Chapter 6: Adding Expressions

Overview

Expressions are used in Govern OpenForms, version 6.0 and above to perform a wide variety of tasks within a Govern entity, such as:

- Performing a validation
- Performing a mathematical operation
- Performing a computation
- Executing a selection query
- Displaying the value of an unselected attribute on a form
- Attaching a condition to an entity or control; i.e., enabling it or making it visible only if a certain condition is met

The correct syntax must be applied as described in this document.

Expressions can be added at the entity level and on most controls in the OFD. They can also be associated with a calculated value for an attribute or a validation rule in the Business Entity Designer. Expressions that are added to the OFD are not saved in the database.

This section describes the use of expressions in Govern. It contains the following topics:

General Procedures for Adding Expressions

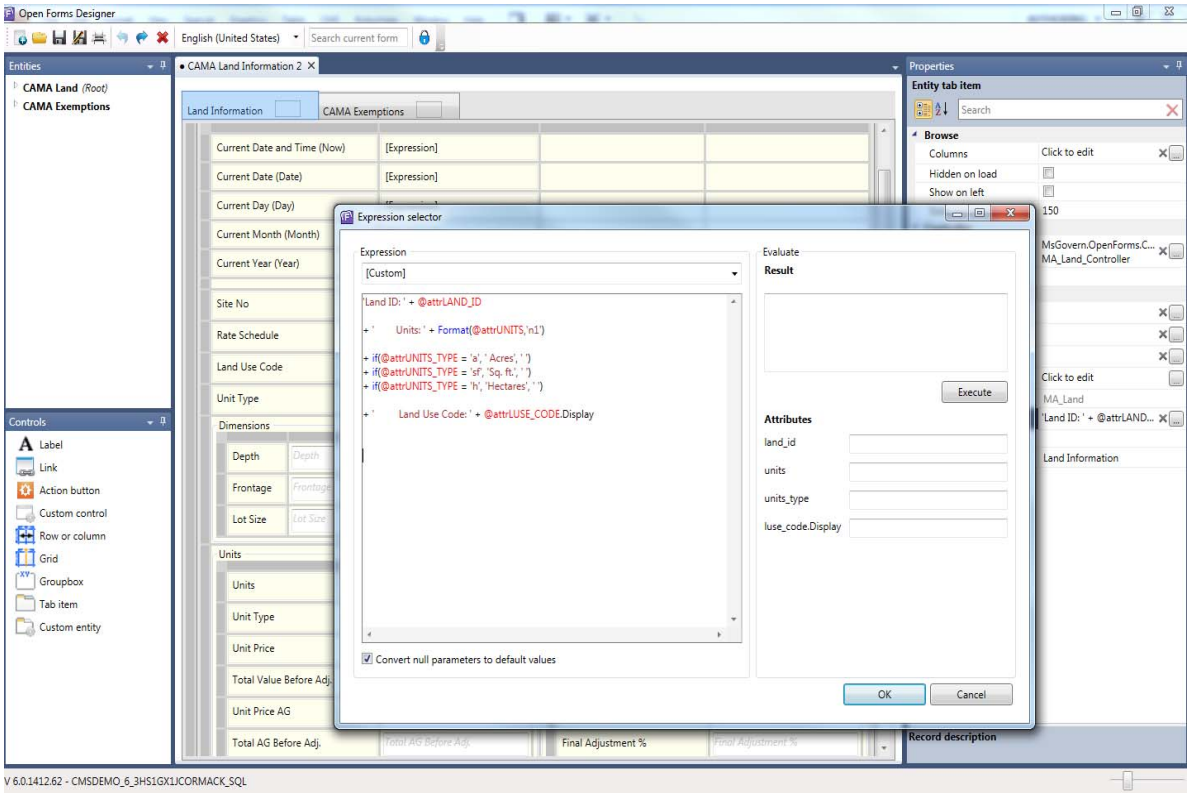
Overview

Expressions are added through the Expression Selector. This is available in the Properties Explorer for the entity in the OFD and for most controls. The procedure is similar. You can also refer to the Govern Business Entity Designer documentation for details on using an expression as a validation rule, or adding an expression to a calculated field.

To add an expression:

1. Launch the OFD.
2. Open the applicable form and entity in the OFD Editor.
3. Select the control or item to which you want to add the expression, such as a label, a link, a groupbox, a tab, or an action item.
4. Select one of the following properties, depending on the item and the type of expression that you want to add:
   - Expression: for attaching any expression to the item.
     This is available for labels.
   - Is enabled: to disable the item or to enable it only under a specified condition.
   - Is visible: to hide the item or to show it only under a specified condition.
5. Click the ellipsis button in the text box of the property you selected in step 4.

   This opens the Expression Selector.



6. Enter the expression.

   For example, the expression in the preceding screen shot could be used to enable or show the selected property only if the Effective Year Built of a building is greater than 2000.

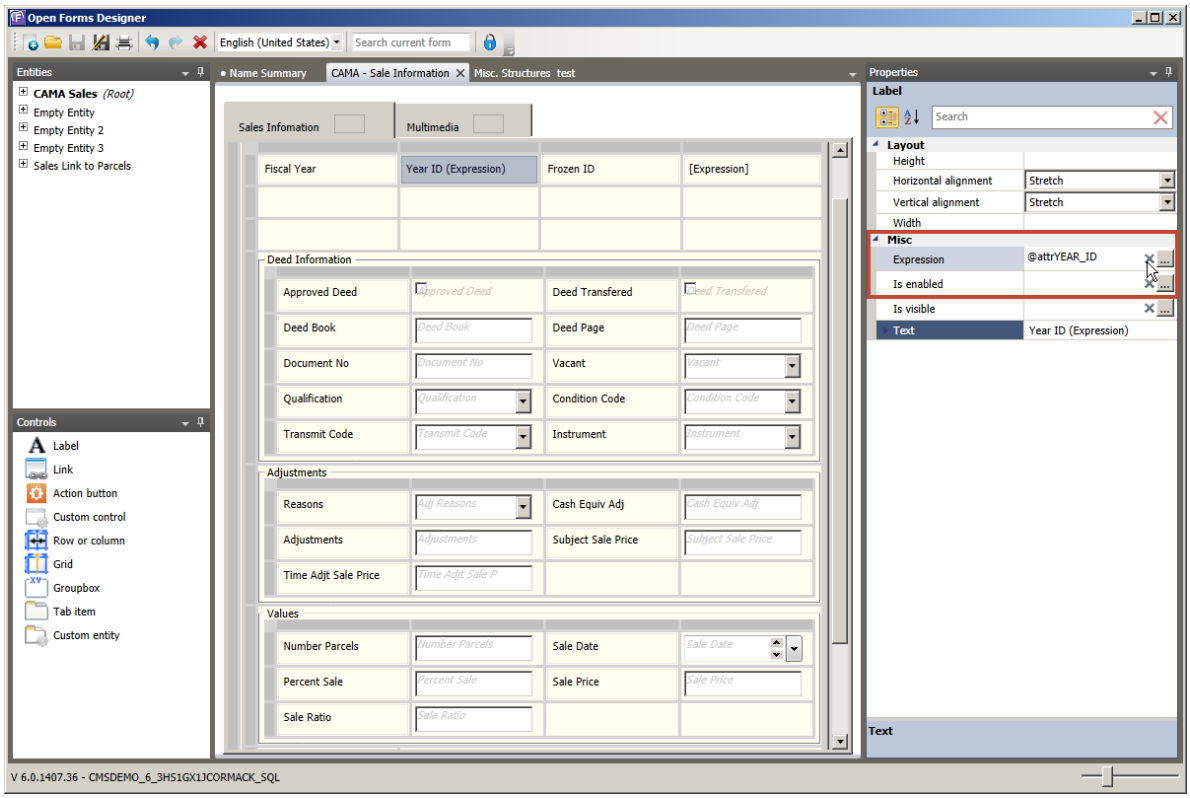
7. Click **OK**.

The Expression Selector is also available for the calculated value of an attribute and validation rules added in the Business Entity Designer. *Refer to the Govern Business Entity Designer (BED) guide for details*.

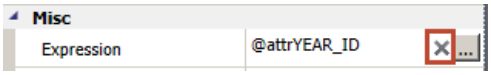
Deleting an Expression

To delete an expression:

1. Launch the OFD.
2. Open the applicable form and entity in the OFD Editor.
3. Select the item with the expression on the editor.



4. Do one of the following:
   - Click the **Delete** icon in the Expression property text box to delete the expression.

   - Click the **Delete** icon on the toolbar to delete the whole item.

5. Click **Save**.

Types of Expressions

Overview

Several types of expressions can be used, including expressions that display an attribute or Govern ID, Govern expressions, standard expressions, formulas, and queries. These are described in the following topics:

Showing an Attribute or Govern ID

You can add an expression that displays an attribute or Govern ID. This can be useful if you want to show an attribute or Govern ID that does not appear on the current form or the current tab. You can also make it easy for users to select a record by displaying an attribute in the Record Selector at the top of the form.

This section describes the following:

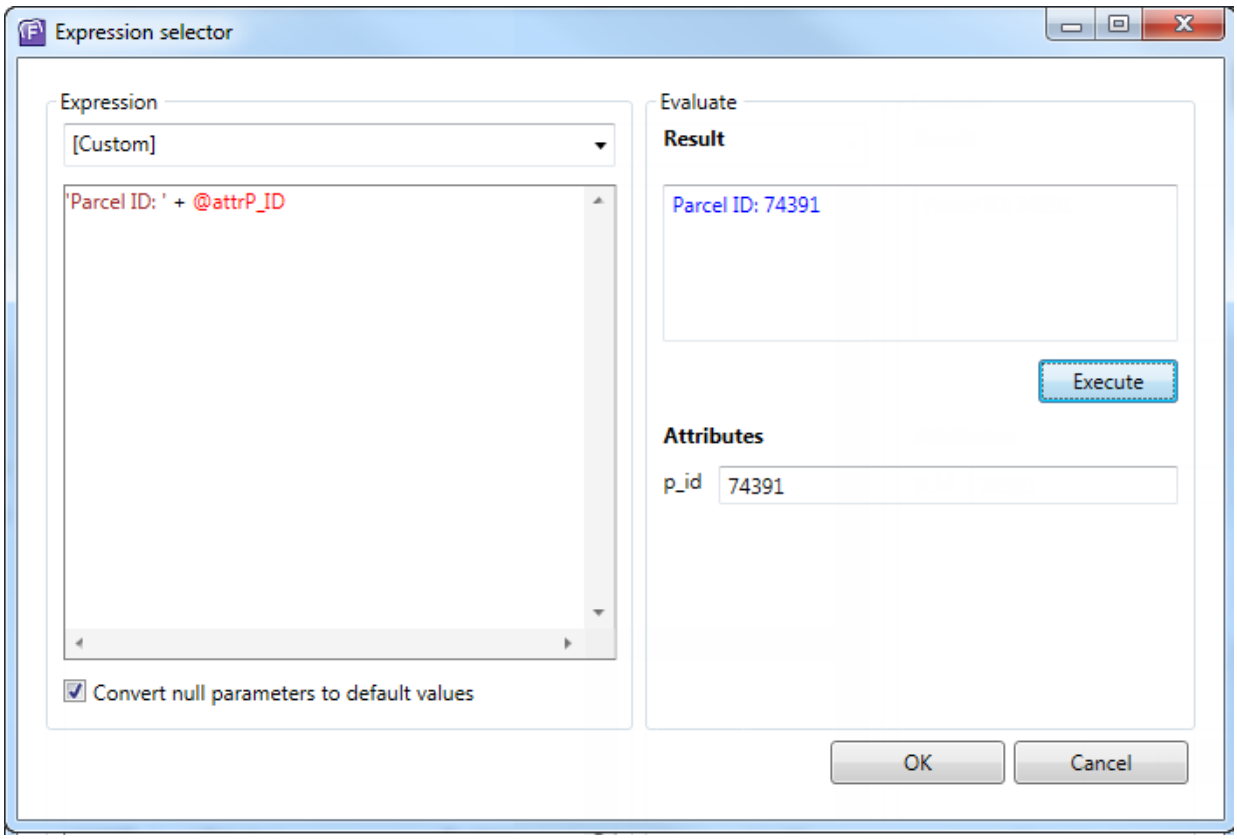
Syntax

The syntax to show an attribute or Govern ID is as follows:

```
@ AttributeName; for example, @attrADJ_TOTAL

or @GovernIDName; for example, @attrFROZEN_ID
```

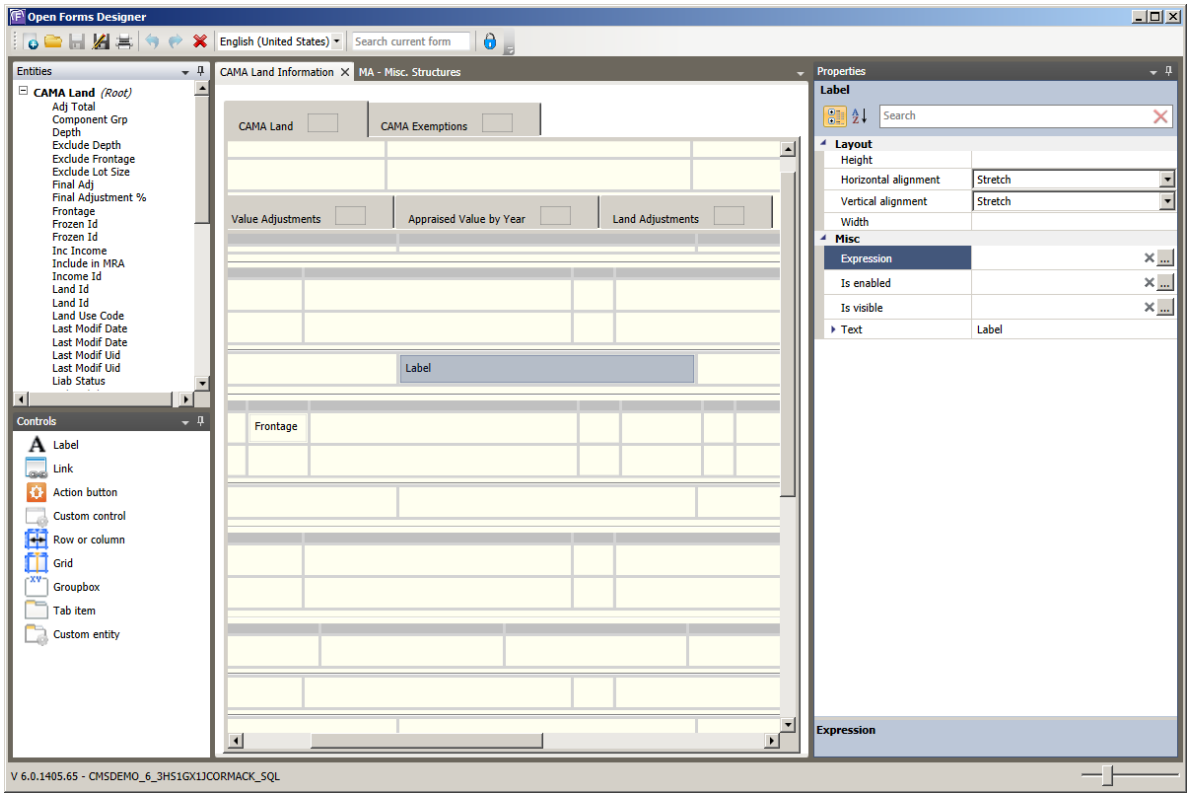
Showing an Attribute or ID on a Label

In the following screen shot a label is added to a form to display the Parcel ID:

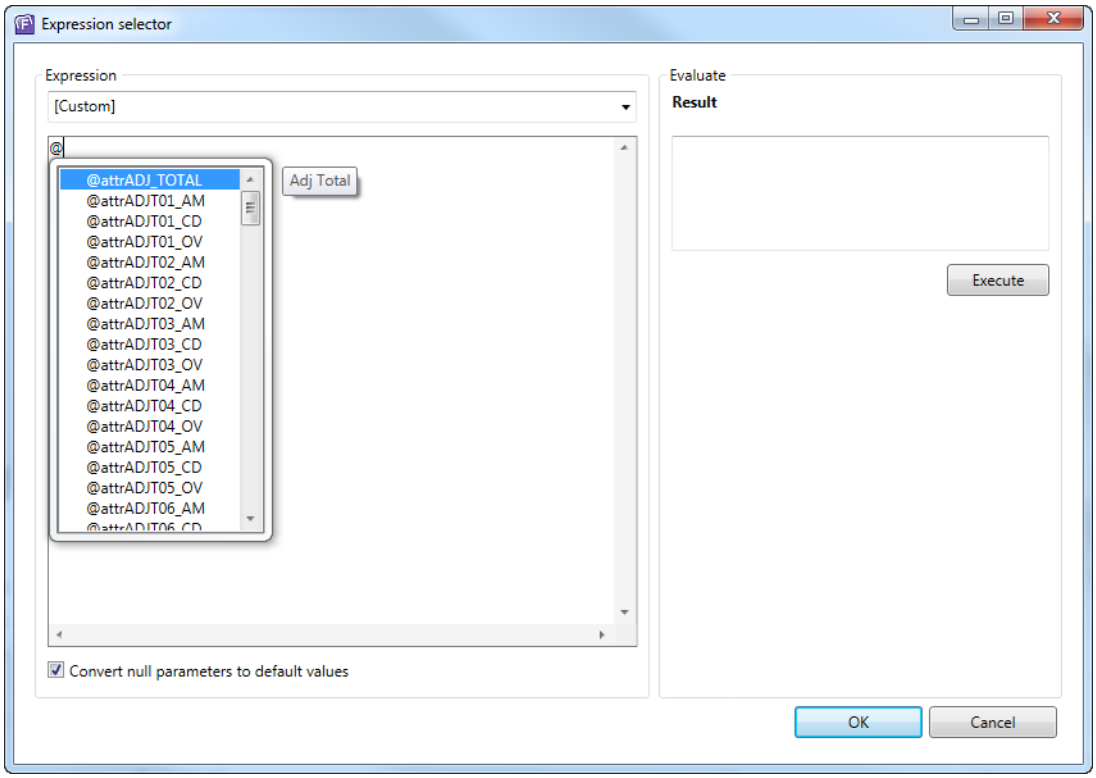


To write the expression:

1. Open the form and the entity in the OFD.
2. Add a label to the entity.



3. Select the **Expression** property in the Property Explorer.
4. Click the ellipsis button ... to open the *Expression Selector*.



5. Hit the commercial at-sign (@) on your keyboard.

   A drop-down list displays the following:
   - All the attributes in the entity, including those not selected for the form
   - All Govern IDs in the entity

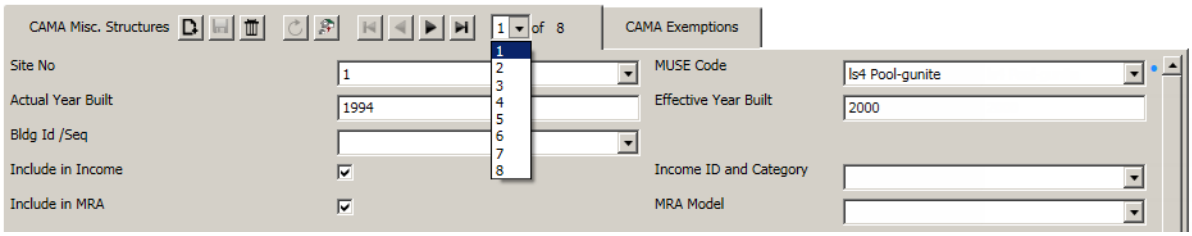
   As you scroll through the list, a tooltip displays the name of each attribute and ID.

6. Select an attribute or Govern ID in the list.
7. Click **OK** on the Expression Selector.
8. Click **Save** on the OpenForm Designer.

The result of the expression is displayed in Govern the next time you open the form. If the form is already open, click the **Refresh** button in Govern

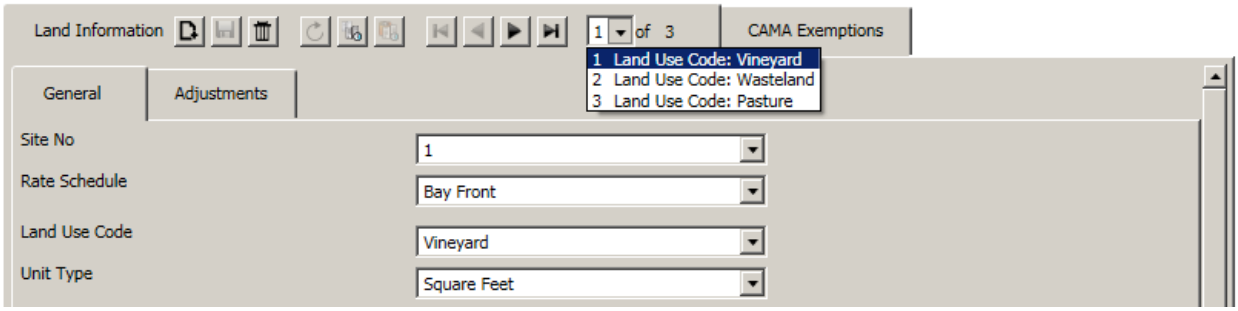
Showing an Attribute in the Record Selector

By default, the Record Selector at the top of the Govern forms displays only the number associated with the record. This is illustrated in the following screen shot.



You can add a meaningful description, with an expression, to the number and make it easier for your users to find the record they want.

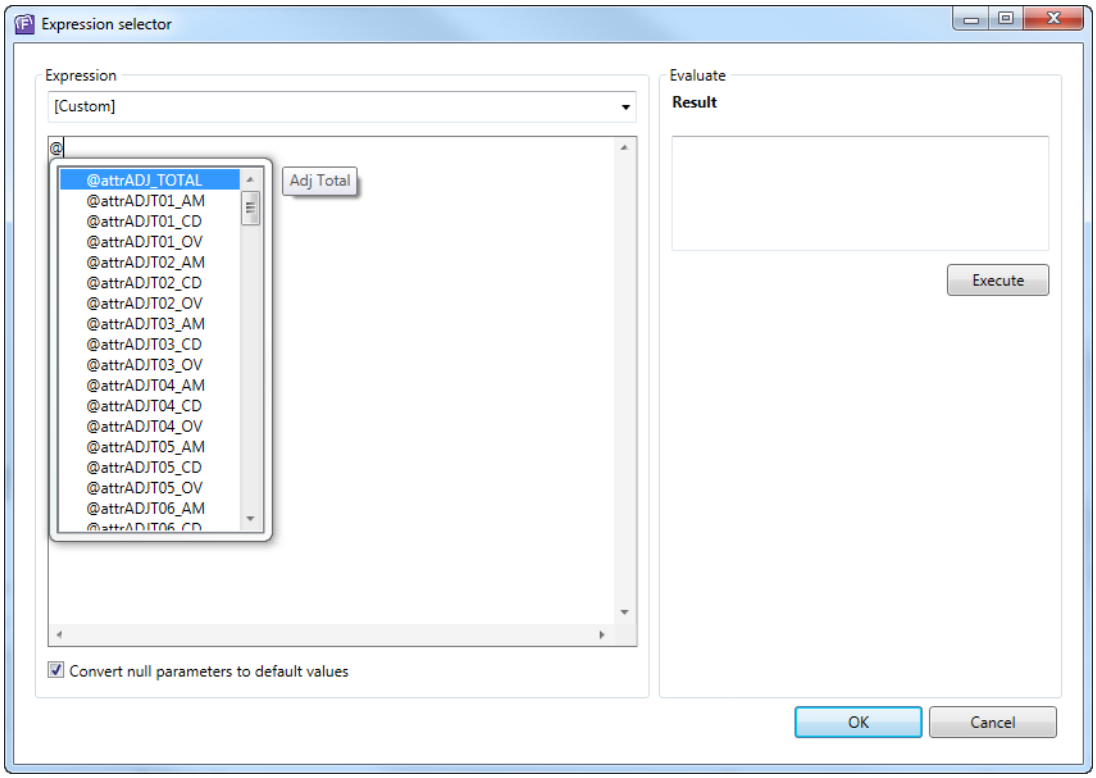
For example, you could show one or more key attributes or values. For example, by displaying the Total Land Value on the CAMA Land Information, you can make it easy for users to select all records that have a specific appraisal value.



To display the value of an attribute in the Record Description:

1. Open the entity in the Govern OpenForms Designer.
2. Select the entity to view the properties for the entity in the Property Explorer.
3. Select the **Record Description** property in the Property Explorer.
4. Hit the commercial at sign (@) on your keyboard.

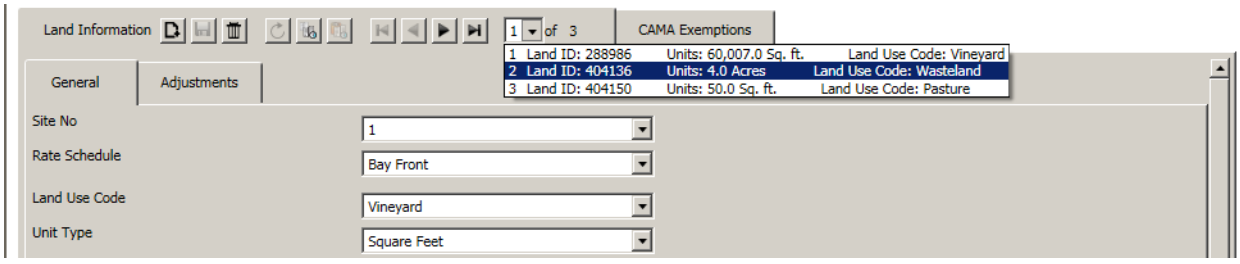
   A drop-down list displays the following:
   - All the attributes in the entity, including those not selected for the form.
   - All Govern IDs in the entity



5. Select an attribute or Govern ID in the list.
6. Click **OK**.

Showing Multiple Attributes in the Record Selector

You can display multiple attributes in the record description that is displayed in the record selector.



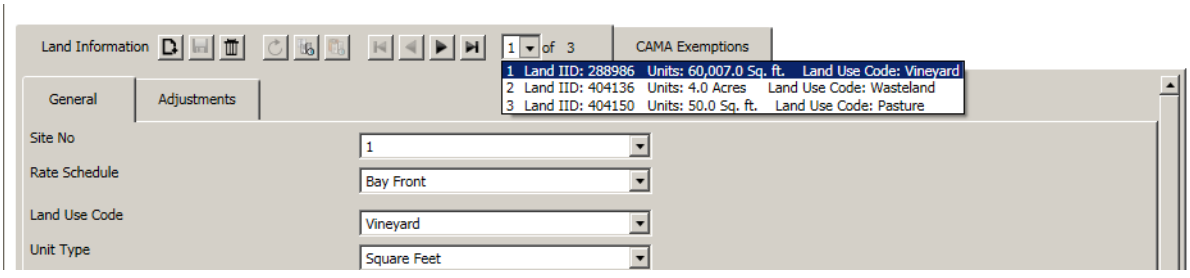
To do this, use the plus sign (+) between each expression type or part of the expression. The following example illustrates how to use the syntax.

```
'Land IID: ' + @attrLAND_ID

+ '   Units: ' + Format(@attrUNITS,'n1')

+ if(@attrUNITS_TYPE = 'a', ' Acres', ' ')
+ if(@attrUNITS_TYPE = 'sf', 'Sq. ft.', ' ')
+ if(@attrUNITS_TYPE = 'h', 'Hectares', ' ')

+ '   Land Use Code: ' + @attrLUSE_CODE.Display
```

In this example:

- A description is added to each attribute. Text descriptions are enclosed in single quotation marks (‘).
- The Format function is used in order to display values and amounts more clearly. *For details about the Format function, see Applying Formatting to Expressions on page 269*.
- If clauses are used in order to show descriptions of the unit type. In this case, these are based on the selection in the Unit Type text box. If the unit type is acres, show acres. If the unit type is square feet, show square feet, etc.
- The Display syntax is used in order to show the description associated with the Land Use Code. *For details, see Displaying the Field Description on page 274*.

The result of this expression is shown in the following screen shot:



Adding a Govern Expression

The Govern expressions are:

- IsNull

- Substr
- Case

These can be included in a longer expression or used as a stand-alone expression:

IsNull. For example, IsNull (@attrP_ID, 0) returns 0 if the parcel ID (P_ID) is null.

IsNull is a requirement for all expressions that could return Null values.

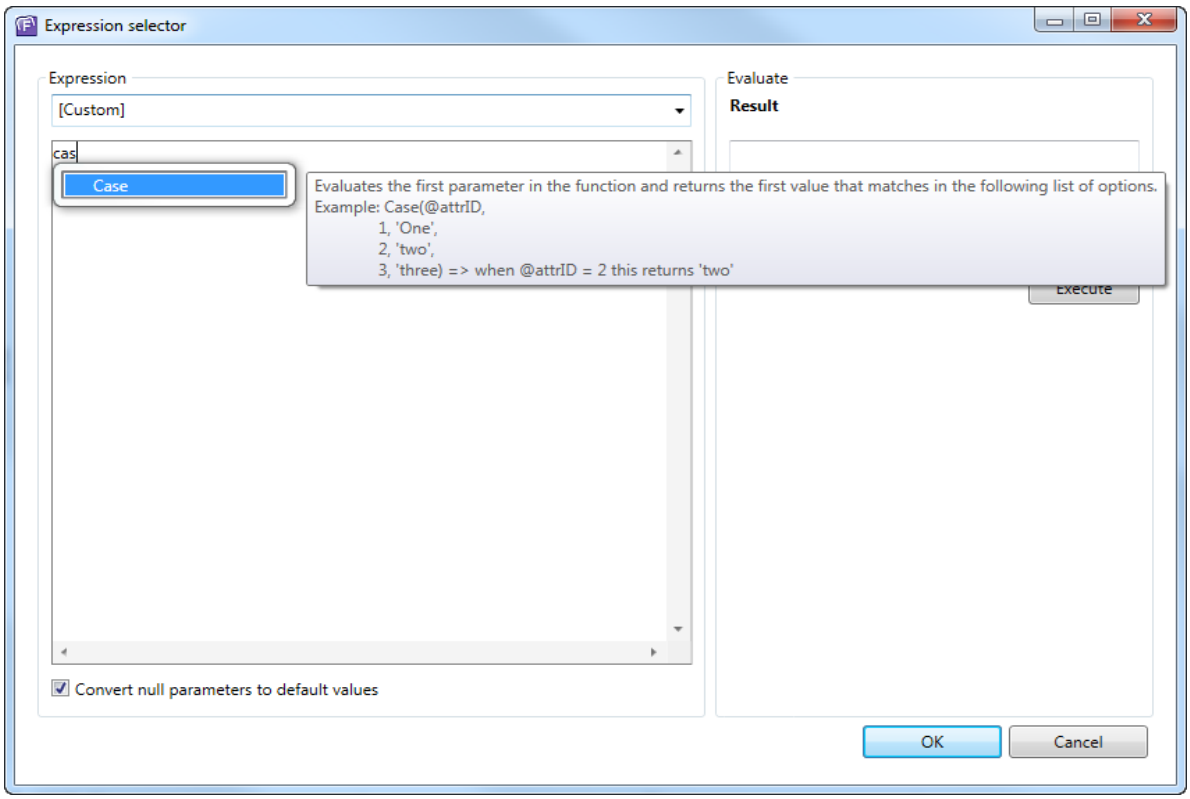
For example, the following expressions show how the Building DIR< GRM, and MRA rates are calculated. IsNull is added to include records that do not have these values.

BUILDING_DIR:
ISNULL(@attrINCOME_DIR_VALUE, 0) - ISNULL(@attrLAND_VALUE, 0)
BUILDING_GRM:
ISNULL(@attrINCOME_GRM_VALUE, 0) - ISNULL(@attrLAND_VALUE, 0)
BUILDING_MRA:
ISNULL(@attrMRA_VALUE, 0) - ISNULL(@attrLAND_VALUE, 0)

Substr. For example, SUBSTR('MSGovern', 3, 6) returns six characters from the third position of the string 'MSGovern'. It returns Govern.

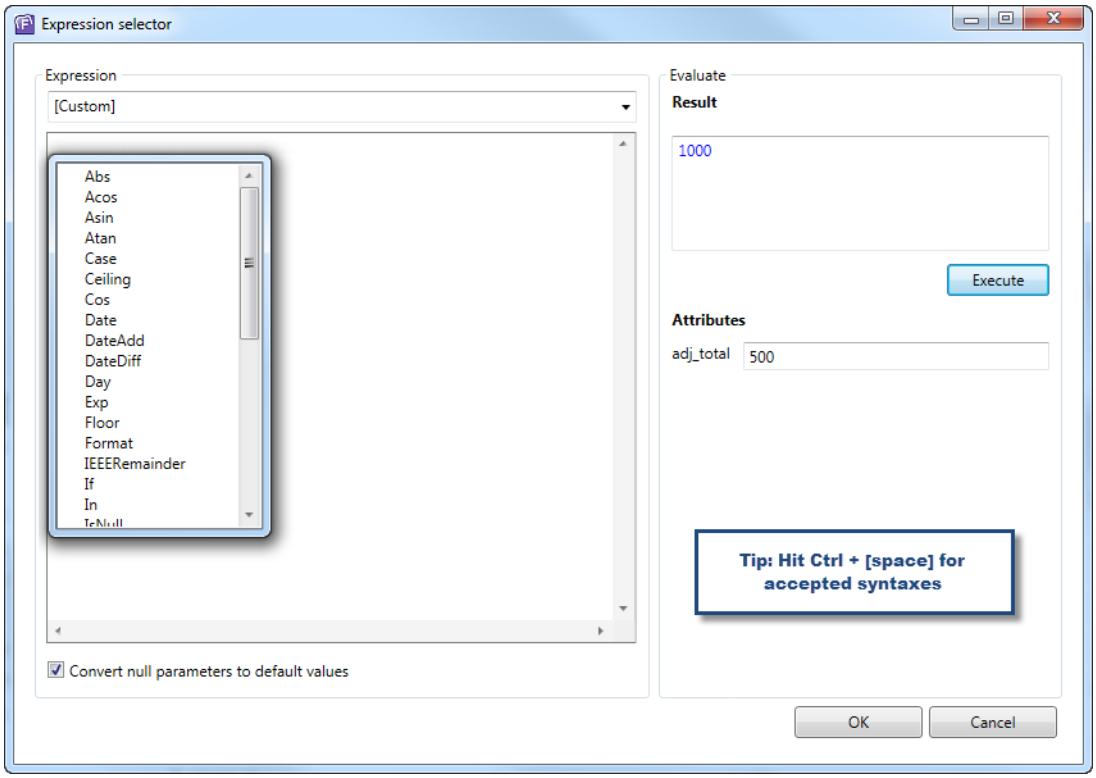
Case. For example, Case(@idYEAR_ID,
2000, 0,
2001, 1,
2002, 2)
Returns 0 if the YEAR_ID is 2000, 1 if it's 2001, or 2 if the year is 2002.

When you begin entering an expression in the *Expression Selector*, a tooltip appears displaying the correct syntax for the expression and provides an example. This is illustrated in the following screen shot:



Alternatively, you can hit the Control key [Ctrl] then the space bar on your computer keyboard to view a list of all the accepted expression syntaxes.

```
[Ctrl] + [space]
```



Converting Logical Expressions or Formulas

If you have an existing user form with a logical expression or formula, you need to convert it to the new syntax before using it in Govern OpenForms. Otherwise, a message stating that the logical expression or formula is not supported appears in the log files.

Adding a Query

A query can be added to an expression in order to select data from the Govern database and display it on a user form or in the Govern Ribbon. The following rules apply:

> As a best practice, write queries that return only one row and one column.

- The query must be a Selection Query and must be predefined in the Select Queries Editor in GNA.

- If multiple results can be retrieved from a query, the first column and first row are returned.
- A query can include a cache value.
- The default cache value for a query is five minutes. You can change this, as described in the next section: Syntax for Queries on page 250. The maximum cache value is 720 minutes (12 hours).
- A query can include a column name.

  By specifying the column names, you can use the same query multiple times on a form and return multiple values. This increases performance over writing multiple queries to return different values from the same table. *For a scenario illustrating this, see Adding the Column Name to a Query to Display Multiple Values on page 253*,

Syntax for Queries

The section provides the syntax for queries. Examples follow:

Basic Query

`Query('QueryName')`:

Returns the value in the first row and first column of the query results.

Basic Query with Specific Cache

```
Query('QueryName', n)
```

Returns the value in the first row and first column of the query results. n is the number of minutes that a cached value remains valid, with a maximum cache of the specified number of minutes.

Note: The maximum cache value for a query is 720 minutes (12 hours).

Basic Query with Specific Column

`Query('QueryName', 'ColumnName')`:

Returns the value in the first row for the specified column from the query results.

Basic Query with Specific Column and Cache

```
Query('QueryName', 'ColumnName', n)
```

Returns the value in the first row for the specified column from the query results, with a maximum cache of the specified number of minutes.

Examples of Queries

The following query returns the Parcel ID (P_ID) from PC_PARCEL.

```
Select * from PC_PARCEL
where P_ID = P_ID
```

This query is predefined in the Select Queries Editor in GNA.

The following list of examples illustrates the different ways that this query can be added to an expression.

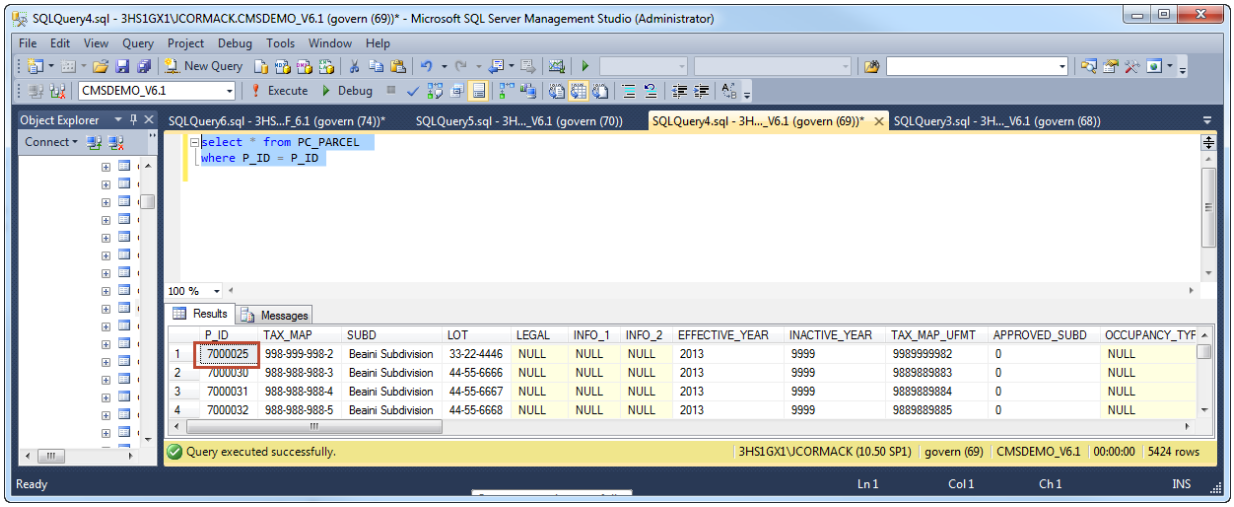
Example 1: Basic Query

Expression:

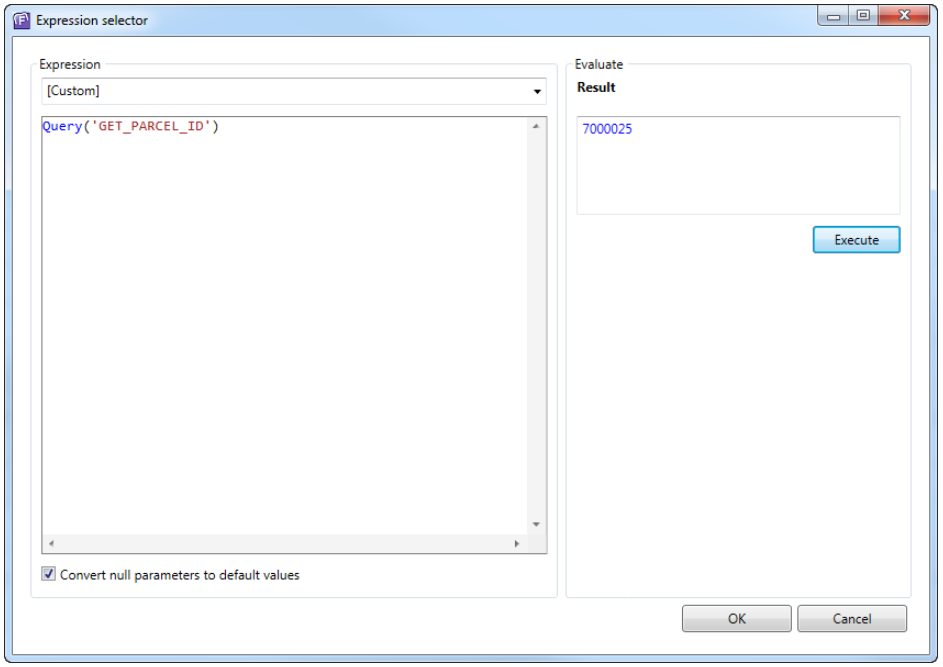
```
Query('GET_PARCEL_ID')
```

Results:

This expression returns the first row and first column from the table: PC_PARCEL



The result can be viewed in the Expression Selector:



Example 2: Basic Query with Cache

Expression:

```
Query('GET_PARCEL_ID', 2)
```

Results:

The results of this expression are the same as the first example, but the results are valid for a maximum of two minutes.

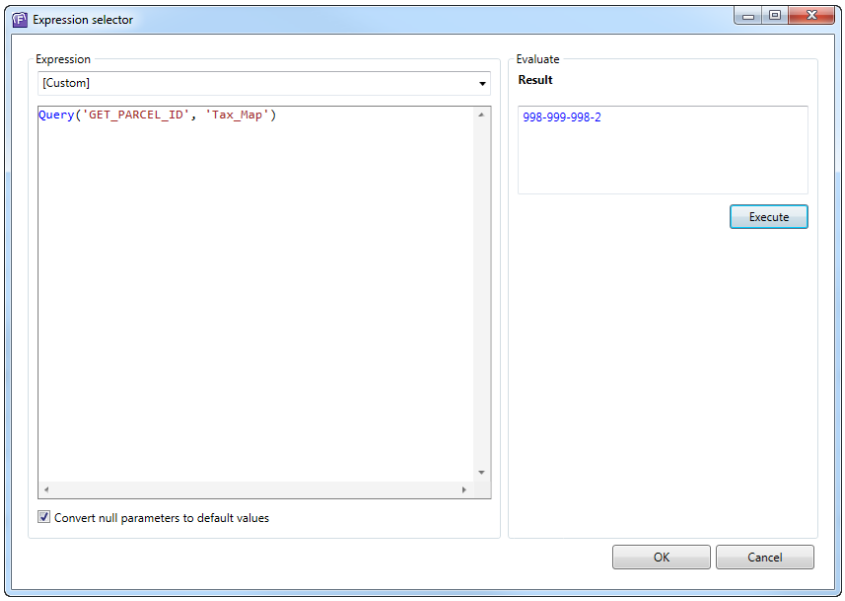
Example 3: Basic Query with a Specified Column

Expression:

```
Query('GET_PARCEL_ID', 'Tax_Map')
```

Results:

The expression returns the tax map number from the first row of PC_PARCEL. You can view the results in the Expression Selector.



Example 4: Basic Query with a Specified Column and Cache

Expression:

```
Query('GET_PARCEL_ID', 'Tax_Map', 2)
```

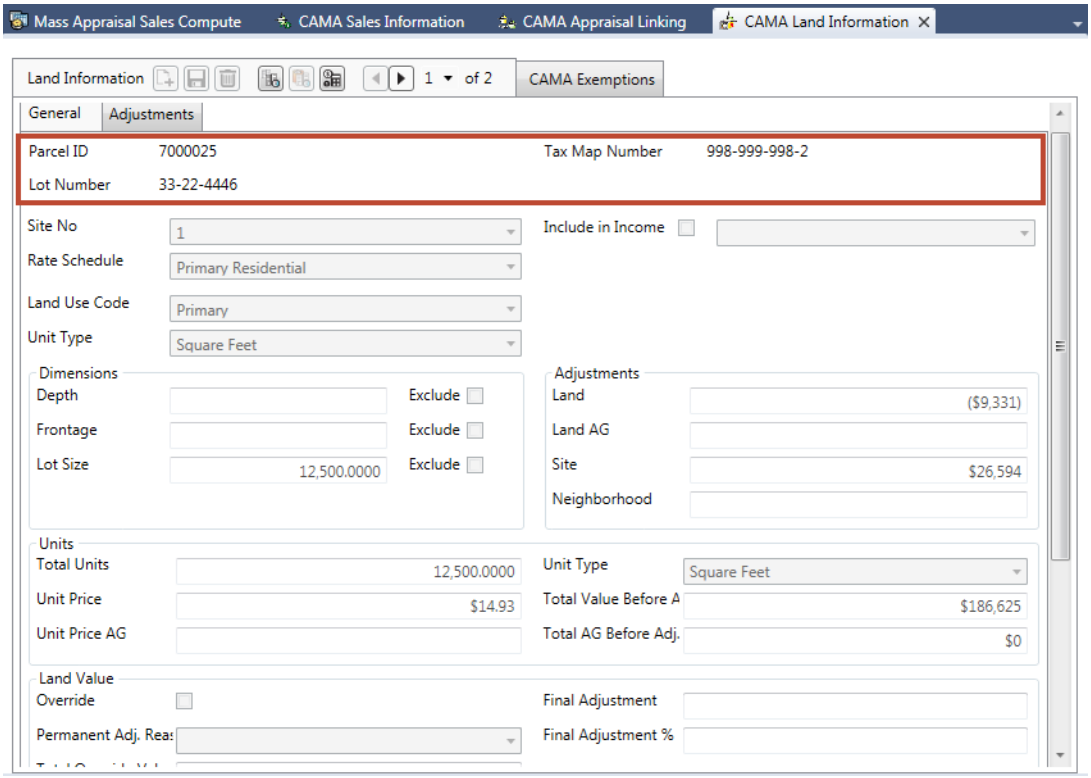
Results:

This expression returns the tax map number from the first row of PC_Parcel. The result is valid for two minutes.

Adding the Column Name to a Query to Display Multiple Values

In this scenario, the tax map number, parcel ID, and lot number are added to the Land Information form. These data are all found in the table PC_PARCEL. They can be returned with the query GET_PARCEL_ID.

```
Select * from PC_PARCEL
where P_ID = P_ID
```



To configure this scenario:

1. Launch the OFD.
2. Open the Land Information form, or any user form.
3. Drag a label to the top of the form.
4. Enter **Parcel ID** for the **Text** property.
5. Drag a second label beside it.
6. Click the ellipsis button ... in the **Expression** property.
7. Enter the expression:

```
Query('GET_PARCEL_ID")
```

8. Repeat steps 3 to 7, with the following entries:
9. Enter **Tax Map Number** in the **Text** property.
10. Enter the expression:

```
Query('GET_PARCEL_ID', 'TAX_MAP')
```

11. Enter **Lot Number** in the **Text** property.
12. Enter the expression:

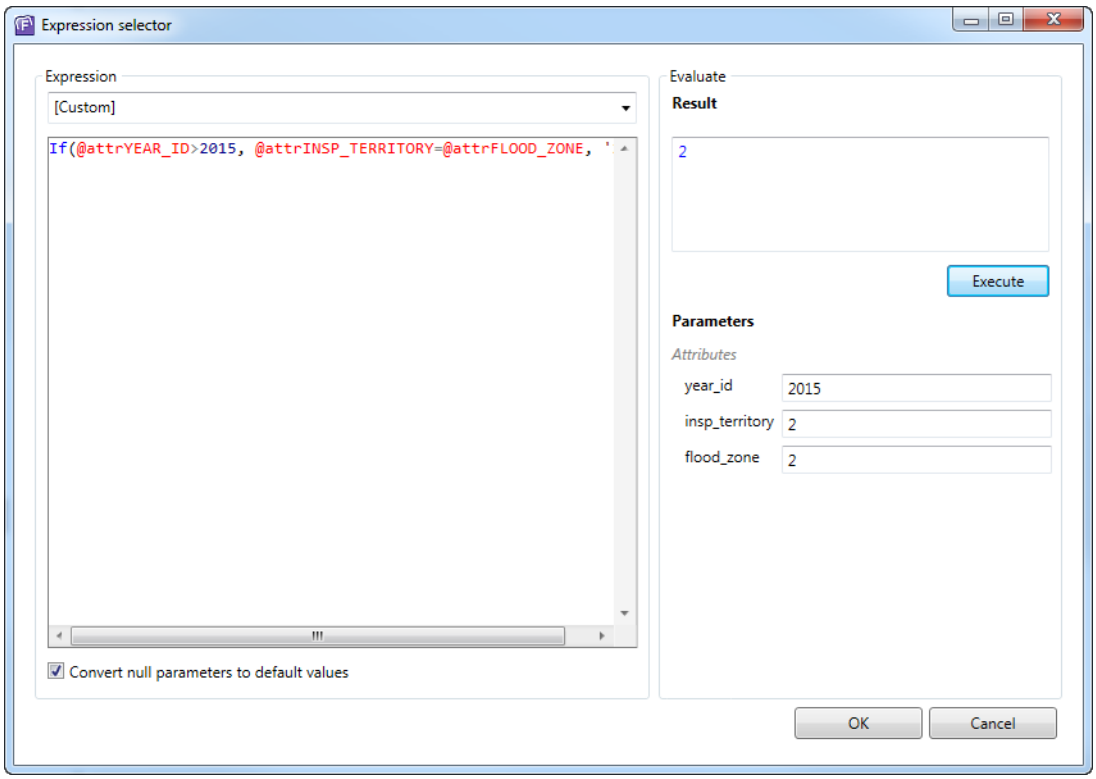
```
Query('GET_PARCEL_ID', 'LOT')
```

13. Click **Save**.

You can verify each expression with the Evaluate options on the Expression Selector.

To verify your expression:

1. Enter your expression in the Expression Selector.
2. If any parameters appear on the Expression Selector, enter values in the **Parameter** fields.
3. Click the **Execute** button.



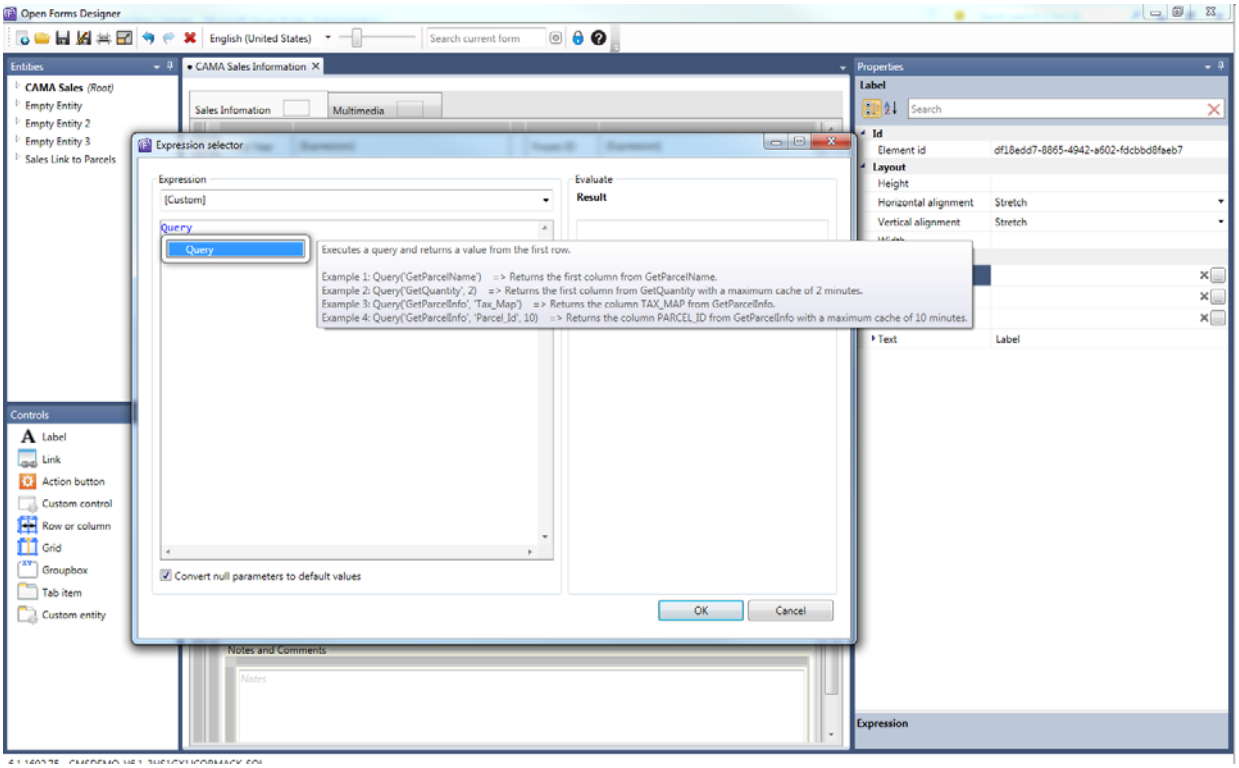
One of the following occurs:

- If the expression is valid, the result appears in the **Result** field.

- If the expression is not valid, an error message appears in the **Result** field.

4. Do one or the following:
   - Click **OK** to save the expression.
   - Click **Cancel** to close the Expression Selector without saving.

Tooltip



If multiple records are returned for the query, the result from the first row is returned.

The query returns a value and saves this value to the cache. The value in the cache is used as long as it is no older than the number of minutes specified in the query.

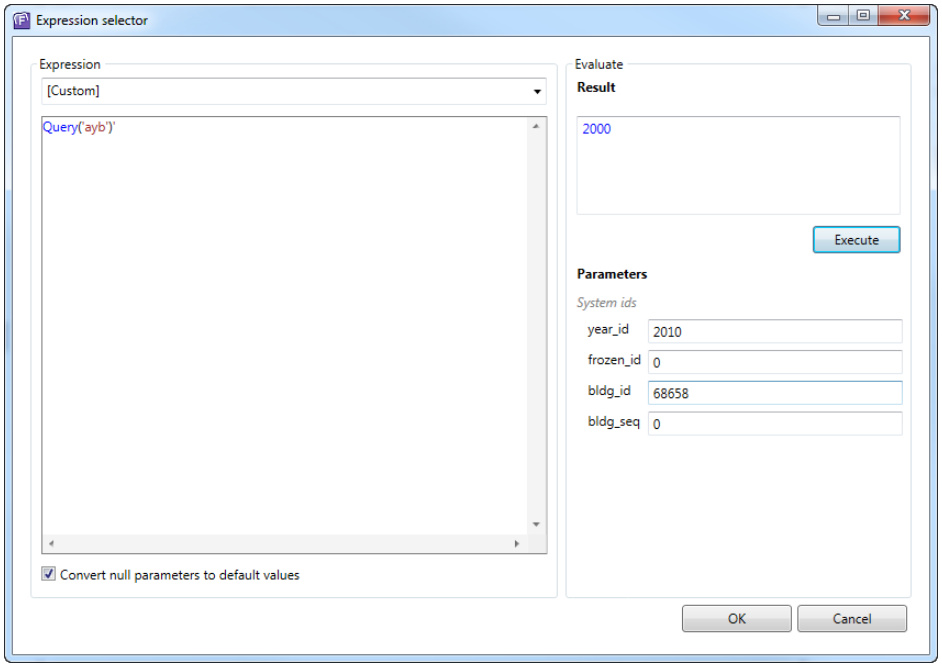
The example, Query('GetQuantity', 2), would use a value that is saved in the cache for two minutes or less.

Testing a Query

As with all expressions, you can test the logic of the query used in the expression with the Evaluate feature. When you enter an expression, the parameters that the query contains are listed on the right under evaluate.

To test the expression:

1. Enter the expression in the Expression Selector.



2. Enter values for the parameters in the **Parameters** text boxes.

   In this case, the parameters are system IDs.

3. Click **Execute**.

The result of the expression is displayed in the **Result** test box. Possible results include the following:

- Null if parameters are missing.
- The value returned by the query.

For example, the query displayed in the screen shot returns the actual year built for the specified year, frozen ID, building ID, and building sequence.

```
SELECT YEAR_BUILT FROM MA_BUILDINGS WHERE YEAR_ID =
@idyear_id AND FROZEN_ID = @idfrozen_id AND BLDG_ID =
@idbldg_id AND BLDG_SEQ = @idbldg_seq
```

All system IDs must be entered for this query.

Other parameters contained in the query could include constants and attributes.

Note: The parameters must be included in the entity that is selected for the query.

Note: If you are using a query in an expression and that query retrieves a value, ensure that the value is up-to-date. It is recommended to run the Mass Appraisal Batch Compute before running queries that affect rates and values.

The following example illustrates this point.

The query named GetSalesCount is used to retrieve the number of property sales:

```
SELECT COUNT(*)
FROM PC_LK_PARCEL_SALE
WHERE P_ID= @attrP_ID
```

For the land, in this example, a rate is defined by the following expression:

```
{ SalesCount = Query('GetSalesCount' }
If(SalesCount > 1000, 10.5, 12.5)
```

If the sales count is greater than 1,000, the applicable rate is 10.5. Otherwise it's 12.5. If one new sale is added when you have 1,000 sales, the rate changes and consequently this changes the value of the property. However, the property would not be recomputed automatically, in this case.

Adding a Mathematical Expression

The following links provide lists and descriptions of all the expression types that are supported:

http://ncalc.codeplex.com/wikipage?title=functions&referringTitle=Home

http://ncalc.codeplex.com/wikipage?title=operators&referringTitle=Home

In order to view the tooltips for these expressions, you need to enter the first few letters that match the type of expression. Then, select the expression from a drop-down list.

Adding an Expression to a Govern Form

Overview

An expression can be added to a Govern form in the OFD as described under the following headings. Also, an expression can be added to an attribute in the Business Entity Designer (BED). If it is added in the BED, it applies to all copies of the form.

In the OFD, expressions can be added at various levels, as described in the following topics:

- Adding an Expression at the Entity Level on page 260
- Adding an Expression to a Control on page 262

Adding an Expression at the Entity Level

You can add an expression to the save, delete, and insert functionality of an entity.

You can write an expression to set these properties to false or to associate a condition with the functionality.

Note: You can also define security for the icons that control these functionalities on the form. If both an expression and security permissions are defined, the definition that is the most restrictive replaces the other.

For example, the following expression, can be added to the **Allow Save** property for the CAMA Land entity.

```
IsNull(@attrFINAL_ADJ_PERCENT,0) = 5
```

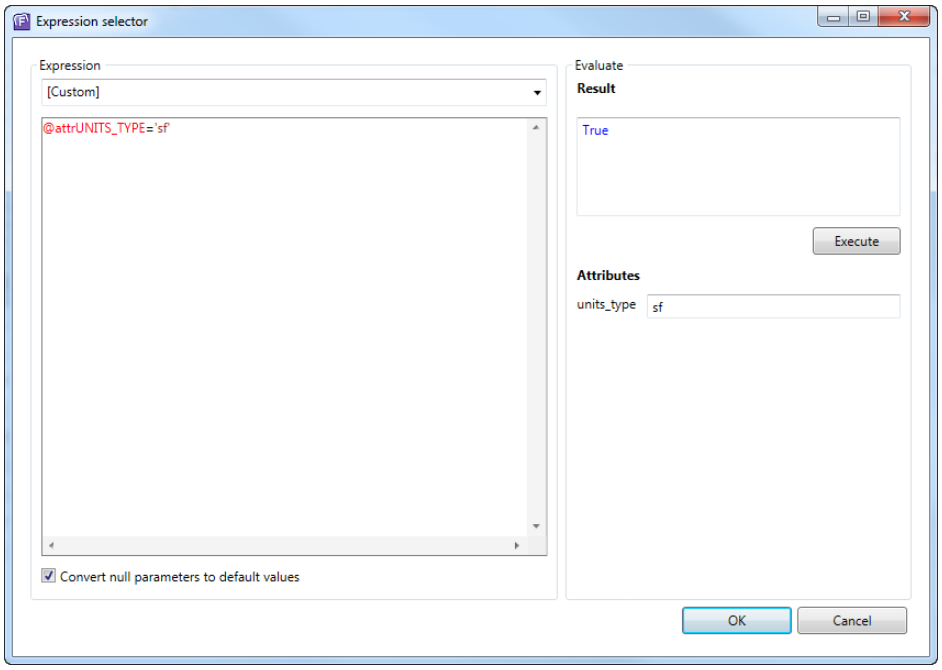
This would disable the Save button unless the Final Adjustment Percentage for the land is equal to five.

If the **Save** button were disabled in the security settings, then this would override the expression. Not being able to save is more restrictive than only being able to save under a specific condition.

To add an expression to the delete, insert or new, or save functionality of the entity:

1. Launch the OFD.
2. Open the entity.
3. Expand the **MIsc** properties in the Property Explorer.
4. Click the ellipsis button ... beside one of the following properties:
   - **Allow delete**: to write an expression for the **Delete** button
   - **Allow insert**: to write an expression for the **New** button
   - **Allow save**: to write an expression for the **Save** button

   The *Expression Selector* opens.



5. Enter the expression.
6. Click **OK**.

Results

If you enter the word **False**, in the *Expression Selector,* for:

- **Allow delete**: the **Delete** button is disabled in Govern.

- **Allow insert**: the **New** button  is disabled in Govern.
- **Allow save**: the **Save** button  and all data entry fields are disabled in Govern.

You could also write an expression that attaches a condition for allowing or disabling any of these functionalities.

Adding an Expression to a Control

You can add an expression to the following controls:

- Label
- Link
- Action button
- Custom control
- Groupbox
- Tab item

Expressions at this level are used to set conditions for enabling the control or making it visible. As with the expressions at the entity level, you can define security for the control. The same rule applies: the most restrictive definition is used. As illustrated in the following example, you could write an expression to make the Land Adjustments tab visible only when the unit type is set to square feet. You could then add a security permission to hide the tab from the Clerks role at all times. In this case, the security permission would apply for the role because it is more restrictive.

IsVisible

The following expression can be added to the **Is Visible** property of a tab in the CAMA Land entity.

```
@attrUNITS_TYPE='sf'
```

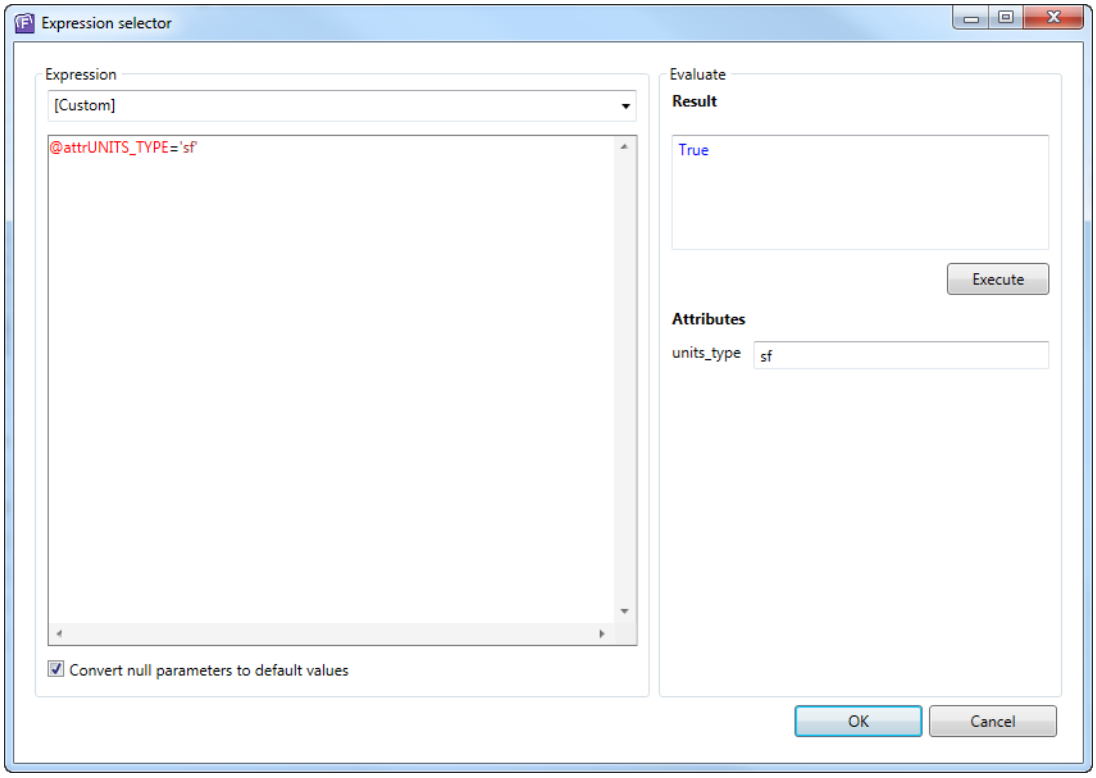
This makes the tab visible only when the unit type is set to square feet. You could then have other labels and attributes inside the tab. These would only need to be configured if the condition were met.

If the view permissions for the tab were removed in the security settings, then these would override the expression. Not being able to view a tab is more restrictive than being able to view it under a specific condition.

To add an expression to a control:

1. Launch the OFD.
2. Open the entity.
3. Select the control.
4. Expand the **MIsc** properties in the Property Explorer.
5. Click the ellipsis button ... beside one of the following properties:
   - **Is enabled**: to enable the control according to an expression.
   - **Is visible**: to make the control visible according to an expression.

   This opens the *Expression Selector*.



6. Enter the expression.
7. Click **OK**.

IsEnabled

The same procedure and logic applies to the IsEnabled parameter for a control. An expression can be used to make the control read-only.

Testing the Expression

You can test the logic of the expression with the Evaluate feature. When you enter an expression, the attributes that it contains are listed on the right under evaluate.

To test the expression:

1. Enter the expression in the Expression Selector.
2. Enter values for the attributes in the Attributes text boxes.
3. Click **Execute**.

The result of the expression is displayed in the **Result** test box. Possible results include the following:

- True or False if the expression evaluates to true or false.
- The result of a calculation; for example, the sum of two attributes.
- The value of an attribute

Adding Text in an Expression

Overview

This section describes several ways of displaying text, or a string, in an expression.

- Single Quotation Marks
- String Syntax
- Substring

It also describes the syntax that can be applied to text.

- Length of a String
- Comparing Strings
- Converting to Uppercase or Lowercase

Single Quotation Marks

The easiest way to display text, or a string, with an expression is to enclose it in single quotation marks.

For example, ‘Govern OpenForms’ returns Govern OpenForms.

String Syntax

You can use the string syntax to add text to an expression.

For example, Str(Govern OpenForms) returns Govern OpenForms.

Substring

Use the substring syntax to return a partial string.

Returns a substring of the first parameter from the x-th character defined by the second parameter on a length defined by the third parameter.

For example, Substr('Govern OpenForms', 7,9) returns OpenForms.

Note that characters in the substring are counted from zero, starting index zero. You can use the Evaluate feature on the Expression Editor to see how the characters are counted.

Length of a String

Use the length syntax to return the number of characters in a string.

For example, Len('Govern OpenForms') returns 16.

Note that characters in the length are counted from one, starting index one. Use the Evaluate feature of the Expression Editor to test this.

Converting to Uppercase or Lowercase

Use the upper and lower syntaxes for converting text, or strings, to upper or lowercase, respectively.

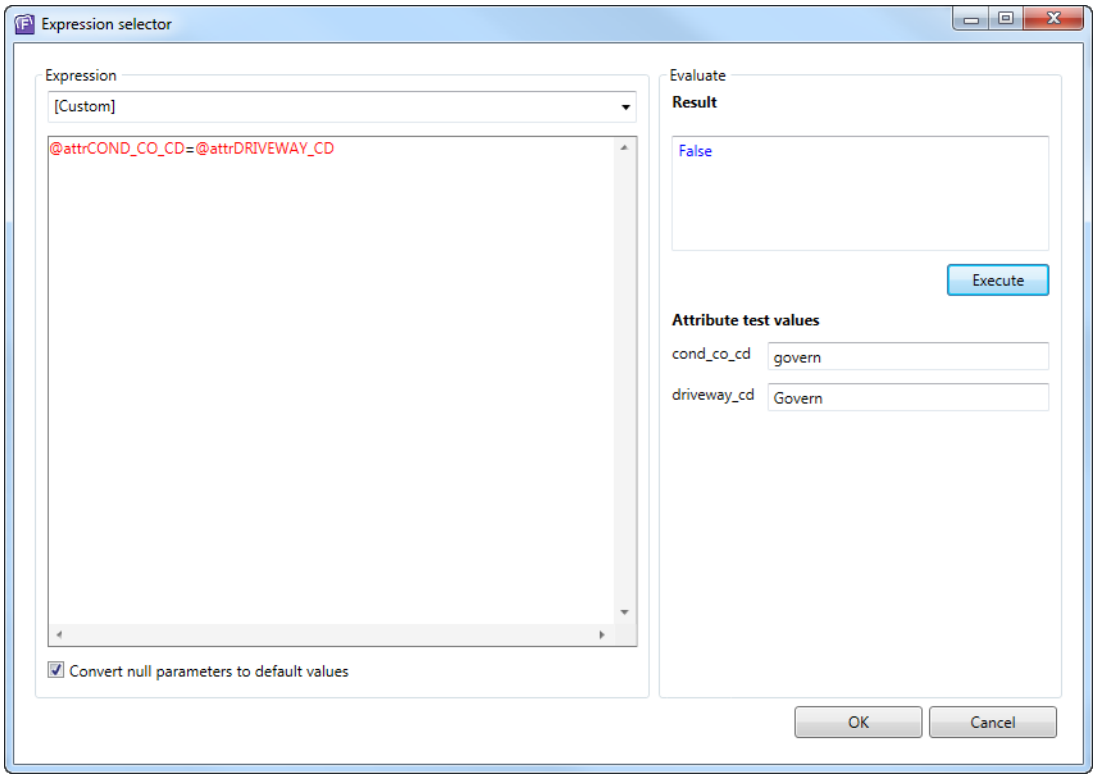
For example, Upper('Govern OpenForms') returns GOVERN OPENFORMS.

For example, Lower('Govern OpenForms') returns govern openforms.

Comparing Strings

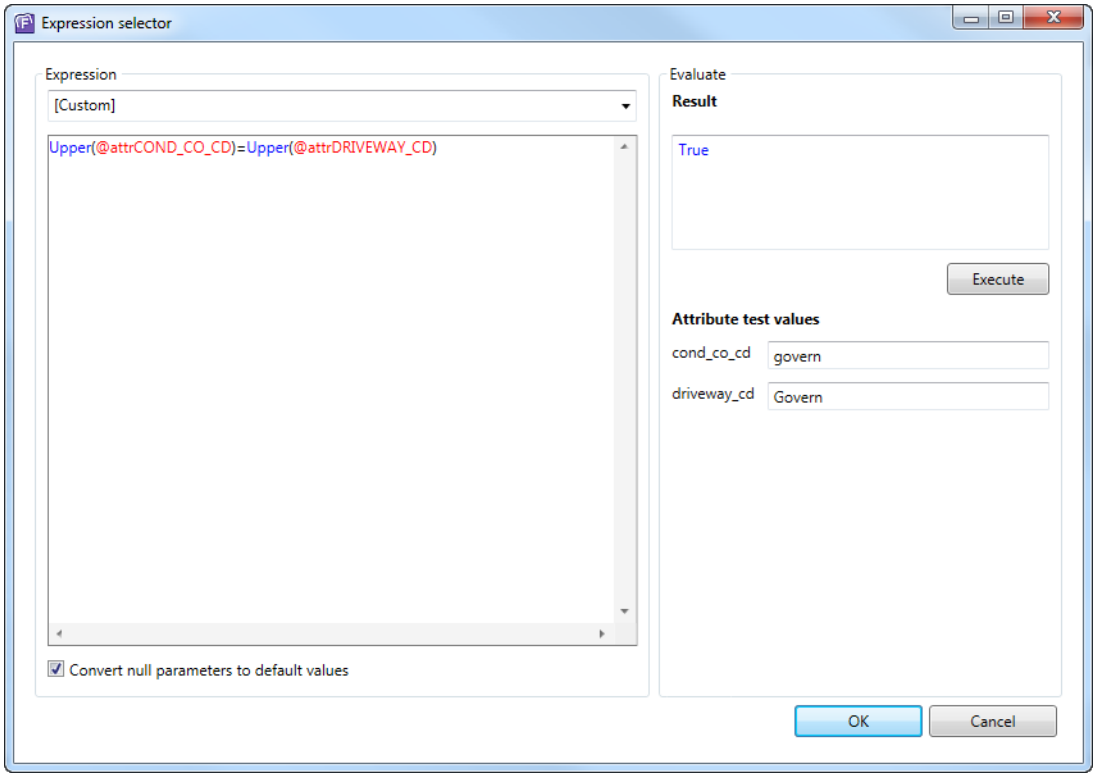
Strings are case-sensitive. This is important to note when you compare strings or use them in an expression. The following scenario illustrates how you can use the upper syntax to ensure that results are not changed due to difference in upper- of lowercase.

The first example compares two attributes that have the text data type.



The result is False because the case is different for the two attributes.

If you add the Upper syntax to the expression, the results are equal.



Applying Formatting to Expressions

Overview

Multiple number and date and time formats are available. There are both predefined and user-defined types. The lists can be viewed on the Microsoft Developer website.

This section describes how to apply formatting to the number, date, and time attributes that you are displaying as a result of your expressions. If you want to display a value or a date and time in a label, or on the record description, you can use the Format function.

For example, if you want to display the Total Land Value with currency formatting, use the following syntax:

```
Format(@attrTOTAL_VALUE, 'c')
```

Format is the function and the second parameter describes how the attribute is formatted.

Available Format Types

Click the following links to view a list of the available formats on the Microsoft Developer Network website:

Numbers

Use predefined numeric formats or create user-defined numeric formats.

Dates and Times

Use predefined date/time formats or create user-defined date/time formats.

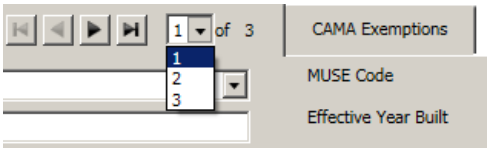
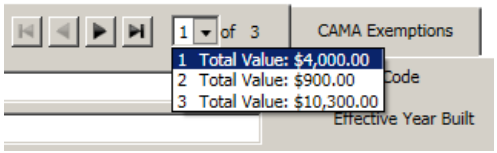
Date and Time Serial Numbers

Use date and time formats or numeric formats.

Examples: Using the Formatting Function

The scenarios in this section provide examples of common expression formats that can be used throughout the user forms.

The following series of screen shots shows the default Record Description and a record description with the Total Land Value. The Record Description is displayed when there are multiple records. It is used to facilitate record selection. Displaying a year or a value is more helpful than the number assigned to the record. *For details about the Record Description, see Defining Properties for the Entity on page 183*.

	
Default Record Description	**Record Description with Expression**

Displaying a Currency Value

This expression shows the currency format type in the Record Description.

Text is added to the label in order to let the users know what value they are viewing. *See Displaying Text with the Formatted Expression on page 273*.

In this example, the Total Miscellaneous Structure Value is displayed in the Record Description in Currency format.

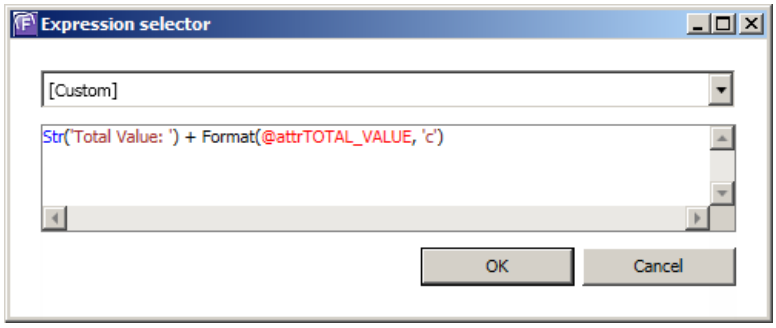
To apply the currency format type:

1. Launch the OFD and open the *CAMA Miscellaneous Structures* form.
2. Select the *Miscellaneous Structures Information* entity.
3. Click the ellipsis button beside the **Record Description** property in the Property Explorer.

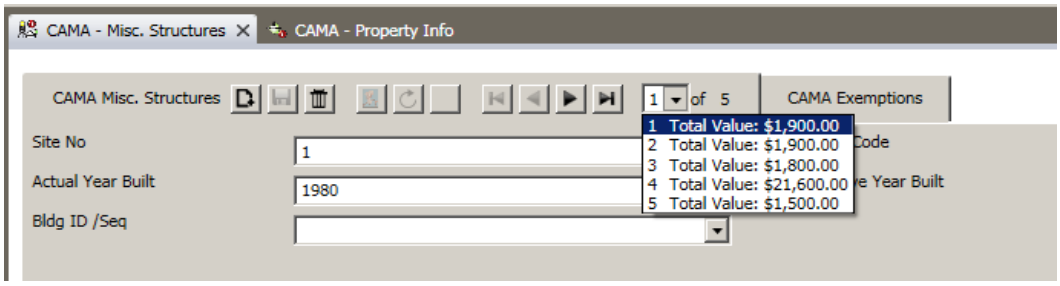
   This opens the **Expression Selector**.

   Enter the following expression.

Note: Str('Total Value: ') + Format(@attrTOTAL_VALUE, 'c')



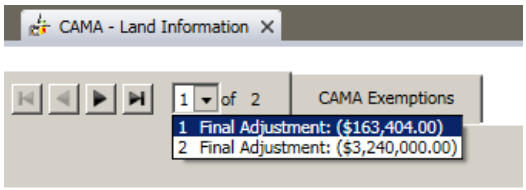
This displays the following in the Record Description.



Displaying a Negative Currency Value

The format for negative currency values is the same as for positive. Negative values are displayed in parentheses.

This is shown in the following screen shot:



Displaying a Percentage

To display a percentage in the Expression result, the following user-defined format is recommended:

```
FORMAT(@attribute, '0 \\%')
```

For example:

```
FORMAT(@attrFINAL_ADJ_PERCENT, '0 \\%')
```

This format displays the percentage correctly.

The predefined format, called **Percent** (see predefined numeric formats), multiplies the value by 100. Therefore, if you were to enter 10% in the Final Adjustment field, it would be multiplied by 100 and 1,000 would be displayed. This would be inaccurate.

Applying Formatting to a Date or Time

To display a date or a date and time in the Record Description, you can select any of the Predefined date and time formats (see predefined date/time formats). The long, short, or medium date and time are displayed according to the Regional Settings selected on your computer.

The following screen shot shows the **Effective Date** in the **Record Description** of the *Building Information* form, using the long date format:

To apply this formatting:

1. Launch the OFD and open the *CAMA Buildings* form.
2. Select the *Building Information* entity.
3. Click the ellipsis button  beside the **Record Description** property in the Property Explorer.

   This opens the **Expression Selector**.

   Enter the following expression.

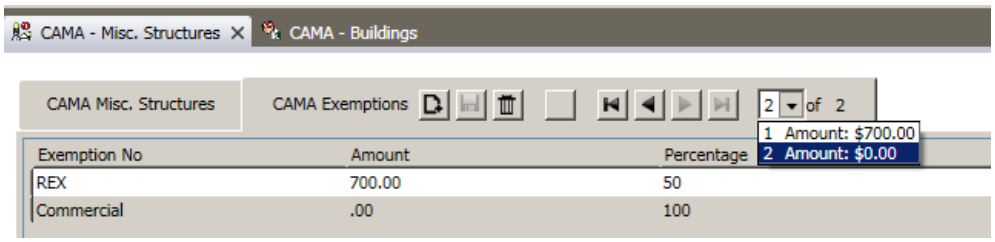
   ```
   'Effective Date: ' + Format(@attrEFFECTIVE_DATE, 'D')
   ```

4. Click **OK** on the Expression Selector window.

5. Click **Save** on the OpenForms Designer.

Applying Formatting in the One-to-Many Format

The same Format function can be used in entities that are set to One-to-Many. The following example shows the Format function applied to the Amount in the Building Exemptions entity.



Displaying Text with the Formatted Expression

The examples in this section show the formatted expressions with a text label beside them. The label describes the value for the users.

To add a text label, you can do one of the following:

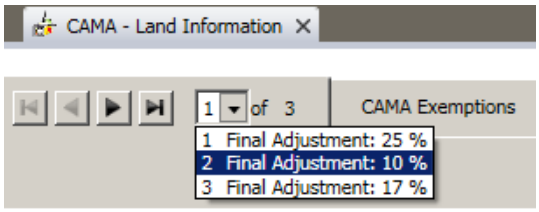
- Add the text to the expression with a plus sign (+), as follows:

```
'Final Adjustment: ' + Format(@attrFINAL_ADJ_PERCENT, '0 \\%')
```

- Include the text in the Format function, as follows:

```
FORMAT(@attrFINAL_ADJ_PERCENT, 'Final Adjustment: 0 \\%')
```

The result is the same.



Displaying the Field Description

Overview

The **Display** expression syntax can be used to show the short or long description that appears on a form. For example, if you want to display the value of a combo box ion a label or in the Record Selector, you could add the Display syntax to the normal attribute. This is created by Govern.

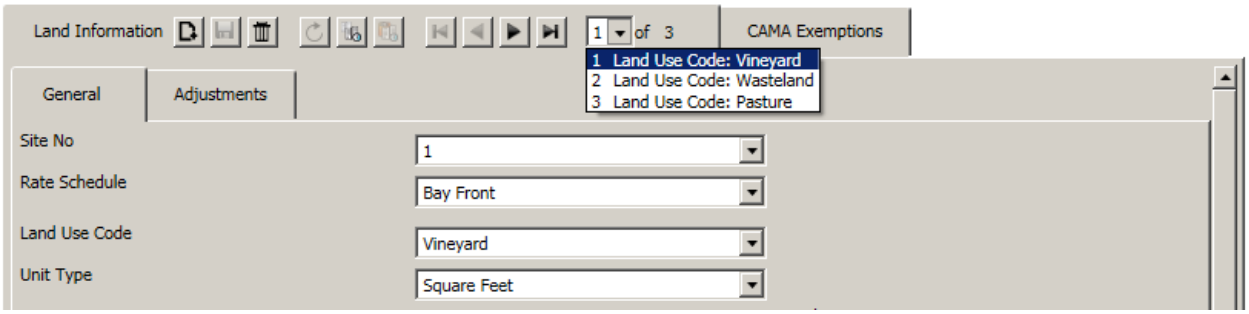
The syntax is as flows:

```
.Display
```

For example, the following expression shows the value selected for the Land Use Code:

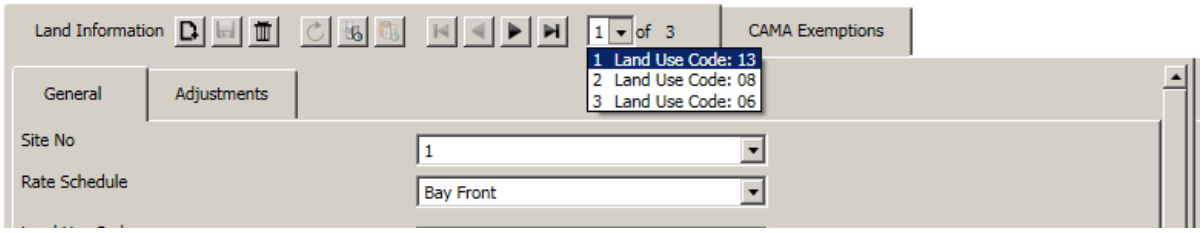
```
'Land Use Code: ' + @attrLUSE_CODE.Display
```

The text inside the single quotation marks is used for identification.



Without the Display syntax, this expression displays the numeric code associated with the Land Use Code. This is how it is saved in the database.

```
'Land Use Code: ' + @attrLUSE_CODE
```



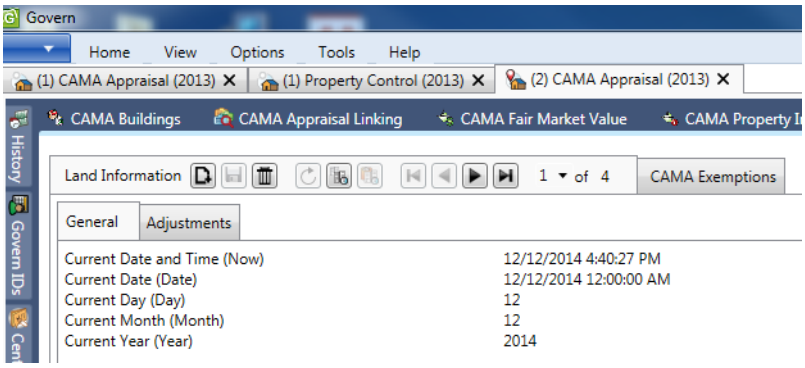
Displaying a Calculated Date Field

Overview

Several options are available for date fields. They can be categorized as follows:

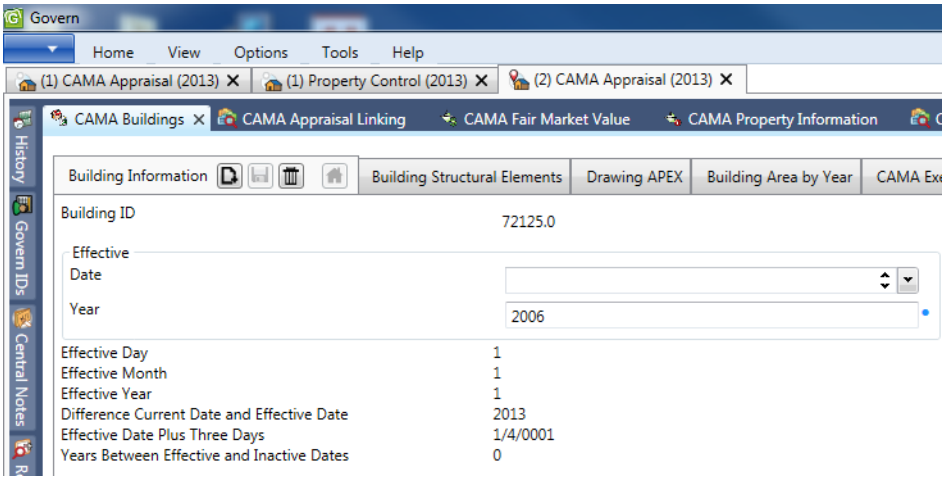
- Displaying a date or a component of a date, such as the day, month, year, or time.

  The following screen shot shows the results of expressions that display dates:



- Adding or subtracting a value to or from a date or a part of a date, such as adding two days to the current date, or showing the difference between two dates.

  The next screen shot shows the results of expressions that calculate dates:



The following table lists the accepted syntaxes for date expression and provides examples. Attributes, system dates, or Govern IDs, such as the Year ID can be used in these expressions.

Expression Syntax	Description	Example
Now()	Displays the current date and time.	12/1/2015 3:24:36 PM
Date()	Displays the current date if used alone.	12/1/2015 3:24:36 PM
Date(@attr)	Displays the full date of the associated attribute if one is included in the expression.	Date(@EffectiveDate) 1/31/2015 5:29:36 PM
Format(Date())	At this time, the Format syntax is required for this expression in order to hide the time.	Format(Date(@EffectiveDate), 'd') 12/1/2015
Day()	Displays the current day.	1
Day(@attr)	Displays the day of the associated attribute if one is included in the expression.	Day(@EffectiveDate) 1
Month()	Displays the current month.	12
Month(@attr)	Displays the month of the associated attribute if one is included in the expression.	Month(@Effective Date) 12
Year()	Displays the current year.	2015
Year(@attr)	Displays the year of the associated attribute if one is included in the expression.	Year(@EffectiveDate) 2015

Adding a Positive or Negative Value to a Date

You can use the following syntax to add or subtract a value to a date. For example, you could add two days to the current date. The syntax for this is

```
DateAdd()
```

Add the part of the date that you want to update, the value, and the date that you want to update in the parentheses.

> For example: DateAdd('day', 2) adds two to the current day.
> If this is December 1, 2014, the result is December 3, 2014.

The following parts of date can be used:

- Day
- DayOfYear
- Hour
- Minute
- Month
- Quarter
- Second
- Weekday
- WeekOfYear
- Year

If you are adding the DateAdd() syntax to a text field, add the string or a format syntax.

> For example: Str(DateAdd('day', 2))
>
> or Format(DateAdd('day', 2), 'd') adds two to the current day.
> If this is December 1, 2014, the result December 3, 2014 is displayed.

You could also add three hours, four weeks, or five months. In other words, you could add or subtract any value to any of the following parts of dates:

Displaying an Interval Between Dates

You can use the following syntax to display an interval between two dates. For example, you may want to show the difference between the effective date of a record and the current date. The syntax for this is

> DateDiff('DatePart', @attr1, @attr2)
>
> where DatePart is the part of the date or time that you want to update, such as, the hour, minute, day, week, or year.

Add the part of the date that you want to update, the value, and the date that you want to update in the parentheses.

> For example: DateDiff('day', @EffectiveDate, @InactiveDate) displays the number of days difference between the effective date and the inactive date of the record.
> If this is December 1, 2014, the result is December 3, 2014.

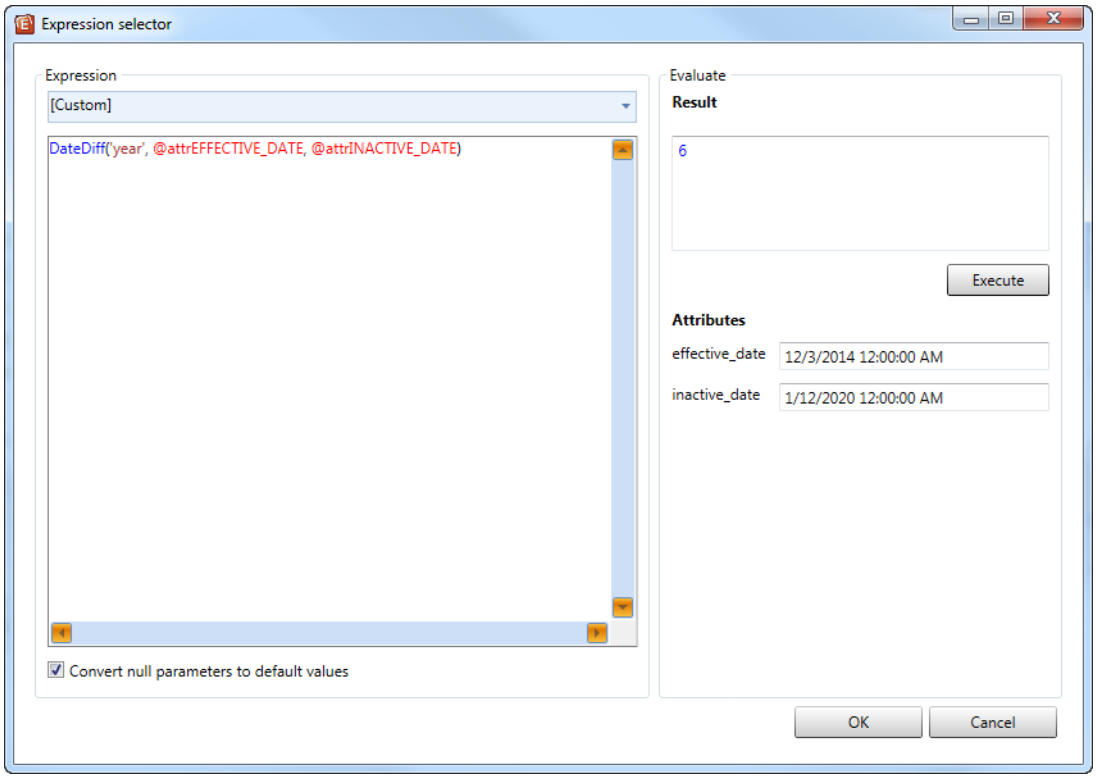
The following components of a date can be used in an expression:

- Day
- DayOfYear
- Hour
- Minute
- Month
- Quarter
- Second
- Weekday
- WeekOfYear
- Year

Adding the Expression

To add an expression to a default value:

1. Open the business entity.
2. Select the attribute.
3. Click the ellipsis button  in the **Default Value** field to open the Expression Selector.



4. Enter the expression in the Expression Selector.

   The attributes that are used in the expression are displayed on the right.
5. Enter typical dates in the effective and inactive date fields.
6. Click **Execute** to test the expression.
7. Click **OK**.
8. Click **Save** in the Business Entity Designer.
9. Open the form in Govern to view the expression.

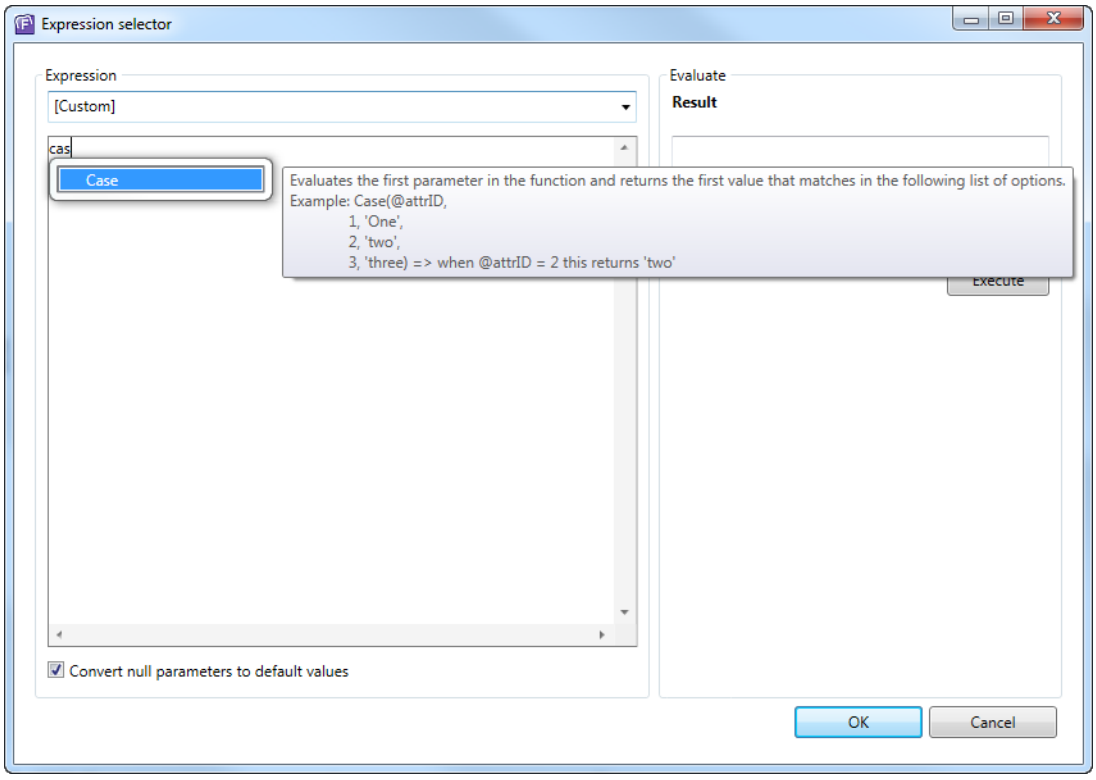
For Further Reference

Overview

A listing and description of the available formats are available from the following links:

Govern Expressions

All Govern expressions as well as the standard expressions are added in tooltips in the Expression Selector. When you enter the first couple of characters, the correct syntax is displayed.



Note: Hit `[Ctrl] + [space]` to view a list of all accepted syntaxes.

Mathematical Expressions

The following links provide lists and descriptions of the supported expression types:

http://ncalc.codeplex.com/wikipage?title=functions&referringTitle=Home

http://ncalc.codeplex.com/wikipage?title=operators&referringTitle=Home

Format Function

The available formats can be viewed from the following lists:

Numbers

Use predefined numeric formats or create user-defined numeric formats.

Dates and Times

Use predefined date/time formats or create user-defined date/time formats.

Using a Variable

Overview

You can add a variable to an expression.

The first step is to declare the variable.

To add a Variable, use curly brackets ({ })to declare it.

For example,

```
{
  variable = Month()
}

variable * 10
```

Returns 100, if the month is October.

It is not necessary to write the word variable. For example, the following are all variables:

```
{
  variable = Day();
  Subtotal = 1*5;
  Total = Subtotal+2;
  }
```

….