDataSet.Section(Sections.section)                                                                                                                |
| <b>Parameters</b>   | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>section</i>.</li> <li>• <b>Definition</b></li> <li>• <b>Data type</b>—Integer.</li> </ul> |

## Section(Sections, Integer)

| Characteristic      | Description                                                                                                                                                                                                                                                                                                                                     |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icon</b>         |                                                                                                                                                                                                                                                              |
| <b>Availability</b> | When the <b>Section</b> and <b>Section Index</b> standard data dimensions are selected for the data set.                                                                                                                                                                                                                                        |
|                     | <div data-bbox="1122 1297 1458 1583" style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>Note:</b></p> <p>The Section Index standard data dimension is required for data sets that contain repeating sections.</p> </div> |
| <b>Returns</b>      | A data set subset with data from only the section.                                                                                                                                                                                                                                                                                              |
| <b>Syntax</b>       | NameOfDataSet.Section(Sections.section, sectionIndex)                                                                                                                                                                                                                                                                                           |
| <b>Parameters</b>   | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>section</i>.</li> <li>• <b>Definition</b>—Section on which to filter.</li> <li>• <b>Data type</b>—Integer.</li> </ul>                                                                                                                                                              |

| Characteristic | Description                                                                                                                                                                                                                                                                       |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| –              | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>sectionIndex</i>.</li> <li>• <b>Definition</b>—Index of the repeating section, from 0 to a maximum value, which equals the count of the repeating section instances.</li> <li>• <b>Data type</b>—Integer.</li> </ul> |

## Item(Items)


| Characteristic      | Description                                                                                                                                                                  |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icon</b>         |                                                                                             |
| <b>Availability</b> | When the <b>Item</b> standard data dimension is selected for the data set.                                                                                                   |
| <b>Returns</b>      | A data set subset with data from only the item.                                                                                                                              |
| <b>Syntax</b>       | NameOfDataSet.Item(Items.item)                                                                                                                                               |
| <b>Parameters</b>   | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>item</i>.</li> <li>• <b>Definition</b>—Item on which to filter.</li> <li>• <b>Data type</b>—Integer.</li> </ul> |

## [NameOfCustomDataDimension



### Note:

A method is generated for every custom data dimension that is created for a data set.

| Characteristic      | Description                                                                                                                                                                                                                                                                  |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icon</b>         |                                                                                                                                                                                           |
| <b>Availability</b> | When a custom data dimension is created for a data set.                                                                                                                                                                                                                      |
| <b>Returns</b>      | A data set subset with data from only the custom data dimension.                                                                                                                                                                                                             |
| <b>Syntax</b>       | NameOfDataSet.dimension(ValueOfDimension )                                                                                                                                                                                                                                   |
| <b>Parameters</b>   | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>dimension</i>.</li> <li>• <b>Definition</b>—Name of the custom dimension on which to filter.</li> <li>• <b>Data type</b>—String or Integer, depending on the data type of the custom data dimension.</li> </ul> |

## Examples—Data set methods in rule expressions

The examples use the following study object names:

- **Study event**—Visit1
- **Form**—Vitals
- **Item**—Pulse
- **Data set**—VitalSigns

### Structure

You must use the following structure when you use a method in a rule expressions.



**1**—Name of the data set.

**2**—Name of the method.

**3**—Filtering information created when you provide a value for the parameter or parameters of the method.

To use a method in a rule expression:



1. Double-click the method.  
If the method has one or more parameters, the Invoke Function dialog box appears.
2. Drag a study object from the **References** tab to the **Values** field to define a value for each parameter.

### Examples

The following table provides examples of rule expressions that you can use to obtain a subset of a data set. The examples use the study object names defined for the examples.

| Rule expression                                                                             | Returns a data set subset with data from..                   |
|---------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| <code>VitalSigns.StudyEvent(StudyEvents.Visit1)</code>                                      | Visit1 only.                                                 |
| <code>VitalSigns.StudyEvent(StudyEvents.Visit1).Form(Forms.Vitals)</code>                   | The Vitals form in Visit1 only.                              |
| <code>VitalSigns.StudyEvent(StudyEvents.Visit1).Form(Forms.Vitals).Item(Items.Pulse)</code> | The Pulse item on the Vitals form in the Visit1 study event. |
| <code>VitalSigns.Form(Forms.Vitals)</code>                                                  | The Vitals form in any study event.                          |
| <code>VitalSigns.StudyEvent(StudyEvents.Visit1).Item(Items.Pulse)</code>                    | The Pulse item on any form in the Visit1 study event.        |

## CurrentStudyEvent()

| Characteristic      | Description                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icon</b>         |                                                                                                                                                                                                                                                                                                                                   |
| <b>Availability</b> | When the <b>Event</b> and <b>Event Index</b> standard data dimensions are selected for the data set.                                                                                                                                                                                                                                                                                                               |
|                     | <div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>Note:</b></p> <ul style="list-style-type: none"> <li>• CurrentStudyEvent() is only available for Query and Calculation rules.</li> <li>• CurrentStudyEvent() is only available for rules on a study event or lower.</li> </ul> </div> |
| <b>Returns</b>      | <p>A data set subset with data from only the current study event.</p> <p>If the study object you map to the data set has multiple values (for example, if you map an item on a repeating form), you might get an array or many values, rather than just one value.</p>                                                                                                                                             |
| <b>Syntax</b>       | NameOfDataSet.CurrentStudyEvent()                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Parameters</b>   | None.                                                                                                                                                                                                                                                                                                                                                                                                              |

## Additional study objects in the Data Mappings tab



In addition to rule model properties and methods, the Data Mappings tab displays study events, forms, sections, and items that are part of the data set that is expanded in the References section. Study objects appear if they are selected as standard data dimensions of a data set. For example, if you select the Form standard dimension, the Forms node appears under the data set and under the Form(Forms) method.

You can use the study events, forms, sections, and items to define values of parameters for methods. When you double-click a method to use it in the rule expression, the Invoke Function dialog box appears, prompting you to define values for the parameters of the method. You can drag a study object to the Value field for each parameter.

For more information, see

- [Study events](#)
- [Forms](#)
- [Sections](#)
- [Items](#)



## Study events


| Characteristic      | Description                                                                                                                                                                                                                                        |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icons</b>        |  <p>—For a collection of study events.</p>  <p>—For individual study events.</p> |
| <b>Availability</b> | <p>All study events that are part of the expanded data set appear in the list.</p> <p>Available as:</p> <ul style="list-style-type: none"> <li>• Properties of data series.</li> <li>• Properties of data set methods.</li> </ul>                  |
| <b>Description</b>  | <p>Use individual study events to define the values of parameters for methods.</p>                                                                                                                                                                 |

 **Note:**



The collective lists of study objects cannot be used in a rule expression. You can use only individual study objects in a rule expression.

# Forms

| Characteristic      | Description                                                                                                                                                                                                                         |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icons</b>        | <br>—For a collection of forms.<br><br><br>—For individual forms. |
| <b>Availability</b> | All forms that are part of the expanded data set appear in the list.<br>Available as: <ul style="list-style-type: none"><li>• Properties of data series.</li><li>• Properties of data set methods.</li></ul>                        |
| <b>Description</b>  | Use individual forms to define the values of parameters for methods.                                                                                                                                                                |

 **Note:**  
The collective lists of study objects cannot be used in a rule expression. You can use only individual study objects in a rule expression.

## Sections



| Characteristic      | Description                                                                                                                                                                                                                               |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icons</b>        | <br>—For a collection of sections.<br><br><br>—For individual sections. |
| <b>Availability</b> | All sections that are part of the expanded data set appear in the list.<br>Available as: <ul style="list-style-type: none"><li>• Properties of data series.</li><li>• Properties of data set methods.</li></ul>                           |
| <b>Description</b>  | Use individual sections to define the values of parameters for methods.                                                                                                                                                                   |




**Note:**

The collective lists of study objects cannot be used in a rule expression. You can use only individual study objects in a rule expression.

# Items

| Characteristic      | Description                                                                                                                                                                                                                         |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Icons</b>        | <br>—For a collection of items.<br><br><br>—For individual items. |
| <b>Availability</b> | All items that are part of the expanded data set appear in the list.<br>Available as: <ul style="list-style-type: none"><li>• Properties of data series.</li><li>• Properties of data set methods.</li></ul>                        |
| <b>Description</b>  | Use individual items to define the values of parameters for methods.                                                                                                                                                                |

 **Note:**  
The collective lists of study objects cannot be used in a rule expression. You can use only individual study objects in a rule expression.



# 6

## Methods, operators, and literals

In this chapter:

- [Methods](#)
- [Operators and literals](#)

### Methods

A method is a block of code that is called by a rule and that is used to manipulate data.

You can use standard C# methods, including math methods, as well as methods that are automatically generated for certain study objects, to create rule expressions. For example, you can use a count method to return the number of instances created for a repeating form.

You can use any method in a rule expression.

For more information, see

- [Math methods](#)
- [Methods for study objects](#)
- [Data set methods](#)

### Math methods

A math method performs a mathematical operation on a value or set of values and can be used in a rule expression. Math methods are provided by the Microsoft .NET framework.



#### Note:

When you use a math method in a rule expression, you must precede it with **Math.** (Math followed by a period). For example, the Pow method, which is used to return a value raised to a specific power, becomes **Math.Pow**.

For a comprehensive list of math methods, see a C# programming guide.

For more information, see

- [Abs](#)
- [Ceiling](#)
- [DivRem](#)
- [Exp](#)
- [Floor](#)
- [IEEERemainder](#)

- [Log](#)
- [Log10](#)
- [Max](#)
- [Min](#)
- [Pow](#)
- [Round](#)
- [Sqrt](#)
- [Truncate](#)
- [Examples—Math methods in rule expressions](#)

## Abs

| Characteristic     | Description                                                                                                                                                                                                         |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Returns</b>     | Absolute value of a specified number.                                                                                                                                                                               |
| <b>Return type</b> | Data type of the parameter that you enter.                                                                                                                                                                          |
| <b>Syntax</b>      | Math.Abs(a)                                                                                                                                                                                                         |
| <b>Parameters</b>  | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>a</i>.</li> <li>• <b>Definition</b>—A number for which an absolute value is to be found.</li> <li>• <b>Data type</b>—Any numeric data type.</li> </ul> |

## Ceiling

| Characteristic     | Description                                                                                                                                                                                                                                        |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Returns</b>     | Smallest integer greater than or equal to the specified number.                                                                                                                                                                                    |
| <b>Return type</b> | Int32 (for Oracle Health Sciences Central Designer Integer and YesNo types).                                                                                                                                                                       |
| <b>Syntax</b>      | Math.Ceiling(a)                                                                                                                                                                                                                                    |
| <b>Parameters</b>  | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>a</i>.</li> <li>• <b>Definition</b>—A number for which the smallest integer greater than or equal to it is to be found.</li> <li>• <b>Data type</b>—Any numeric data type.</li> </ul> |

## DivRem

| Characteristic | Description                                                            |
|----------------|------------------------------------------------------------------------|
| <b>Returns</b> | Quotient of two numbers and also the remainder in an output parameter. |

| Characteristic     | Description                                                                                                                                                                                                                                               |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Return type</b> | <ul style="list-style-type: none"> <li>• <b>Quotient</b>—Int32 (for Oracle Health Sciences Central Designer Integer and YesNo types).</li> <li>• <b>Remainder</b>—Int32 (for Oracle Health Sciences Central Designer Integer and YesNo types).</li> </ul> |
| <b>Syntax</b>      | Math.DivRem(a, b, c)                                                                                                                                                                                                                                      |
| <b>Parameters</b>  | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>a</i></li> <li>• <b>Definition</b>—Dividend</li> <li>• <b>Data type</b>—Int32</li> </ul>                                                                                                     |
| –                  | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>b</i></li> <li>• <b>Definition</b>—Divisor</li> <li>• <b>Data type</b>—Int32</li> </ul>                                                                                                      |
| –                  | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>c</i></li> <li>• <b>Definition</b>—Remainder</li> <li>• <b>Data type</b>—Int32</li> </ul>                                                                                                    |

## Exp

| Characteristic     | Description                                                                                                                                                                              |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Returns</b>     | e raised to the specified power.                                                                                                                                                         |
| <b>Return type</b> | Double (for Oracle Health Sciences Central Designer float types).                                                                                                                        |
| <b>Syntax</b>      | Math.Exp(a)                                                                                                                                                                              |
| <b>Parameters</b>  | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>a</i></li> <li>• <b>Definition</b>—Number specifying a power.</li> <li>• <b>Data type</b>—Any numeric data type.</li> </ul> |

## Floor

| Characteristic     | Description                                                                                                                                                                                                                                   |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Returns</b>     | Largest integer less than or equal to the specified number.                                                                                                                                                                                   |
| <b>Return type</b> | Int32 (for Oracle Health Sciences Central Designer Integer and YesNo types).                                                                                                                                                                  |
| <b>Syntax</b>      | Math.Floor(a)                                                                                                                                                                                                                                 |
| <b>Parameters</b>  | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>a</i></li> <li>• <b>Definition</b>—A number for which the largest integer less than or equal to it is to be found.</li> <li>• <b>Data type</b>—Any numeric data type.</li> </ul> |

## IEEERemainder

| Characteristic     | Description                                                                                                                                                                                                                                                                                           |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Returns</b>     | Remainder resulting from the division of a specified number by another specified number.                                                                                                                                                                                                              |
| <b>Return type</b> | Double (for Oracle Health Sciences Central Designer float types).                                                                                                                                                                                                                                     |
| <b>Syntax</b>      | Math.IEEERemainder(a, b)                                                                                                                                                                                                                                                                              |
| <b>Parameters</b>  | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>a</i></li> <li>• <b>Definition</b>—Dividend.</li> <li>• <b>Data type</b>—Any numeric data type.</li> <li>• <b>Parameter</b>—<i>b</i></li> <li>• <b>Definition</b>—Divisor.</li> <li>• <b>Data type</b>—Any numeric data type.</li> </ul> |
| –                  |                                                                                                                                                                                                                                                                                                       |

## Log

| Characteristic     | Description                                                                                                                                                                                                  |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Returns</b>     | Logarithm of a specified number.                                                                                                                                                                             |
| <b>Return type</b> | Double (for Oracle Health Sciences Central Designer float types).                                                                                                                                            |
| <b>Syntax</b>      | Math.Log(a)                                                                                                                                                                                                  |
| <b>Parameters</b>  | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>a</i></li> <li>• <b>Definition</b>—A number for which a logarithm is to be found.</li> <li>• <b>Data type</b>—Any numeric data type.</li> </ul> |

## Log10

| Characteristic     | Description                                                                                                                                                                                                  |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Returns</b>     | Base 10 logarithm of a specified number.                                                                                                                                                                     |
| <b>Return type</b> | Double (for Oracle Health Sciences Central Designer float types).                                                                                                                                            |
| <b>Syntax</b>      | Math.Log10(a)                                                                                                                                                                                                |
| <b>Parameters</b>  | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>a</i></li> <li>• <b>Definition</b>—A number for which a logarithm is to be found.</li> <li>• <b>Data type</b>—Any numeric data type.</li> </ul> |

## Max

| Characteristic     | Description                           |
|--------------------|---------------------------------------|
| <b>Returns</b>     | Larger of two specified numbers.      |
| <b>Return type</b> | Data type of the parameter you enter. |

| Characteristic    | Description                                                                                                                                                                                              |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>     | Math.Max(a, b)                                                                                                                                                                                           |
| <b>Parameters</b> | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>a</i></li> <li>• <b>Definition</b>—The value of the first number to compare.</li> <li>• <b>Data type</b>—Any numeric data type.</li> </ul>  |
| –                 | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>b</i></li> <li>• <b>Definition</b>—The value of the second number to compare.</li> <li>• <b>Data type</b>—Any numeric data type.</li> </ul> |

## Min

| Characteristic     | Description                                                                                                                                                                                              |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Returns</b>     | Smaller of two specified numbers.                                                                                                                                                                        |
| <b>Return type</b> | Data type of the parameter you enter.                                                                                                                                                                    |
| <b>Syntax</b>      | Math.Min(a, b)                                                                                                                                                                                           |
| <b>Parameters</b>  | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>a</i></li> <li>• <b>Definition</b>—The value of the first number to compare.</li> <li>• <b>Data type</b>—Any numeric data type.</li> </ul>  |
| –                  | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>b</i></li> <li>• <b>Definition</b>—The value of the second number to compare.</li> <li>• <b>Data type</b>—Any numeric data type.</li> </ul> |

## Pow

| Characteristic     | Description                                                                                                                                                                                       |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Returns</b>     | Specified number raised to the specified power.                                                                                                                                                   |
| <b>Return type</b> | Double (for Central Designer float types).                                                                                                                                                        |
| <b>Syntax</b>      | Math.Pow(a, b)                                                                                                                                                                                    |
| <b>Parameters</b>  | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>a</i></li> <li>• <b>Definition</b>—The number to be raised to a power.</li> <li>• <b>Data type</b>—Any numeric data type.</li> </ul> |
| –                  | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>b</i></li> <li>• <b>Definition</b>—The number that specifies a power.</li> <li>• <b>Data type</b>—Any numeric data type.</li> </ul>  |

## Round

| Characteristic     | Description                                                                                                                                                                                                                                                                                                                                               |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Returns</b>     | Value rounded to the nearest integer or specified number of decimal places.                                                                                                                                                                                                                                                                               |
| <b>Return type</b> | Int32 (for Oracle Health Sciences Central Designer Integer and YesNo types).<br>or<br>Double (for Oracle Health Sciences Central Designer float types).                                                                                                                                                                                                   |
| <b>Syntax</b>      | Math.Round(a, b)                                                                                                                                                                                                                                                                                                                                          |
| <b>Parameters</b>  | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>a</i></li> <li>• <b>Definition</b>—A number to be rounded.</li> <li>• <b>Data type</b>—Any numeric data type.</li> <li>• <b>Parameter</b>—<i>b</i></li> <li>• <b>Definition</b>—The number of decimal places (precision) in the return value.</li> <li>• <b>Data type</b>—Integer</li> </ul> |
| –                  |                                                                                                                                                                                                                                                                                                                                                           |

## Sqrt

| Characteristic     | Description                                                                                                                                                                                                    |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Returns</b>     | Square root of a specified number.                                                                                                                                                                             |
| <b>Return type</b> | Double (for Oracle Health Sciences Central Designer float types).                                                                                                                                              |
| <b>Syntax</b>      | Math.Sqrt(a)                                                                                                                                                                                                   |
| <b>Parameters</b>  | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>a</i></li> <li>• <b>Definition</b>—A number for which a square root is to be found.</li> <li>• <b>Data type</b>—Any numeric data type.</li> </ul> |

## Truncate

| Characteristic     | Description                                                                                                                                                                                                        |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Returns</b>     | Integral part of a number.                                                                                                                                                                                         |
| <b>Return type</b> | Int32 (for Oracle Health Sciences Central Designer Integer and YesNo types).                                                                                                                                       |
| <b>Syntax</b>      | Math.Truncate(a)                                                                                                                                                                                                   |
| <b>Parameters</b>  | <ul style="list-style-type: none"> <li>• <b>Parameter</b>—<i>a</i></li> <li>• <b>Definition</b>—A number for which the integral part is to be found.</li> <li>• <b>Data type</b>—Any numeric data type.</li> </ul> |

## Examples—Math methods in rule expressions

For example, you might want to raise a query when the value of an item is a specific distance from a known mean value. You can create constants called Mean and Range for the mean and the range from the mean value. The rule might look like this:

```
evaluate on Form Submission
value = (Mean + Range) >= item.Value && item.Value >= (Mean - Range)
when value is false
query "Value is out of range"
```

However, because the difference between the mean value and the range could be positive or negative, you must check the absolute value against the range. In that case, you can use the math method that returns an absolute value. Your rule might look like this:

```
evaluate on Form Submission
value = Math.Abs(Mean - item.Value) <= Range
when value is false
query "Value is out of range"
```

## Methods for study objects

For a list of methods for repeating and non-repeating study objects, see:

- [Methods for repeating study objects.](#)
- [Methods for non-repeating study objects.](#)

## Data set methods

Data sets can contain multiple data series, which can contain multiple items. You use data set methods to return a data set that is a subset of the original data set. Data set methods always return a data set, not a single value.

Methods for data sets appear in the Rule Wizard in the Expression tab > Data Mappings tab. The list of methods that appear is based on the selection of standard data dimensions and the creation of custom data dimensions in the definition of the data set. In addition to the data set methods listed in the following table, one method appears for each custom dimension that is created for the data set.

| Standard data dimensions selected | Data set method in Data Mappings tab             |
|-----------------------------------|--------------------------------------------------|
| Event                             | <a href="#">StudyEvent(StudyEvents)</a>          |
| Event and Event Index             | <a href="#">StudyEvent(StudyEvents, Integer)</a> |
| Form                              | <a href="#">Forms(Forms)</a>                     |
| Form and Form Index               | <a href="#">Form(Forms, Integer)</a>             |
| Section                           | <a href="#">Section(Sections)</a>                |
| Section and Section Index         | <a href="#">Section(Sections, Integer)</a>       |
| Item                              | <a href="#">Item(Items)</a>                      |

**Note:**

You cannot use a data set method more than once in a rule expression.

## Operators and literals

You can use operators:

- To create rule expressions.
- To include an expression in the parameters of a function.
- To include an expression in the "when" part of a rule.

For more information, see:

- [Frequently used operators](#)
- [Frequently used literals](#)

## Frequently used operators

For more information, see a C# or Java programming reference.

| Operator category             | Operator | Use          | Description                                                                                                                                                 |
|-------------------------------|----------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Arithmetic                    | +        | $x + y$      | Adds x and y.                                                                                                                                               |
| –                             | -        | $x - y$      | Subtracts y from x.                                                                                                                                         |
| –                             | *        | $x * y$      | Multiplies x by y.                                                                                                                                          |
| –                             | /        | $x / y$      | Divides x by y.                                                                                                                                             |
| –                             | %        | $x \% y$     | Returns the remainder of integer division of x and y.                                                                                                       |
| Logical (Boolean and bitwise) | &        | $x \& y$     | Evaluates both x and y and returns the logical conjunction ("AND") of their results. If x and y are integers, the logical conjunction is performed bitwise. |
| –                             |          | $x   y$      | Evaluates x and y and returns the logical disjunction ("OR") of their results. If x and y are integers, the logical disjunction is performed bitwise.       |
| –                             | ^        | $x \wedge y$ | Returns the exclusive or ("XOR") of their results. If x and y are integers, the exclusive or is performed bitwise.                                          |



| Operator category    | Operator | Use    | Description                                                                                                                                                                                                                                                                                                                                   |
|----------------------|----------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| –                    | !        | !x     | Evaluates x and returns the negation ("NOT") of the result. <ul style="list-style-type: none"> <li>• If x evaluates to false, it returns true.</li> <li>• If x evaluates to true, it returns false.</li> </ul>                                                                                                                                |
| –                    | ~        | ~x     | Evaluates x and returns the bitwise negation of the result. ~x returns a value where each bit is the negation of the corresponding bit in the result of evaluating x.                                                                                                                                                                         |
| –                    | &&       | x && y | Evaluates x. <ul style="list-style-type: none"> <li>• If the result is false, it returns false.</li> <li>• Otherwise, it evaluates and returns the results of y.</li> </ul> <p>Note that if evaluating y would hypothetically have no side effects, the results are identical to the logical conjunction performed by the &amp; operator.</p> |
| –                    |          | x    y | Evaluates x. <ul style="list-style-type: none"> <li>• If the result is true, it returns true.</li> <li>• Otherwise, it evaluates and returns the results of y.</li> </ul> <p>Note that if evaluating y would hypothetically have no side effects, the results are identical to the logical disjunction performed by the   operator.</p>       |
| –                    | ()       | –      | Used to group information.                                                                                                                                                                                                                                                                                                                    |
| String concatenation | +        | x + y  | Concatenates strings.                                                                                                                                                                                                                                                                                                                         |

| Operator category | Operator | Use        | Description                                                                                                                                                                                       |
|-------------------|----------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Relational        | ==       | x == y     | <ul style="list-style-type: none"> <li>If x and y have the same value, it returns true.</li> <li>Otherwise, it returns false.</li> </ul>                                                          |
| –                 | !=       | x != y     | <p>Returns the logical negation of the operator ==.</p> <ul style="list-style-type: none"> <li>If x is not equal to y, it returns true.</li> <li>If x is equal to y, it returns false.</li> </ul> |
| –                 | <        | x < y      | <ul style="list-style-type: none"> <li>If x is less than y, it returns true.</li> <li>Otherwise, it returns false.</li> </ul>                                                                     |
| –                 | >        | x > y      | <ul style="list-style-type: none"> <li>If x is greater than y, it returns true.</li> <li>Otherwise, it returns false.</li> </ul>                                                                  |
| –                 | <=       | x <= y     | <ul style="list-style-type: none"> <li>If x is less than or equal to y, it returns true.</li> <li>Otherwise, it returns false.</li> </ul>                                                         |
| –                 | >=       | x >= y     | <ul style="list-style-type: none"> <li>If x is greater than or equal to y, it returns true.</li> <li>Otherwise, it returns false.</li> </ul>                                                      |
| Member access     | .        | this.Value | Used to form long names.                                                                                                                                                                          |
| Indexing          | [ ]      | –          | Used to access array elements.                                                                                                                                                                    |
| Conditional       | ?:       | x ? y : z  | <ul style="list-style-type: none"> <li>If x is true, it returns y.</li> <li>Otherwise, it returns z.</li> </ul>                                                                                   |
| Assignment        | =        | x = y      | Assigns the value of y to x.                                                                                                                                                                      |

## Frequently used literals

Literals are like constants that you can include in expressions.

| Literal type | Literal | Description                   |
|--------------|---------|-------------------------------|
| Boolean      | true    | A literal for true and false. |
| –            | false   | –                             |

---

| <b>Literal type</b> | <b>Literal</b> | <b>Description</b>                                                                                                                               |
|---------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| String              | "xxx"          | A group of characters. Strings are delimited with double quotes ("). To quote within a string, use a backslash (\) to precede the double quotes. |
| –                   | "xx\"x"        | –                                                                                                                                                |
| Integer             | Example: 13    | A number without a decimal point or exponent.                                                                                                    |
| Float               | Example: 13.2  | A number with a decimal point or exponent.                                                                                                       |

---

# 7

## Sample expressions for data-entry rules

In this chapter:

- [Sample expressions that use operators](#)
- [Sample data-entry rule that uses the Data tab](#)
- [Sample data-entry rule that uses rule model properties](#)
- [Sample data-entry rules that use methods](#)
- [Sample data-entry rule that uses constants](#)
- [Sample data-entry rules that use functions](#)
- [Sample calculation rules](#)
- [Sample data-entry rules that use mappings](#)

### Sample expressions that use operators

| Example:                                                                                | Expression:                                                                                                  |
|-----------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| x is greater than y                                                                     | <code>x &gt; y</code>                                                                                        |
| x is less than or equal to z                                                            | <code>x &lt;= z</code>                                                                                       |
| x is greater than y and x is less than or equal to z                                    | <code>(x &gt; y) &amp;&amp; (x &lt;= z)</code>                                                               |
| x is greater than y or x is less than or equal to z                                     | <code>(x &gt; y)    (x &lt;= z)</code>                                                                       |
| x is equal to 3 or x is equal to 6.4                                                    | <code>(x = =3)    (x = =6.4)</code>                                                                          |
| x is not more than ten percent greater than y                                           | <code>x &lt;= y*1.1</code><br><code>!(x&gt;y*1.1)</code>                                                     |
| If x is greater than y, then value is 100, else value is 200.                           | <code>x&gt;y?100:200</code>                                                                                  |
| If x is not greater than y, then check if y is greater than z. Otherwise, return false. | <code>!(x&gt;y)?y&gt;z:false</code><br><code>x&gt;y?false:y&gt;z</code><br><code>x&lt;=y?y&gt;z:false</code> |

### Sample data-entry rule that uses the Data tab

Example 1

| Characteristic     | Description                                                                                                                                                                                                                                               |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Description</b> | Create a rule named <code>rulCompareQTcQRS</code> that checks that the QTc interval (on the ECG form) is greater than (exclusive) the QRS interval. If false, issue a query on the QTc item that says <b>QTc must be greater than QRS. Please verify.</b> |

| Characteristic         | Description                                                                                                                                                                                                                    |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Scope</b>           | ECG form                                                                                                                                                                                                                       |
| <b>Study structure</b> | ECG (form) <ul style="list-style-type: none"> <li>• QTcInt (item)</li> <li>• QRSDur (item)</li> </ul>                                                                                                                          |
| <b>Rule summary</b>    | <p>evaluate on Form Submission</p> <pre>value = this.itmQTcInt.Value &gt; this.itmQRSInt.Value</pre> <p>when value is false<br/>issue query on this.QTcInt: QTc interval must be greater than QRS interval. Please verify.</p> |

#### Example 2

| Characteristic         | Description                                                                                                                                                                                                                                                    |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Description</b>     | Create a rule named <code>rulExclusionCheck1</code> that checks for a response other than <b>No</b> for a question on the Exclusion form. If the response is not <b>No</b> , fire a query. (Use the codelist item from the Data tab.)                          |
| <b>Scope</b>           | Exc1 item                                                                                                                                                                                                                                                      |
| <b>Study structure</b> | Excl (form) <ul style="list-style-type: none"> <li>• Exc1 (item) <ul style="list-style-type: none"> <li>– YesNoCodes (codelist) <ul style="list-style-type: none"> <li>* Yes (codelist item)</li> <li>* No (codelist item)</li> </ul> </li> </ul> </li> </ul>  |
| <b>Rule summary</b>    | <p>evaluate on Form Submission</p> <pre>value = this.Value == this.clYesNoCodelist.citmclNo</pre> <p>when value is false<br/>issue query: The answer to this question indicates that the patient is not eligible for the study. Please clarify or correct.</p> |

## Sample data-entry rule that uses rule model properties

| Characteristic         | Description                                                                                                                                                                                                                                                                                                                                      |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Description</b>     | Create a rule named <code>rulCheckCompletionDate</code> that checks that a study completion date has been entered if <b>Yes</b> is selected for <b>Study complete?</b> . Issue a query when the rule evaluates to true.                                                                                                                          |
| <b>Scope</b>           | Study Completion form                                                                                                                                                                                                                                                                                                                            |
| <b>Study structure</b> | CompletionStatus (form) <ul style="list-style-type: none"> <li>• CompDate (item)</li> <li>• CompletionStatus (item)               <ul style="list-style-type: none"> <li>– YesNoCodes (codelist)                   <ul style="list-style-type: none"> <li>* Yes (codelist item)</li> <li>* No (codelist item)</li> </ul> </li> </ul> </li> </ul> |
| <b>Rule summary</b>    | <pre> evaluate on Form Submission  value = this.itmCompletionStatus.Value == this.itmCompletionStatus.clYesNoCodeList.citmYes ? this.itmCompDate.Empty : false  when value is true issue query on this.itmCompletionStatus: Study completion date cannot be empty, if study completion status is complete. Please verify.</pre>                  |

## Sample data-entry rules that use methods

### Example 1

| Characteristic         | Description                                                                                                                                                                                                                                                                                                                                                         |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Description</b>     | Use a math method to round the result of the existing rule, <code>rulCalcBMI</code> , to two decimal places.                                                                                                                                                                                                                                                        |
| <b>Scope</b>           | Baseline event                                                                                                                                                                                                                                                                                                                                                      |
| <b>Study structure</b> | <ul style="list-style-type: none"> <li>• Baseline (study event)           <ul style="list-style-type: none"> <li>– Demog (form)               <ul style="list-style-type: none"> <li>* HT (item)</li> </ul> </li> <li>– Vitals (form)               <ul style="list-style-type: none"> <li>* Weight (item)</li> <li>* BMI (item)</li> </ul> </li> </ul> </li> </ul> |

| Characteristic | Description |
|----------------|-------------|
|----------------|-------------|

**Rule summary**

evaluate on Form Submission

```
value =
Math.Round(_CalculateBMI(this.frmDemographics.itmHeight.Value,this.frmVitals.itmWeight.Value),2)
```

```
always
set this.frmVitals.itmBMI.Value =
value
```

 **Note:**

The solution above is the simplest solution, but not the best, as it does not clear the BMI field if Height or Weight are subsequently cleared, and it will cause an ISE if Height is 0.0. A better solution follows.

evaluate on Form Submission

```
value = !
this.frmDemographics.itmHeight.Empty
&& !this.frmVitals.itmWeight.Empty
?
(this.frmDemographics.itmHeight.Value
!= 0 ?
1:
(!this.frmVitals.itmBMI.Empty ?
2 : 3)) :
(!this.frmVitals.itmBMI.Empty ?
2 : 3)
```

```
when value == 1
    set this.frmVitals.itmBMI.Value
=
Math.Round(_CalculateBMI(this.frmDemographics.itmHeight.Value,this.frmVitals.itmWeight.Value),2)
when value == 2
```

| Characteristic         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                        | <pre>set this.frmVitals.itmBMI.Empty = true</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Characteristic         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Description</b>     | Create a rule named <code>rulTabsDispensed</code> that checks that the sum of tablets dispensed is greater than zero and less than 500. The sum of tablets should use 0 as a replacement value. If false, fire a query.                                                                                                                                                                                                                                                                  |
| <b>Scope</b>           | Dispensing Record section                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Study structure</b> | <ul style="list-style-type: none"> <li>• DISPREC (section) <ul style="list-style-type: none"> <li>– Tabs (compound item) <ul style="list-style-type: none"> <li>* onemg (item)</li> <li>* twomg (item)</li> <li>* threemg (item)</li> </ul> </li> </ul> </li> </ul>                                                                                                                                                                                                                      |
| <b>Rule summary</b>    | <pre>evaluate on Form Submission  value = (this.itmTabNum.itmonemg.GetValue(0) + this.itmTabNum.itmthreemg.GetValue(0) + this.itmTabNum.itmtwomg.GetValue(0)) &gt; 0 &amp;&amp; (this.itmTabNum.itmonemg.GetValue(0) + this.itmTabNum.itmthreemg.GetValue(0) + this.itmTabNum.itmtwomg.GetValue(0)) &lt; 500  when value is false     issue query on this.itmTabNum: The total number of tablets dispensed {TabsDisp} is not within the expected range. Please clarify or correct.</pre> |



## Sample data-entry rule that uses constants

| Characteristic         | Description                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Description</b>     | Create a rule named <code>rulTempRangeCheckInclusive</code> that checks that the Temperature item is $\geq 36.1$ and $\leq 37.8$ . If not create a query: The entered temperature {EnteredTemp} {EnteredTempUnit} is outside the expected range. Use the TempMin and TempMax constants that you created in the expression. Use the EnteredValue and EnteredUnit rule model properties to construct the query text. |
| <b>Scope</b>           | Temperature item on the Vitals form                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Study structure</b> | <ul style="list-style-type: none"><li>• Vitals (form)<ul style="list-style-type: none"><li>– Pulse (item)</li></ul></li></ul>                                                                                                                                                                                                                                                                                      |

| Characteristic | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Rule summary   | <pre>evaluate on Form Submission  value = Value must be greater than or equal to {MinValue:Constants.TemperatureRan ge.TempMin}, and less than or equal to {MaxValue:Constants.TemperatureRan ge.TempMax}  when value is false issue query: The entered temperature {EnteredTemp} {EnteredTempUnit} is outside the expected range of {TempMin} to {TempMax}  OR evaluate on Form Submission  value = this.Value &gt;= (Constants.TemperatureRange.TempMi n) &amp;&amp; this.Value &lt;=(Constants.TemperatureRange.Temp Max)  when value is false issue query: The entered temperature {EnteredTemp} {EnteredTempUnit} is outside the expected range.</pre> |

 **Note:**

In the first solution, the rule is defined as an intrinsic rule that uses the range check rule template. In the second solution, the rule is defined as a constraint rule that does not use a template.

## Sample data-entry rules that use functions

The `_GetDateDifference` function uses twenty-four hour periods, not calendar days, to compute the differences in days. If the DOV item contained date and time information, there are two possible solutions. The first solution uses the complete DOV date/time and the `_GetDateDifference` function and therefore evaluates days as twenty-four hour periods. The alternate solution evaluates the differences between DOVs using calendar days by creating new `DateTime` objects that contain no Time information from the DOVs and passing those new `DateTime` objects to the `_GetDateDifference` function. Each of the solutions uses the predefined constant `DateTimeParts.Days`.

### Example 1

| Characteristic         | Description                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Description</b>     | Create a rule named <code>rulWeekSeqCk</code> that checks that there are 7 days between Week 1B and Week 2B visits. Issue a query on the Week 2B DOV item (in Treatment Arm B), if the DOV on the Week 2B study event is less than 7 days after the DOV on the Week 1B study event.                                                                                             |
| <b>Scope</b>           | Treatment Arm B study element                                                                                                                                                                                                                                                                                                                                                   |
| <b>Study structure</b> | <ul style="list-style-type: none"> <li>• Week 1               <ul style="list-style-type: none"> <li>– frmDOV (form)</li> <li>* DOV (item)</li> </ul> </li> <li>• Week 2               <ul style="list-style-type: none"> <li>– frmDOV (form)</li> <li>* DOV (item)</li> </ul> </li> </ul>                                                                                      |
| <b>Rule summary</b>    | <pre> evaluate on Form Submission  value = _GetDateDifference (this.evtWeek1B.frmDOV.sctDOV.itmDOV.Value, this.evtWeek2B.frmDOV.sctDOV.itmDOV.Value, Constants.DateTimeParts.Days)&gt;=7  when value is false   issue query on this.evtWeek2B.frmDOV.sctDOV.itmDOV: Date of Visit is less than a week after the Date of Visit given at Week 1. Please clarify or correct.</pre> |

| Characteristic                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Alternate rule summary</b> | <p>evaluate on Form Submission</p> <pre> value = _GetDateDifference (new DateTime(this.evtWeek1B.frmDOV.sct DOV.itmDOV.Value.Year, this.evtWeek1B.frmDOV.sctDOV.itmDO V.Value.Month, this.evtWeek1B.frmDOV.sctDOV.itmDO V.Value.Day, 00, 00, 00), new DateTime(this.evtWeek2B.frmDOV.sct DOV.itmDOV.Value.Year, this.evtWeek2B.frmDOV.sctDOV.itmDO V.Value.Month, this.evtWeek2B.frmDOV.sctDOV.itmDO V.Value.Day, 00, 00, 00), DateTimeParts.Days) &gt;= 7  when value is false issue query on this.evtWeek2B.frmDOV.sctDOV.itmDO V: Date of Visit is less than a week after the Date of Visit given at Week 1. Please clarify or correct. </pre> |

| Characteristic     | Description                                                                                                                                                                                                                                                    |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Description</b> | Create a rule named <code>rulDateCompareWithRange</code> that checks that the DOV in Week 2A is within 6-8 days after the DOV in Week 1A (in Treatment Arm A).                                                                                                 |
| <b>Scope</b>       | Treatment Arm A study element                                                                                                                                                                                                                                  |
| <b>Structure</b>   | <ul style="list-style-type: none"> <li>• Week1A <ul style="list-style-type: none"> <li>– frmDOV (form)</li> <li>* DOV (item)</li> </ul> </li> <li>• Week2A <ul style="list-style-type: none"> <li>– frmDOV (form)</li> <li>* DOV (item)</li> </ul> </li> </ul> |

| Characteristic                | Description                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Rule summary</b>           | <p>evaluate on Form Submission</p> <p>value =<br/>_CompareDatesWithRange(this.evtWeek1A.frmDOV.sctDOV.itmDOV.Value,this.evtWeek2A.frmDOV.sctDOV.itmDOV.Value,Constants.DateTimeParts.Days,6,8)</p> <p>when value is false<br/>issue query on<br/>this.Week2A.frmDOV.sctDOV.itmDOV:<br/>The date of the Week 1A visit and the date of the Week2A visit are not within 6-8 days of each other.<br/>Please clarify or correct.</p> |
| <b>Alternate rule summary</b> | <p>evaluate on Form Submission</p> <p>value =<br/>_CompareDatesWithRange(this.evtWeek1A.frmDOV.sctDOV.itmDOV.Value,this.evtWeek2A.frmDOV.sctDOV.itmDOV.Value,Constants.DateTimeParts.Days,6,8)</p> <p>when value is false<br/>issue query on<br/>this.Week2A.frmDOV.sctDOV.itmDOV:<br/>The date of the Week 1A visit and the date of the Week2A visit are not within 6-8 days of each other.<br/>Please clarify or correct.</p> |

## Sample calculation rules

### Example 1

| Characteristic     | Description                                                                                                                                                                 |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Description</b> | Using the CalcAge function, create a rule named rulCalcAge that calculates a subject's age and populates the age field. Use the DOV from the visit containing the DEM form. |
| <b>Scope</b>       | Baseline study event                                                                                                                                                        |

| Characteristic         | Description                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Study structure</b> | <ul style="list-style-type: none"> <li>• Baseline (study event)               <ul style="list-style-type: none"> <li>– frmDOV (form)                   <ul style="list-style-type: none"> <li>* DOV (item)</li> </ul> </li> <li>– Demog (form)                   <ul style="list-style-type: none"> <li>* DOB (item)</li> <li>* AGE (item)</li> </ul> </li> </ul> </li> </ul>                                                          |
| <b>Rule summary</b>    | <pre> evaluate on Form Submission  value = ! this.frmDOV.sctDOV.itmDOV.Empty &amp;&amp; ! this.frmDemographics.itmDOB.Empty ? 1 :(! this.frmDemographics.itmAGE.Empty ? 2:3)  when value == 1     set     this.frmDemographics.itmAGE.Value     =     _CalcAge(this.frmDemographics.itmD     OB.Value,     this.frmDOV.sctDOV.itmDOV.Value)  when value == 2     set     this.frmDemographics.itmAGE.Empty     = true           </pre> |

Example 2

| Characteristic         | Description                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Description</b>     | <p>Create a rule named <code>rulBMIRangeCheck</code> that checks that the BMI is between 18.5 and 30. Specify an action to issue a query if the BMI is out of range.</p>                                                                                                                                                                                                        |
| <b>Scope</b>           | Baseline event                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Study structure</b> | <ul style="list-style-type: none"> <li>• Baseline (study event)               <ul style="list-style-type: none"> <li>– Vitals (form)                   <ul style="list-style-type: none"> <li>* Weight (item)</li> <li>* BMI (item)</li> </ul> </li> <li>– Demog (form)                   <ul style="list-style-type: none"> <li>* HT (item)</li> </ul> </li> </ul> </li> </ul> |

| Characteristic      | Description                                                                                                                                                                                                                                                                                                        |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Rule summary</b> | <p>evaluate on Form Submission</p> <pre>value = (this.Value &gt;= Constants.BMIRange.BMILow) &amp;&amp; (this.Value &lt;= Constants.BMIRange.BMIHigh)</pre> <p>when value is false<br/>             issue query on this: BMI<br/>             {CalculatedBMI} is out of range.<br/>             Please verify.</p> |

## Sample data-entry rules that use mappings

### Example 1

| Characteristic         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Description</b>     | Create a rule called <code>rulCompletionLaterThanDOV</code> that checks whether the completion date is later than any of the visit dates entered. If not, issue a query.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Scope</b>           | Global; rule is on the Study Completion form, using data from every instance of the DOV form.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Study structure</b> | <ul style="list-style-type: none"> <li>• <code>fmStudyCompletion</code> (form)             <ul style="list-style-type: none"> <li>– <code>itmCompletionDate</code> (item)</li> </ul> </li> <li>• <code>fmDOV</code> (form in each study event)             <ul style="list-style-type: none"> <li>– <code>itmDOV</code> (Item)</li> </ul> </li> <li>• <code>VisitDates</code> (mapping)             <ul style="list-style-type: none"> <li>– <code>VisitDatesDataSet</code> (data set)                 <ul style="list-style-type: none"> <li>* <code>DOVs</code> (data series with <code>itmDOV</code> item mapped using the Always mapping type so the item is available in every form and every study event where it occurs)</li> </ul> </li> </ul> </li> </ul> |

| Characteristic      | Description                                                                                                                                                                                                                                                        |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Rule summary</b> | <p>evaluate on Form Submission</p> <pre>value = _IsValueGreaterThanOrEqualToArray (this.Value,this.VisitDatesDataSet .DOVs.Values)</pre> <p>when value is false<br/>             issue query: Termination Date<br/>         must be later than any visit date.</p> |

Example 2

| Characteristic         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Description</b>     | Create a rule called <code>rulHemoglobinRange</code> that checks whether the entered hemoglobin range is between 140 and 180 for males or between 120 and 160 for females. If not, issue a query.                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Scope</b>           | Global; rule is on the Hematology form, using data from the Demographics form.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Study structure</b> | <ul style="list-style-type: none"> <li>• <code>fmHematology</code> (form)             <ul style="list-style-type: none"> <li>– <code>itmHgb</code> (item)</li> </ul> </li> <li>• <code>fmDemog</code> (form)             <ul style="list-style-type: none"> <li>– <code>itmGender</code> (item)</li> </ul> </li> <li>• <code>Demog</code> (mapping)             <ul style="list-style-type: none"> <li>– <code>DemogDataSet</code> (data set)                 <ul style="list-style-type: none"> <li>* <code>GenderDataSeries</code> (data series with <code>itmGender</code> item mapped)</li> </ul> </li> </ul> </li> </ul> |



| Characteristic      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Rule summary</b> | <pre>evaluate on Form Submission  this.DemogDataSet.GenderDataSeries.Ge nderCodes.Female &amp;&amp; this.Value &gt;= Constants.HgbRanges.HgbLowF &amp;&amp; this.Value &lt;=Constants.HgbRanges.HgbHighF)    (this.DemogDataSet.GenderDataSeries.V alue == this.DemogDataSet.GenderDataSeries.Ge nderCodes.Male &amp;&amp; this.Value &gt;=Constants.HgbRanges.HgbLowM &amp;&amp; this.Value &lt;= Constants.HgbRanges.HgbHighM)  when value is false     issue query: This {EnteredValue} is outside valid hemoglobin range for {EnteredGender}s.</pre> |

---

# 8

## Option descriptions

In this appendix:

- [Rule expressions](#)
- [Functions](#)
- [Constants](#)

### Rule expressions

In this section:

- [New Rule Template dialog box—Option descriptions](#)
- [Rule Templates tab—Option descriptions](#)

### New Rule Template dialog box—Option descriptions

| Option                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Properties tab</b>        | –                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Name                         | Name of the rule template.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Classification               | User-defined term used to organize rule templates.                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Description                  | Description of the rule template.                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Display Text                 | Text that appears in the Rule Summary section of the Rule wizard after the <b>When Value Is</b> information.<br>If this field is blank, the contents of the Expression workspace are used. If the expression contains parameters, the name of the parameter and the value of the parameter appear. For example, if the expression is <b>value must be between {a} and {b}</b> , and the value of <b>a</b> is 10 and the value of <b>b</b> is 100, the parameters appear as <b>a:10</b> and <b>b:100</b> . |
| <b>Definition tab</b>        | –                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Return Type drop-down list   | Return type of the rule template; one of the following: Integer, Float, String, Boolean, Date/Time, or Array (A list of values, all of the same type).                                                                                                                                                                                                                                                                                                                                                    |
| Expression                   | Expression of the rule.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Parameters (Optional)</b> | –                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Parameter                    | Name of the parameter.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Data Type                    | Return type of the parameter; one of the following: Integer, Float, String, Boolean, Date/Time, or Array (A list of values, all of the same type).                                                                                                                                                                                                                                                                                                                                                        |

| Option            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Default Value     | Specified value of the parameter.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>References</b> | –                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Data tab          | Lists study objects in the scope of the rule. Optionally, to view the rule model properties of all of the study objects, select <b>Show all</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Functions tab     | Lists functions registered in a study and the libraries that appear in the Libraries List in the Study Editor. Any rule in the study can reference a function.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Constants tab     | Lists constants created in the study and the libraries that appear in the Libraries List in the Study Editor. Any rule in the study can reference a constant.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Data Mappings tab | Lists: <ul style="list-style-type: none"> <li>• RefNames of the data mappings, data sets, and data series in the study or library.</li> <li>• Rule model properties for data series. A data series has the properties of the item that is mapped to it. If a data series contains an item that collects more than one value, the rule model properties for repeating study objects appear so you can access an array of all of the values of the item.</li> <li>• Methods for data sets. A method appears if you select the corresponding standard data dimension of the data set. You can use data set methods to return a subset of the data in the data set.</li> <li>• Study events, forms, sections, and items that are mapped to each data set.</li> <li>• Study objects appear if you select the corresponding standard data dimension of the data set. The properties of the study objects are used as parameters of data set methods.</li> </ul> |

## Rule Templates tab—Option descriptions

The grid displays rule templates created on the study object selected in the Project Explorer.

| Option                     | Description                                                                                                  |
|----------------------------|--------------------------------------------------------------------------------------------------------------|
| <b>Rule templates grid</b> | –                                                                                                            |
| Data Type                  | Return type of the rule template.                                                                            |
| Description                | Description of the rule template.                                                                            |
| Display Text               | Text that appears in the Rule Summary section of the Rule wizard after the <b>When Value Is</b> information. |
| Id                         | Identification information for the rule.                                                                     |
| Expression                 | Rule expression of the rule template.                                                                        |

| Option        | Description                                                                                     |
|---------------|-------------------------------------------------------------------------------------------------|
| Parameters    | Specified parameters of the rule template, separated by semicolons.                             |
| Template Name | Name of the rule template.                                                                      |
| Type          | Indicates that the rule was created using the Oracle Central Designer rule expression language. |

## Functions

In this section:

- [Functions tab on the Study and Library information Explorer bars](#)
- [Functions tab—Option descriptions](#)
- [Edit Function dialog box—Option descriptions](#)

### Functions tab on the Study and Library information Explorer bars

The Functions tab displays imported, user-defined functions that you can use with rules and rule templates in a study or library. Functions defined in the Functions tab are available study- or library-wide, regardless of the scope of any rules.

A function is a reusable piece of code that extends the behavior of a rule. Many predefined functions are available by default in the System Library.

The Functions tab is available in the following locations:

- Study Information Explorer bar > study node.
- Library Information Explorer bar > library node.

### Functions tab—Option descriptions

| Option        | Description |
|---------------|-------------|
| <b>Fields</b> |             |


#### Note:

Not all fields appear in the default view. Optionally, you can add the other fields to the browser view, and you can rearrange the order of fields. For more information, see *Showing and hiding a field in the User Guide*.

| Option                        | Description                                                                                                                        |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| Class                         | The class in which the function is included.                                                                                       |
| Classification                | User-defined text used to group, sort, and filter functions.                                                                       |
| Data Type                     | Return type of the function, either Integer, Float, String, Boolean, Date/Time, or Array (A list of values, all of the same type). |
| Description                   | Description of the function.                                                                                                       |
| Filename                      | Name of the assembly file that contains the function.                                                                              |
| ID                            | Unique identifier for the function.                                                                                                |
| Language                      | Language in which the function was written.                                                                                        |
| Name                          | Name of the function.                                                                                                              |
| Parameters                    | Parameters for the function.                                                                                                       |
| Published (only in libraries) | Indicates that the study object has been published.                                                                                |
| Title                         | Title of the function.                                                                                                             |
| Type                          | Function type. All functions are .NET class functions.                                                                             |
| <b>Toolbar buttons</b>        | –                                                                                                                                  |
| Import function               | Import an existing function.                                                                                                       |
| Edit                          | Open the selected function to view or edit it.                                                                                     |
| Delete                        | Delete the selected function.                                                                                                      |
| Publish (only in libraries)   | Publish the selected function. This button is enabled for library objects that have been saved.                                    |

## Edit Function dialog box—Option descriptions

| Option                | Description                                                                                                                                                             |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Properties tab</b> | –                                                                                                                                                                       |
| Name                  | Name of the function.                                                                                                                                                   |
| Classification        | User-defined text used to group, sort, and filter functions.                                                                                                            |
| Description           | Description of the function.                                                                                                                                            |
| <b>Definition tab</b> | –                                                                                                                                                                       |
| Return Type           | Data type for the returned value for the function. One of the following: Integer, Float, String, Boolean, Date/Time, or Array (A list of values, all of the same type). |
| Assembly              | Name of the assembly file that contains the function.                                                                                                                   |
| Browse                | Locate the assembly for the function.                                                                                                                                   |
| Class                 | The class in which the function is included.                                                                                                                            |

| Option                                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Parameters section                               | <p>In the parameters section, you can provide one or more parameters for the function.</p> <div data-bbox="1084 363 1377 909" style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>Note:</b></p> <p>Not all fields appear in the default view. Optionally, you can add the other fields to the browser view, and you can rearrange the order of fields. For more information, see <i>Showing and hiding a field in the User Guide</i>.</p> </div> |
| Data Type, Default Value, Description, Parameter | Data type, default value, description, and name of the parameter.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

## Constants

In this section:

- [Constants tab on the Study and Library Information Explorer bars](#)
- [Constants tab—Option descriptions](#)
- [New Constant dialog box—Option descriptions](#)

### Constants tab on the Study and Library Information Explorer bars

In the Constants tab of the Study and Library Editors, you can define constants to be used in rules in the study or library. Constants defined in the Constants tab are available study- or library-wide, regardless of the scope of any rules.

A constant is a value that is defined in a library or study and that can be referenced by any rule.

## Constants tab—Option descriptions

| Option                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Fields</b>                 | <div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>Note:</b></p> <p>Not all fields appear in the default view. Optionally, you can add the other fields to the browser view, and you can rearrange the order of fields. For more information, see Showing and hiding a field in the <i>User Guide</i>.</p> </div> |
| Classification                | User-defined text used to group, sort, and filter constants.                                                                                                                                                                                                                                                                                                                                                                                |
| Data Type                     | Return type of the constant: <ul style="list-style-type: none"> <li>• <b>Integer</b>—Contains only numbers.</li> <li>• <b>Float</b>—Contains numbers and a decimal point.</li> <li>• <b>String</b>—Contains alphanumeric characters.</li> <li>• <b>Boolean</b>—True or False.</li> <li>• <b>Date/Time</b>—Contains date and time information.</li> </ul>                                                                                    |
| Description                   | Description of the constant.                                                                                                                                                                                                                                                                                                                                                                                                                |
| Name                          | Name of the constant. The name must: <ul style="list-style-type: none"> <li>• Begin with a letter.</li> <li>• Contain only letters and digits.</li> <li>• Be unique.</li> </ul>                                                                                                                                                                                                                                                             |
| Published (only in libraries) | Indicates that the study object has been published.                                                                                                                                                                                                                                                                                                                                                                                         |
| Value                         | Value of the constant.                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Toolbar buttons</b>        | —                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| New Constant                  | Create and delete a constant.                                                                                                                                                                                                                                                                                                                                                                                                               |
| Delete                        |                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Publish (only in libraries)   | Publish the selected constant. This button is enabled for library objects that have been saved.                                                                                                                                                                                                                                                                                                                                             |

## New Constant dialog box—Option descriptions

| <b>Option</b>  | <b>Description</b>                                                                                                                                                                                                               |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name           | Name of the constant.                                                                                                                                                                                                            |
| Data Type      | Data type of the constant, such as Integer, Float, String, Boolean, or Date/Time.                                                                                                                                                |
| Value          | Value of the constant. The value must: <ul style="list-style-type: none"><li>• Follow C# variable standards.</li><li>• Start with a letter or an underscore.</li><li>• Contain only letters, numbers, and underscores.</li></ul> |
| Classification | (Optional) User-defined classification of the constant.                                                                                                                                                                          |
| Description    | (Optional) Description of the constant.                                                                                                                                                                                          |