Working with Common Attribute Framework

Understanding Common Attribute Framework

Common Attribute Framework enables you to define a common attribute and add the common attribute as a data field to a Campus Solutions page . A user can then enter a value on the page for the data field.

A common attribute is a data element and is associated with a record in the system. A common attribute has the following properties:

- Name
- Type
- Default value (for example, a common attribute field on the user interface can show a default value that a user can change)
- Format (for example, suppose a user enters a value in a common attribute zip code field. In such a case, you may want the system to automatically append a hyphen between the fifth and sixth character when saving the value to the database)
- Validation

The framework delivers the following attribute types:

- Date
- Time
- Yes/No
- Number
- Long text
- Text (maximum number of characters is 50)
- Short text (maximum number of characters is 20)
- List of values

To implement a common attribute, you must perform the following steps.

1. Use the Application Designer to:
    a. Identify the record that needs to be extended with attributes. For example, to add attributes to the General Materials page, choose the GENL_MATERIALS record.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

b. Create a child record with the same key fields as the identified record. Add the CAF subrecord SCC_CAF_SBR to this child record. The framework uses this child record to store attribute data.

c. On pages where the identified record is used, add the delivered subpage (SCC_CAF_LAUNCH_SBP or SCC_CAF_SBP depending on the desired user interface experience).

2. Use the Common Attribute page to:

   a. Create attributes that you intend to associate with the identified record. For example, if you want to add Passport Number and Primary Institution fields to the General Materials page, create two attributes: Passport Number and Primary Institution.

   b. Define the attribute type and format.

3. Use the Record Context page to associate the attributes to the record. Continuing with the above example, use this page to add the two attributes to the GENL_MATERIALS record so that the two fields appear on the General Materials page.

In the following sections of this documentation, we use this example of adding attributes to the General Materials page to illustrate the implementation.

Using Application Designer to Configure a Record

Steps:

1. Identify the record to which attributes should be associated. For example, the following shows a record in Application Designer:

   Image: GENL_MATERIALS record

   This example illustrates the fields and controls on the GENL_MATERIALS record. You can find definitions for the fields and controls later on this page.

Num	Field Name	Type	Key	Ordr	Dir	CurC
1	EMPLID	Char	Key	1	Asc	
2	GENL_MATL_GROUP	Char	Key	2	Asc	
3	GENL_MATL_TYPE	Char	Key	3	Asc	
4	GENL_MATL_NBR	Nbr	Key	4	Desc	
5	CHKLST_ITM_UPDT	Char				
6	DT_RECVD	Date				

2. Create a child record for the identified record.

3. Make sure the child record has all the keys of the parent record and the subrecord SCC_CAF_SBR. The SCC_CAF_SBR subrecord adds a new key SCC_CAF_ATTR_SEQ to this new record, thereby

 Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

making this a true child record to GENL_MATERIALS as this new record now has one additional key. The following shows the child record that you can create for the GENL_MATERIALS record:

Image: New child record: GENL_MATERLS_CA

This example illustrates the fields and controls on the New child record: GENL_MATERLS_CA. You can find definitions for the fields and controls later on this page.

Num	Field Name	Type	Key	Ordr	Dir	CurC
1	EMPLID	Char	Key	1	Asc	
2	GENL_MATL_GROUP	Char	Key	2	Asc	
3	GENL_MATL_TYPE	Char	Key	3	Asc	
4	GENL_MATL_NBR	Nbr	Key	4	Desc	
5	SCC_CAF_SBR	SRec				
	SCC_CAF_ATTR_SEQ	Nbr	Key	5	Asc	
	SCC_CAF_ATTRIB_NM	Char				
	SCC_CAF_ATTR_VAL	Char				
	SCC_CAF_ATTR_NVAL	Nbr				
	SCC_CAF_ATTR_DVAL	Date				
	SCC_CAF_ATTR_TVAL	Char				
	SCC_CAF_ATTR_LVAL	Long				

The framework uses this child record to store the attribute data.

Note:

Bundle 42: Updated information for multi-language functionality.

The SCC_CAF_LN_SBR subrecord provides multi-language functionality. This subrecord has an additional key field LANGUAGE_CD and all the fields from the SC_CAF_SBR subrecord that can contain descriptions such as SCC_CAF_ATTR_VAL and SCC_CAF_ATTR_TVAL.
In cases where the CAF field is added to a non-transaction table, Oracle recommends that you also create a related language record as appropriate. For an example of CAF functionality on a non-transactional table, refer to the CAF subrecord PS_EXT_ORG_TBL_CA created for table PS_EXT_PROG_TBL.

---

4. Modify the definition of the runtime page that displays the attributes. To do so, choose from the two delivered subpages:

   - SCC_CAF_LAUNCH_SBP: Displays attributes in a secondary page.
   - SCC_CAF_SBP: Displays attributes in a scroll area.

   These two generic subpages, which the framework delivers, enable a user to access and manipulate the attributes in two different ways. The SCC_CAF_LAUNCH_SBP subpage displays a link. When the user clicks the link, the system launches a secondary page with all attributes associated with a record. The other subpage SCC_CAF_SBP displays attributes in a scroll area. You can choose one of these subpages for a quick implementation. There are no further technical changes or coding required. If you require a different type of user interface, then you must use the Common Attribute API.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

See Common Attribute Framework Application Class Reference.

Generic subpage SCC_CAF_LAUNCH_SBP

The following is the SCC_CAF_LAUNCH_SBP subpage:

Image: SCC_CAF_LAUNCH_SBP subpage in Application Designer

This example illustrates the fields and controls on the SCC_CAF_LAUNCH_SBP subpage in Application Designer. You can find definitions for the fields and controls later on this page.

This subpage displays only a hyperlink on the page. Click this link to launch a modal secondary page containing all the attributes associated with the record. You can customize the hyperlink label on the Record Context page.

Considerations for using this subpage:

- If you want a small user-interface footprint on an existing page.
- This subpage does not contain a scroll area. This means that users are not able to add multiple instances of an attribute, therefore, you should employ this option only in areas where a single instance of an attribute is required. For example, if the subpage is intended for users to record only a single value for an attribute such as Which Social Media Tool Do You Use, use this subpage.

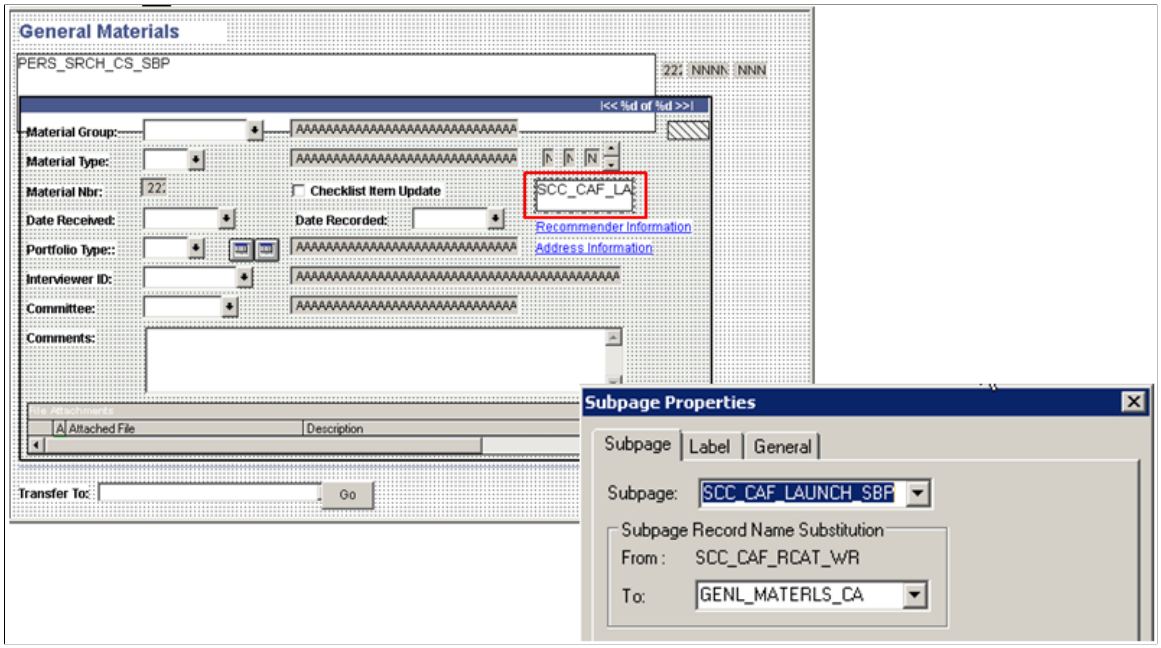
You must place this subpage in the scroll area where the extended record is the primary record. Also, ensure that the subpage is in the same component buffer level as the record in the page order tab.

 Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

This example shows the addition of SCC_CAF_LAUNCH_SBP to GENL_MATERIALS in Application Designer:

Image: Adding SCC_CAF_LAUNCH_SBP subpage to GENL_MATERIALS

This example illustrates the fields and controls on the Adding SCC_CAF_LAUNCH_SBP subpage to GENL_MATERIALS. You can find definitions for the fields and controls later on this page.



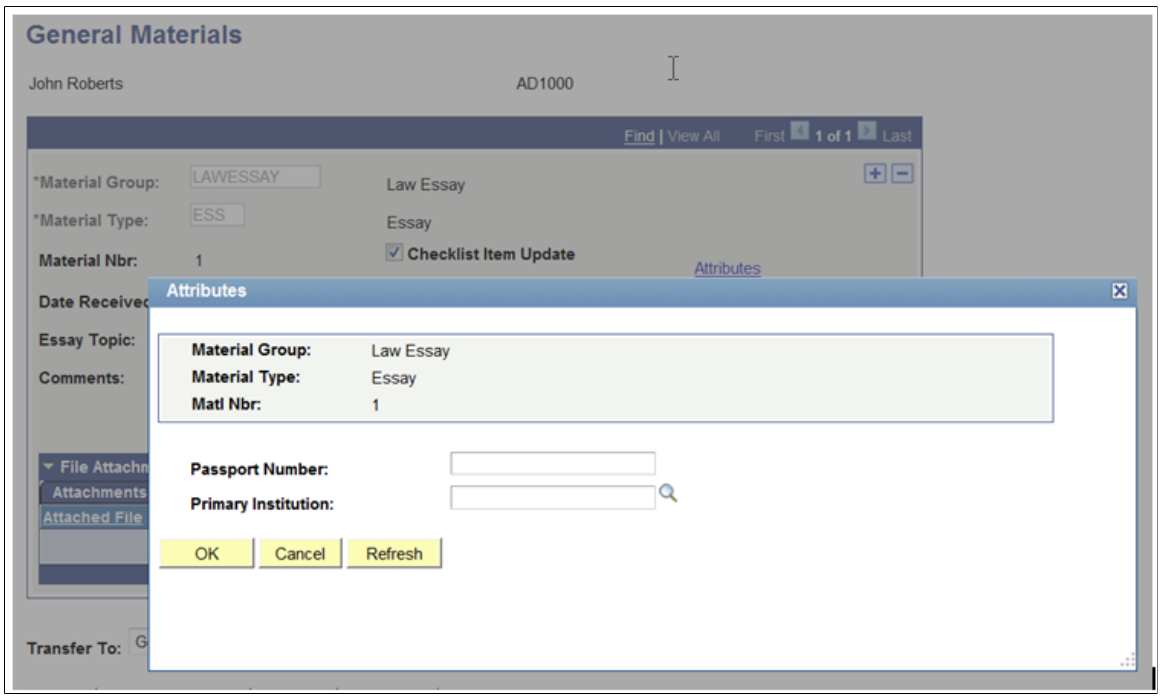
As shown in the above example, the attribute record (the new child record) must be entered in the To field of the Subpage Record Name Substitution section in the Subpage Properties window.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The following example shows the runtime result of implementing SCC_CAF_LAUNCH_SBP. In the example, the user can click the Attributes link on the General Materials page to enter the passport number and primary institution.

Image: Attributes on a secondary page

This example illustrates the fields and controls on the Attributes on a secondary page. You can find definitions for the fields and controls later on this page.



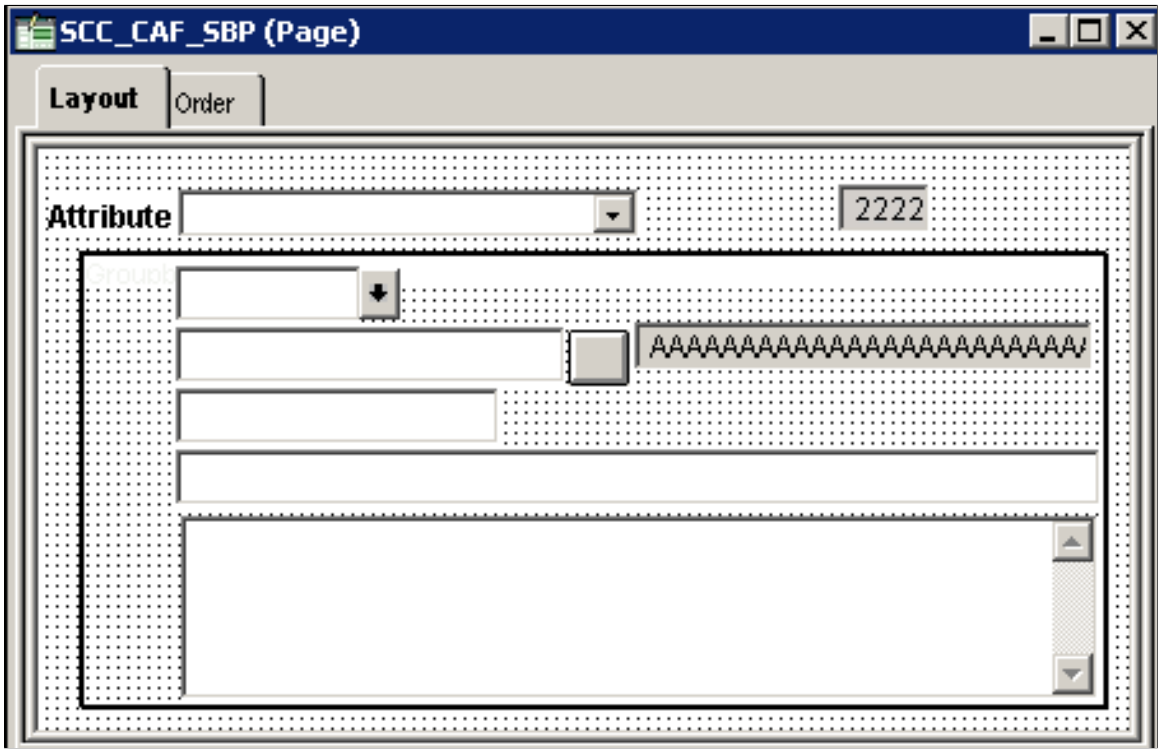
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Generic Subpage SCC_CAF_SBP

The following is the SCC_CAF_SBP subpage:

Image: SCC_CAF_SBP subpage in Application Designer

This example illustrates the fields and controls on the SCC_CAF_SBP subpage in Application Designer. You can find definitions for the fields and controls later on this page.



This subpage displays the associated attributes directly on the page. You must place this subpage inside the scroll area of the page. Also, the attribute record (the new child record) must be entered in the To field of the Subpage Record Name Substitution section in the Subpage Properties window. The component level of the scroll area should be one level higher than the record being extended.

Considerations for using this subpage:

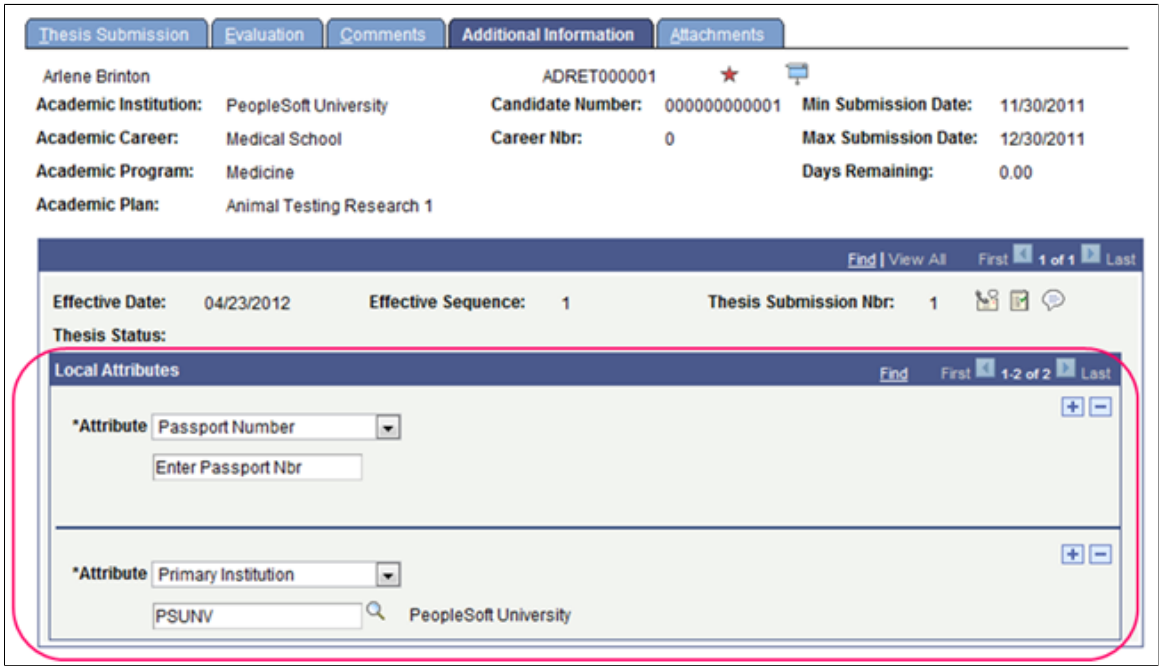
- The user-interface footprint on an existing page is not an issue.
- There is a functional requirement for a scroll area, where users need to enter multiple instances of a repeatable attribute. For example, if you want to enable the user to record multiple values for a Which Social Media Tool Do You Use attribute, use this subpage.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The following example shows the runtime result of implementing SCC_CAF_SBP:

Image: Attributes on the primary page

This example illustrates the fields and controls on the Attributes on the primary page. You can find definitions for the fields and controls later on this page.



Viewing the Common Attribute Type Details

This section discusses how to view the details of the delivered common attribute types.

Page Used to View the Common Attribute Type Details

Page Name	*Definition Name*	*Navigation*	*Usage*
Attribute Type	SCC_CAF_ATTR_TYPE	Set Up SACR, System Administration, Utilities, Common Attributes Setup, Attribute Type	View the details of the delivered common attribute types.

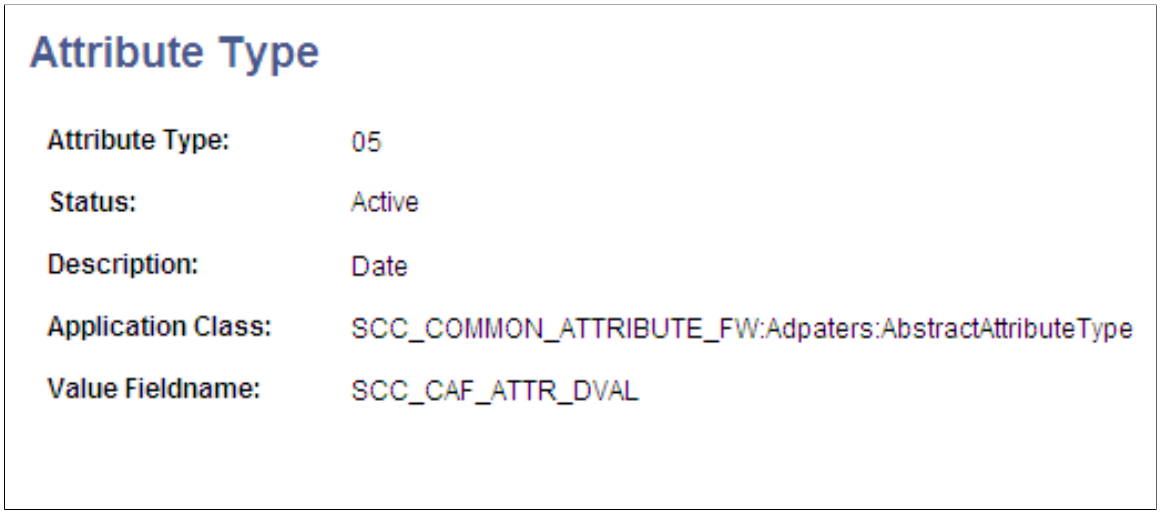
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Viewing the Common Attribute Type Details

Access the Attribute Type page (Set Up SACR, System Administration, Utilities, Common Attributes Setup, Attribute Type).

Image: Attribute Type page

This example illustrates the fields and controls on the Attribute Type page. You can find definitions for the fields and controls later on this page.



Defining a Common Attribute

This section discusses how to define a common attribute and how to:

- Create a common attribute.
- Define a list of values for a common attribute.

Pages Used to Define a Common Attribute

Page Name	*Definition Name*	*Navigation*	*Usage*
Common Attribute	SCC_CAF_ATTRIBUTES	Set Up SACR, Common Definitions, Common Attributes Setup, Common Attribute	Create, delete, or modify a common attribute.
CAF Values - LOV Definition	SCC_LOV_DEFN	Click the Create New CAF Values link or the Open CAF Value Setup link on the Common Attribute page.	Create or modify a list of values for an attribute of a list of values type. You can create a list of values yourself or you can have the framework retrieve the list of values from either a table or a translate field at runtime.

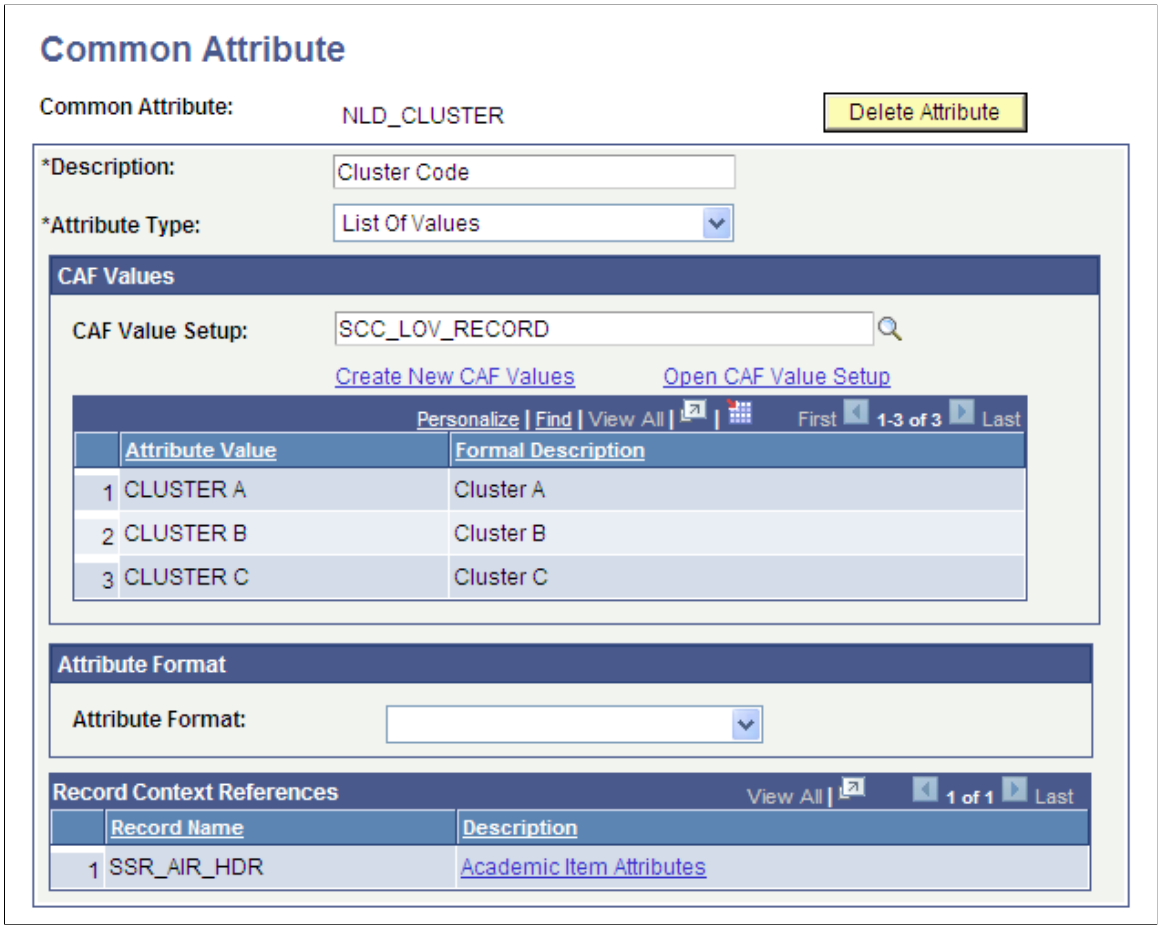
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Creating a Common Attribute

Access the Common Attribute page (Set Up SACR, Common Definitions, Common Attributes Setup, Common Attribute).

Image: Common Attribute page

This example illustrates the fields and controls on the Common Attribute page. You can find definitions for the fields and controls later on this page.

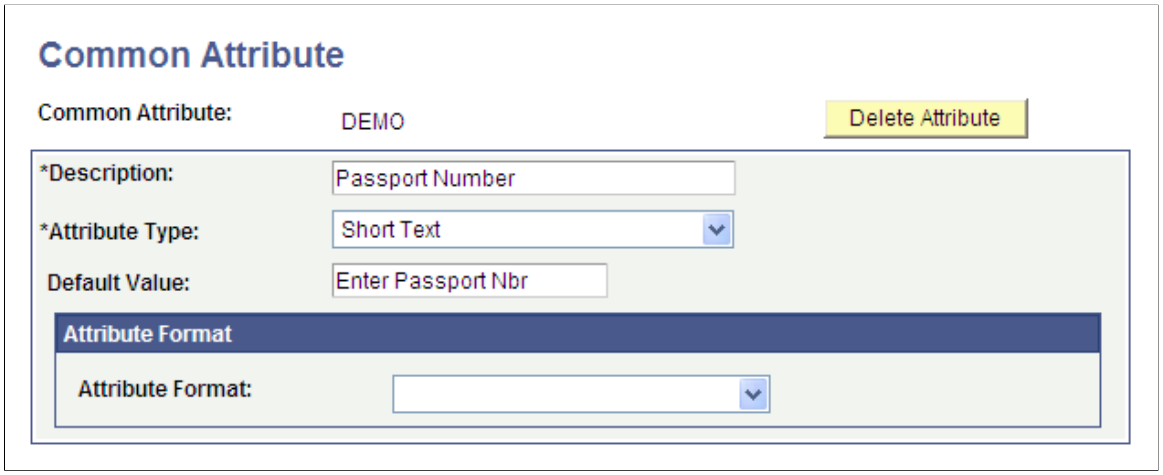


 Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Continuing with the GENL_MATERIALS record example that was discussed in the Application Designer section, we use the Common Attribute page to create the common attribute - Passport Number - as shown in this example screenshot:

Image: Example of a Common Attribute

This example illustrates the fields and controls on the Example of a Common Attribute . You can find definitions for the fields and controls later on this page.



The value that you select in the Attribute Type field determines which fields are available for entry on this page. For example, the CAF Values group box is available only when you select an attribute type of *List of Values.*

Common Attribute

Displays the name of the attribute.

You specify an attribute name when you access the page in Add mode.

Delete Attribute

Available only after you save an attribute.

Description

The description that you enter here appears as a label for the attribute in the delivered subpages.

Attribute Type

The Attribute Type determines the type of value that a user can enter for the attribute.

Select:

- *Date:* The attribute appears as a date control on the delivered subpages.
- *Time:* The attribute appears as a time control on the delivered subpages.
- *Yes/No:* The attribute appears as a yes/no field on the delivered subpages.
- *Number:* The attribute appears as a number data control on the delivered subpages.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

- *Long text:* The attribute appears as a long edit box on the delivered subpages.
- *Text:* The attribute appears as a regular edit box on the delivered subpages. The maximum number of characters that a user can enter for this type is 50.
- *Short text:* The attribute appears as a regular edit box on the delivered subpages. The maximum number of characters that a user can enter for this type is 20.
- *List of values:* The attribute appears as an edit box with a lookup icon on the delivered subpages. Users can click the lookup icon and select a value for the attribute. Users can also enter a value in the box and when they tab out, the framework validates if the entered value exists in the list of values. If the entered value does not exist in the list, then the system displays an error message.

Integer Position

Available only when the Attribute Type is *Number*.

Enter the maximum number of digits that are allowed before the decimal for numeric attribute types. For example, if you enter *3* here and the user enters a value *9934.89* for the attribute, the system returns an error. However, if you enter *3* here and if the user enters a value *993.89,* it means the user is entering the number in the correct format.

Decimal Position

Available only when the Attribute Type is *Number*.

Enter the maximum number of decimal places that are allowed for numeric attribute types. For example, if you enter *1* here and the user enters a value *993.89* for the attribute, the system returns an error. However, if you enter *1* here and if the user enters a value *993.8,* it means the user is entering the number in the correct format.

Enable Rich Text Editing

Available only when the Attribute Type is *Long Text.*

Select this check box if you want enable editing and formatting capabilities for users when they edit the attribute value in the delivered subpages.

Default Value

Available only when the Attribute Type is *Short Text, Long Text, Number, Text,* or *Yes/No*. When the Attribute Type is *Yes/No,* you can set the default value by selecting the Yes or No radio button.

The value that you enter here appears as the default value for the attribute in the delivered subpages.

Default to current date

Available only when the Attribute Type is *Date.*

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Select this check box if you want the system date to appear as the default value for the attribute in the delivered subpages.

Default to current time

Available only when the Attribute Type is *Time.*

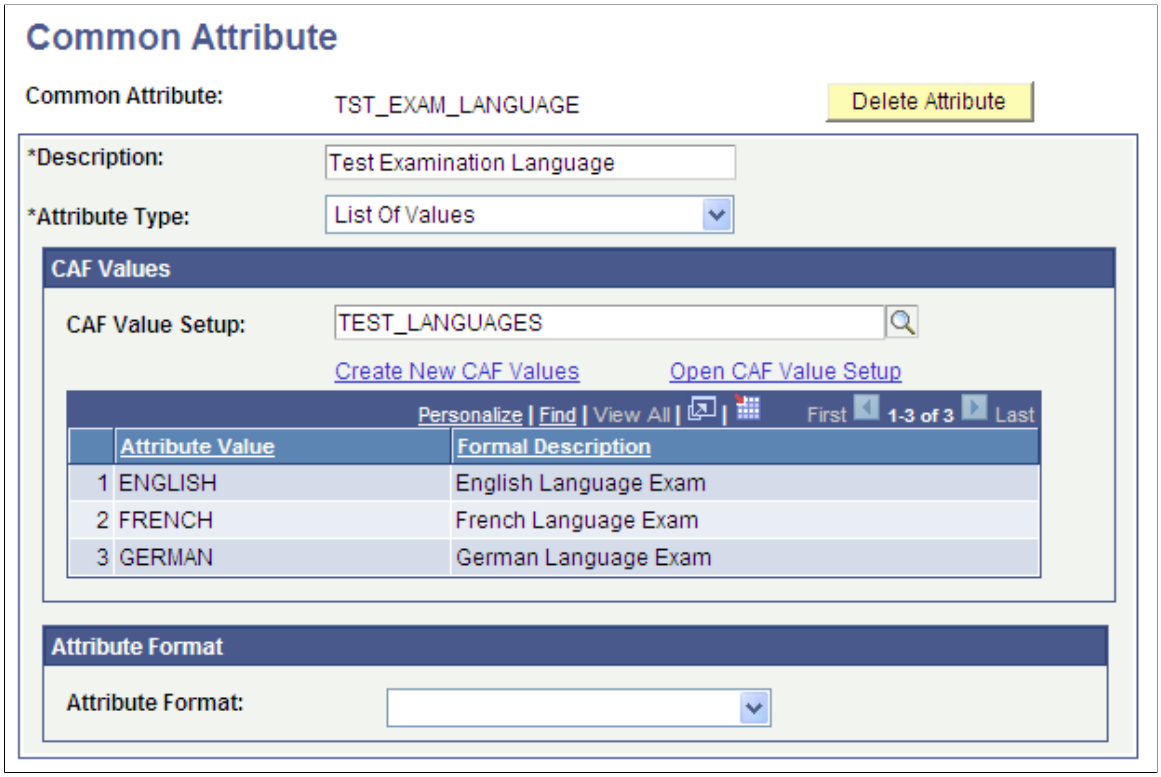
Select this check box if you want the system time to appear as the default value for the attribute in the delivered subpages.

CAF Values

This group box is available only when the Attribute Type is *List of Values.*

Image: Example of a Common Attribute page for List of Values

This example illustrates the fields and controls on the Example of a Common Attribute page for List of Values. You can find definitions for the fields and controls later on this page.



Use the CAF Values group box to define the values in the list.

CAF Value Setup

Indicates the list of values definition that you have selected for the attribute.

If you enter this page in Add mode, the CAF Value Setup field is blank. You can either use this field to select a list of values that already exists in the system or click the Create New CAF Values link to create a new list of values definition.

Create New CAF Values

Click to access the CAF Values - LOV Definition page where you can define a new list of values for the attribute.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Open CAF Value Setup

This link appears only if the CAF Value Setup field is populated.

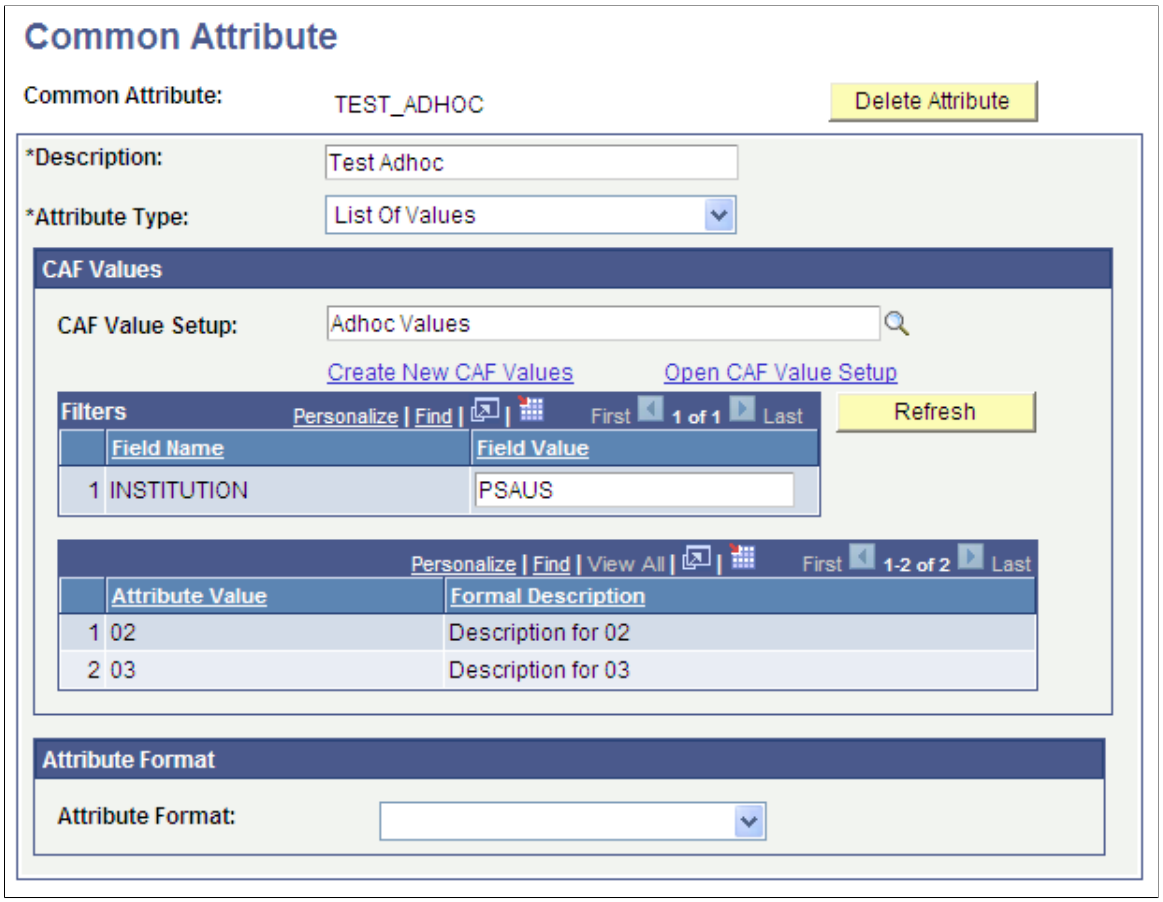
Click this link to access the CAF Values - LOV Definition page where you can view or edit the list of values that is selected for the attribute.

Filters

When a common attribute with a List of Values type based on Ad Hoc Values or Table is saved and the associated CAF value setup in the CAF Values - LOV Definition page uses a filter field, the Filters grid is available on the Common Attribute component:

Image: Example of a Common Attribute page with the Filters grid

This example illustrates the fields and controls on the Example of a Common Attribute page with the Filters grid. You can find definitions for the fields and controls later on this page.



This feature enables you to preview the values that will be available to a user based on the filter field values defined in the CAF Values - LOV Definition page. To preview the values that are available, enter a field value in the Filters grid and click the Refresh button.

See Defining a Common Attribute, “Defining a List of Values for a Common Attribute.”

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Attribute Format

Use this group box to specify the formatting and validation that the framework must perform on the attribute value. Select from the delivered formatting and validation options, or specify your own custom application class.

Attribute Format

Specify the formatting and validation that the framework must perform on the attribute value:

- *Email Address:* Validates if the value entered for the attribute is a valid email address or not. The validation logic used in Campus Community Bio/Demo pages is used here.
- *ISO Country Code:* Verifies if the data entered for the attribute is a valid ISO country code.
- *US State Code:* Verifies if the data entered for the attribute is a valid US state code.
- *US Zip Code:* This is a formatting class. If the attribute value is of 9 digits, it introduces a hyphen between the 5th and 6th character on tab out.
- *Custom Class:* Enables you to specify an application class that performs the formatting and validation.

Application Class

Available only when the Attribute Format is *Custom Class.*

Specify the application class that formats and validates the attribute value.

To write a custom class using PeopleCode, see Writing Custom Logic for Validation and Formatting.

Record Context References

This section lists all the record contexts with which the attribute is associated. Click the link on the Description column to access the Record Context component for the particular record context.

When you define a common attribute, the Record Context References section does not appear on the page. The section appears only after you associate the common attribute to one or more records using the Record Context component.

Defining a List of Values for a Common Attribute

Access the CAF Values - LOV Definition page (click the Create New CAF Values link or the Open CAF Value Setup link on the Common Attribute page).

LOV Unique ID

Displays the system-assigned ID for the list of values.

When you create a new list of values, this field displays *NOID.* When you save the list, the system assigns an ID.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

CAF Value Type

Specify whether you want to define the values yourself (Ad Hoc Values), or whether you want the framework to retrieve the values from a table (Table) or a translate field (Translate Values).

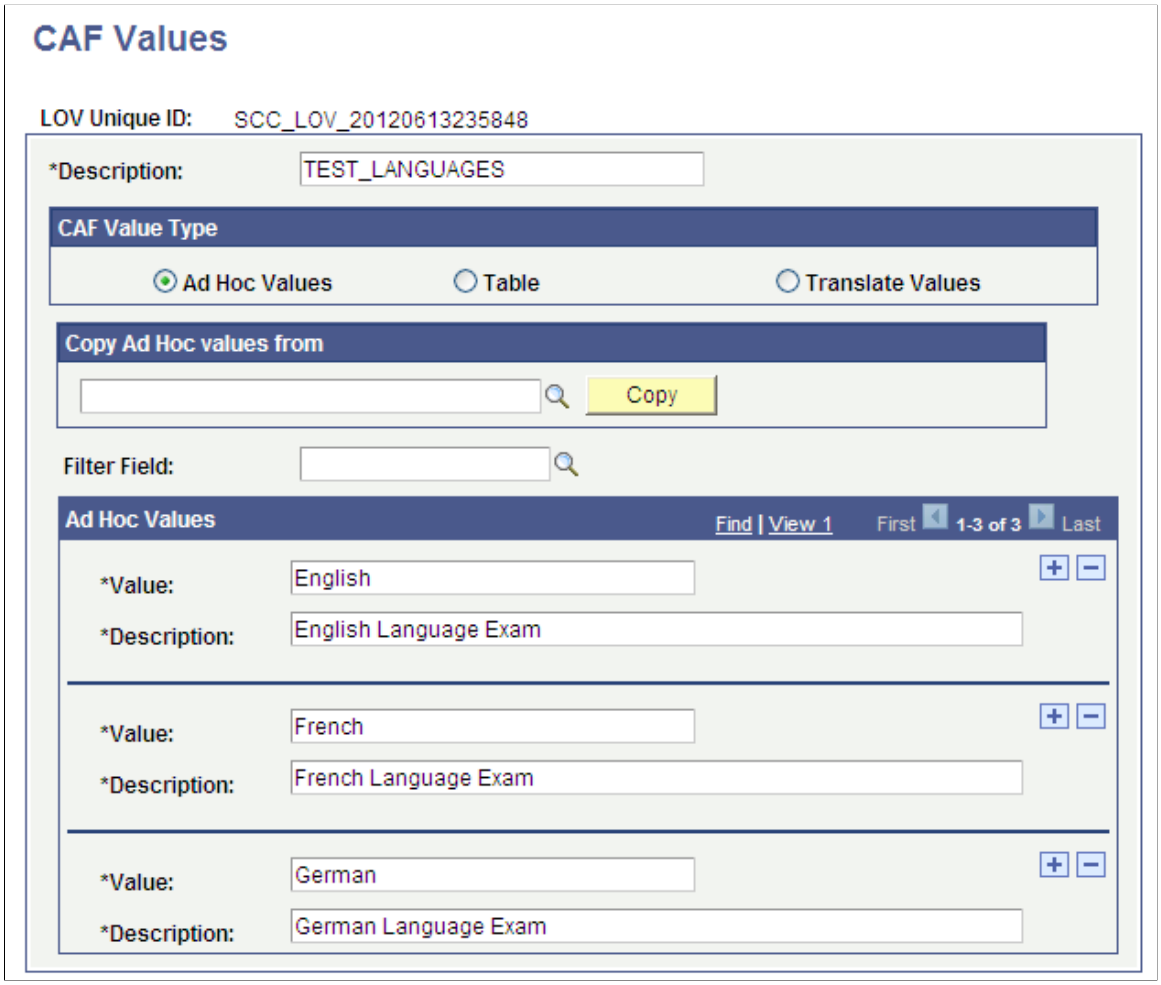
The fields on the page change depending on the CAF value type that you select.

Defining Ad Hoc List of Values

If the CAF Value Type is *Ad Hoc Values,* then you can enter a free-form list of value and description pairs.

Image: CAF Values - LOV Definition page (Ad Hoc list of values)

This example illustrates the fields and controls on the CAF Values - LOV Definition page (Ad Hoc list of values). You can find definitions for the fields and controls later on this page.



Enter the value-description pairs in the Ad Hoc Values group box.

Use the Copy Ad Hoc Values From group box to copy values from an existing list. To copy values from an existing list: Click the lookup icon in the Copy Ad Hoc Values From group box, select a list, and click the Copy button. If you want to define new values for the list, leave the group box blank.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

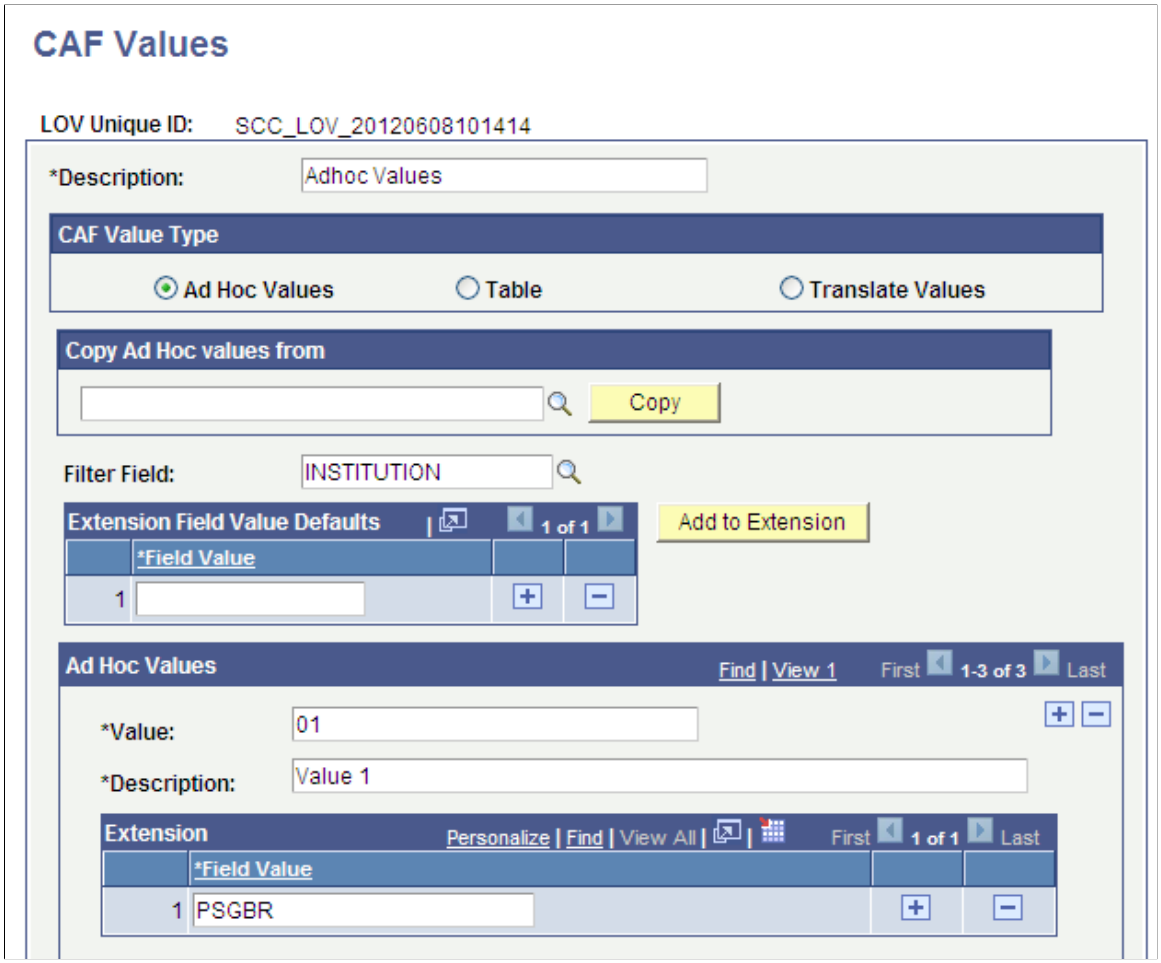
If the ad hoc list of values, which an attribute on a page displays, should be filtered based on a field value in that page, then select the field name in Filter Field. When you select a filter field, the Extension grid becomes available in the Ad Hoc Values scroll area. Use the Extension grid to specify the field value for which a value-description pair becomes available. For example, suppose you have a list of values attribute on a page and the same page has an Institution field. When a user enters *PSAUS* in the Institution field, you want the attribute to display *02* and *03* as the list of values. And when a user enters *PSGBR* in the

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Institution field, you want the attribute to display *01* and *03* as the list of values. The following example shows how to set up this scenario:

Image: Example of filtering an attribute's list of values (1 of 2)

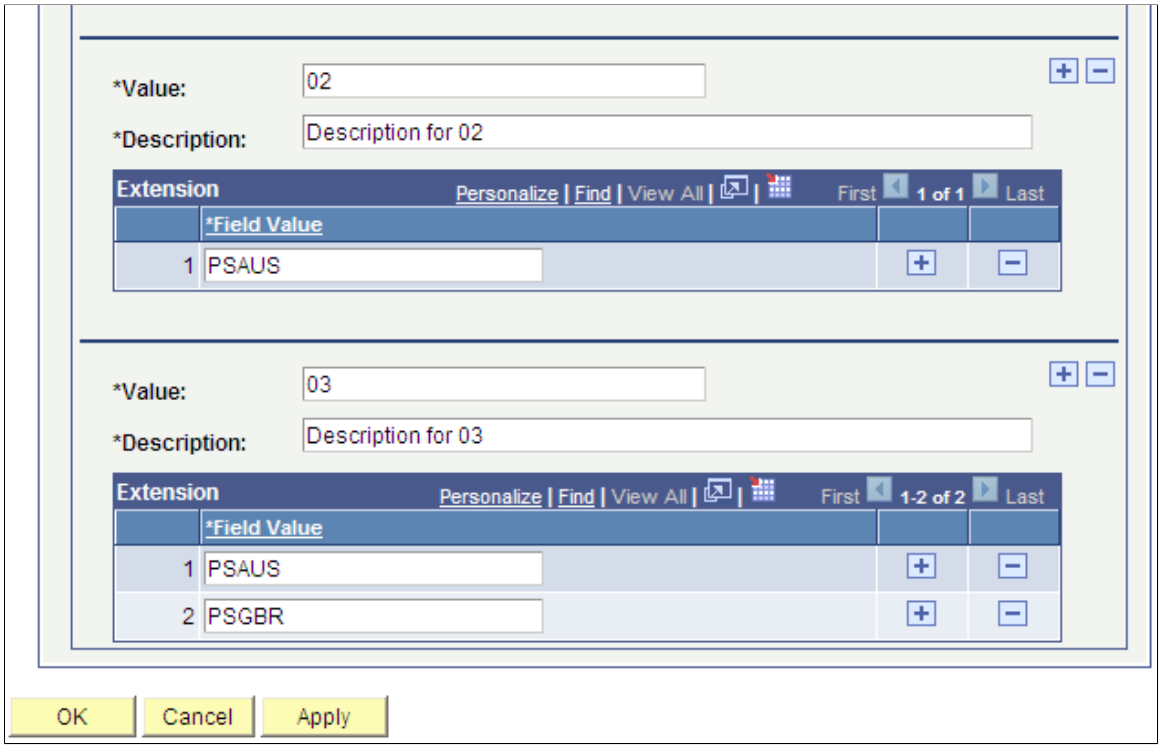
This example illustrates the fields and controls on the Example of filtering an attribute's list of values (1 of 2). You can find definitions for the fields and controls later on this page.



 Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Image: Example of filtering an attribute's list of values (2 of 2)

This example illustrates the fields and controls on the Example of filtering an attribute's list of values (2 of 2). You can find definitions for the fields and controls later on this page.



To aid in data entry, you can use the Extension Field Value Defaults grid to quickly add a field value to all the Extension grids in the Ad Hoc Values scroll area. Enter a field value in the Extension Field Value Defaults grid and then click the Add to Extension button to add the field value to all the Extension grids.

Note: Field Value is a free-form field. Therefore, you must be careful to provide the correct value for the Extension grid. The value you enter as the Field Value is case sensitive.

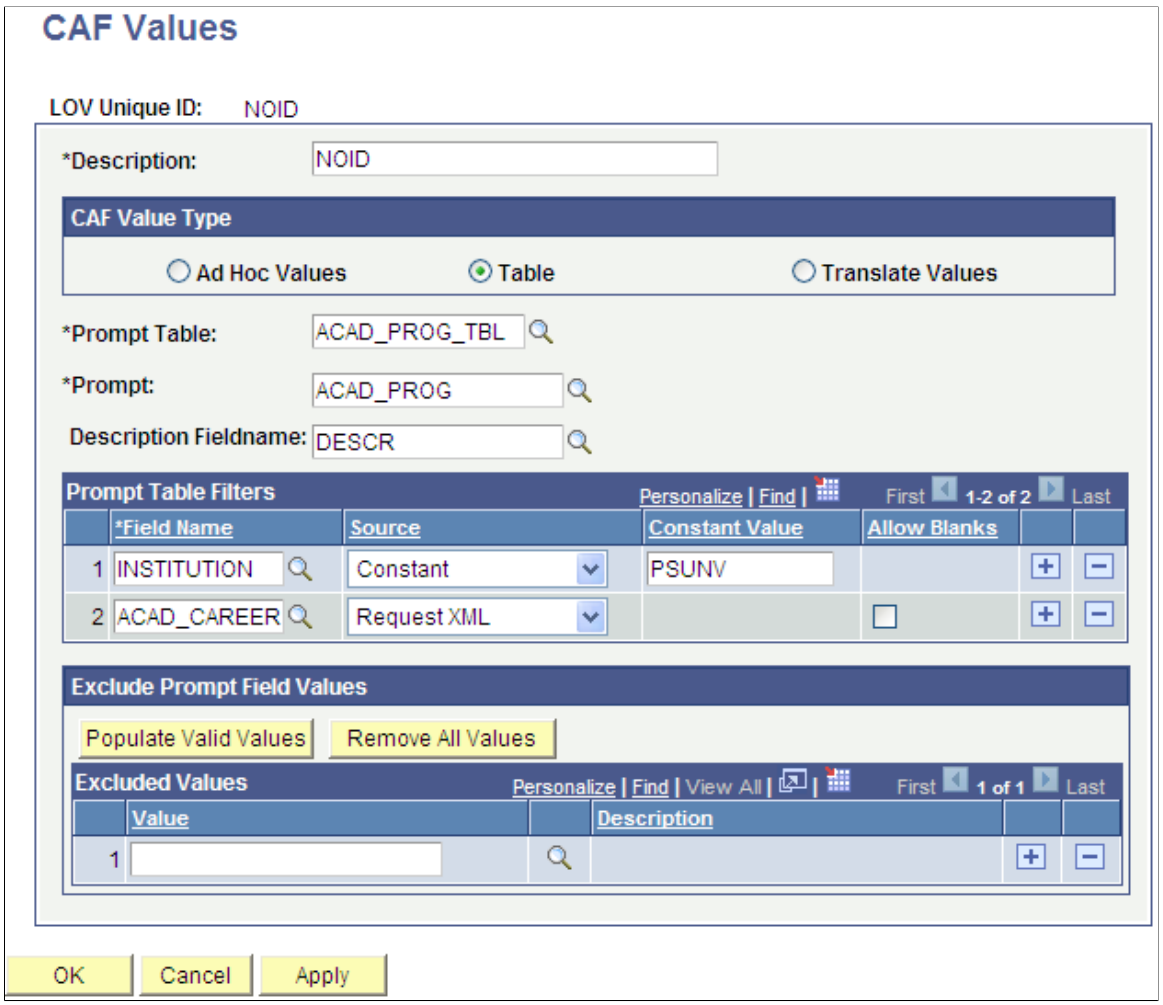
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Defining List of Values from a Table

If the CAF Value Type is *Table,* then specify the record (Prompt Table) and field (Prompt) on which the list of values should be based.

Image: CAF Values - LOV Definition page (list of values retrieved from a table)

This example illustrates the fields and controls on the CAF Values - LOV Definition page (list of values retrieved from a table). You can find definitions for the fields and controls later on this page.



Use Description Fieldname to indicate the description field for the value's description.

If there are any fields in the prompt table whose value the system should use to filter the list of values, specify these fields in the Prompt Table Filters grid.

If you select a source option of *Constant,* then you can enter the filter value in the Constant Value field. If you select a source option of *Request XML,* then the framework will try to automatically figure out the value of the filter field from the page at runtime. For example, suppose you have added an Academic Program list of values attribute on a page, and the same page displays an Institution and Academic Career field. You want the list of values to display academic programs for only the *PSUNV* institution and for the academic career which the user selects. The above screenshot shows how to set up this scenario.

To exclude any values from being displayed at runtime, specify these values in the Exclude Prompt Field Values grid. Click the Populate Valid Values button to populate the Excluded Values grid with the field's

 Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

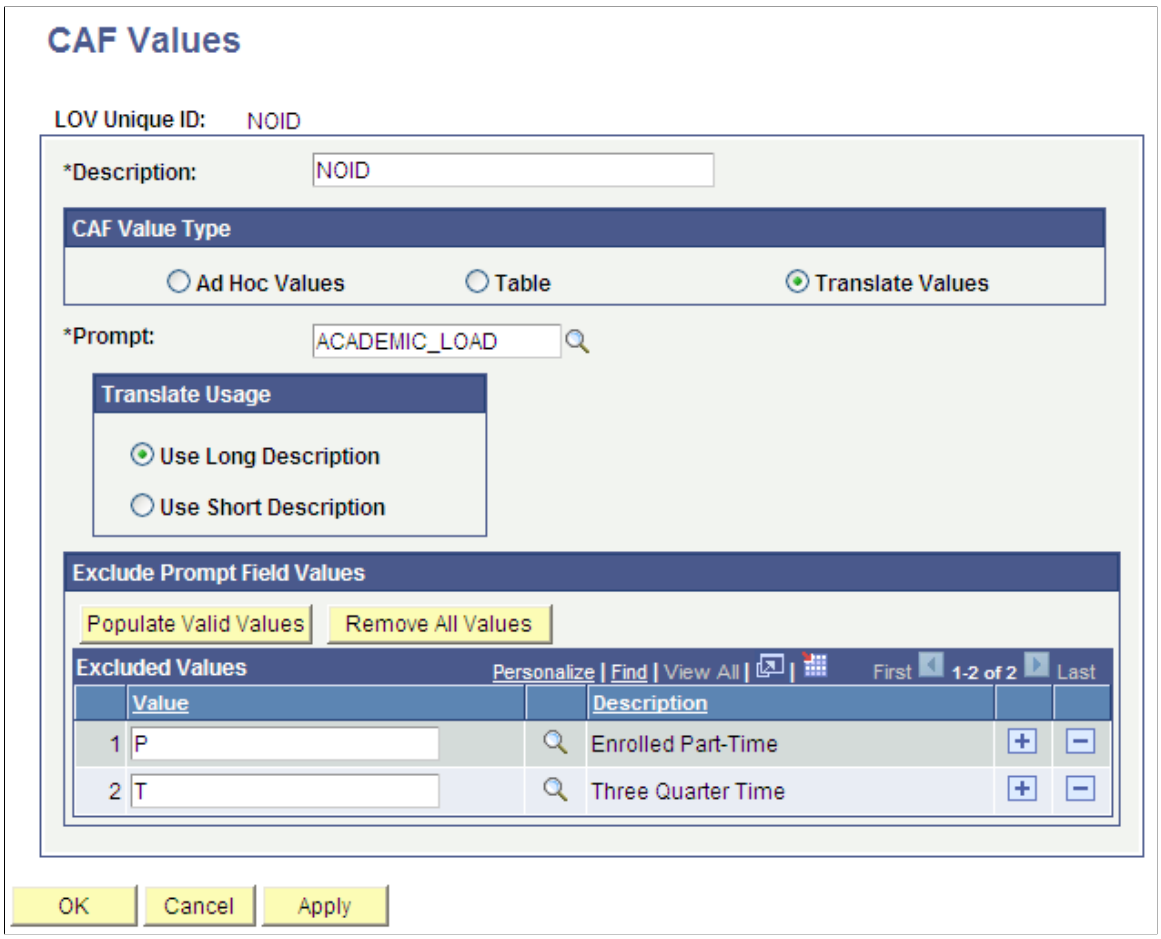
(Prompt's) values. Click the Remove All Values button to remove all the values from the Excluded Values grid.

Defining Translate List of Values

If the CAF Value Type is *Translate Values,* then specify the translate field (Prompt) on which the list of values should be based.

Image: CAF Values - LOV Definition page (list of values retrieved from a translate field)

This example illustrates the fields and controls on the CAF Values - LOV Definition page (list of values retrieved from a translate field). You can find definitions for the fields and controls later on this page.



Use the Translate Usage grid to specify if the framework should use the translate long description or short description as the value's description.

Associating a Common Attribute to a Record

After configuring the record with Application Designer and creating the attributes, extend the record with these attributes.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Pages Used to Associate a Common Attribute to a Record

Page Name	*Definition Name*	*Navigation*	*Usage*
Record Context	SCC_CAF_REC_ATTRIB	Set Up SACR, Common Definitions, Common Attributes Setup, Record Context	Associate an attribute with a record. Later on, you can modify or delete this association.
Secondary Page Options	SCC_CAF_SECPG_OPTN	Set Up SACR, Common Definitions, Common Attributes Setup, Record Context, Secondary Page Options	Set up the secondary page if you are using SCC_CAF_LAUNCH_SBP subpage.

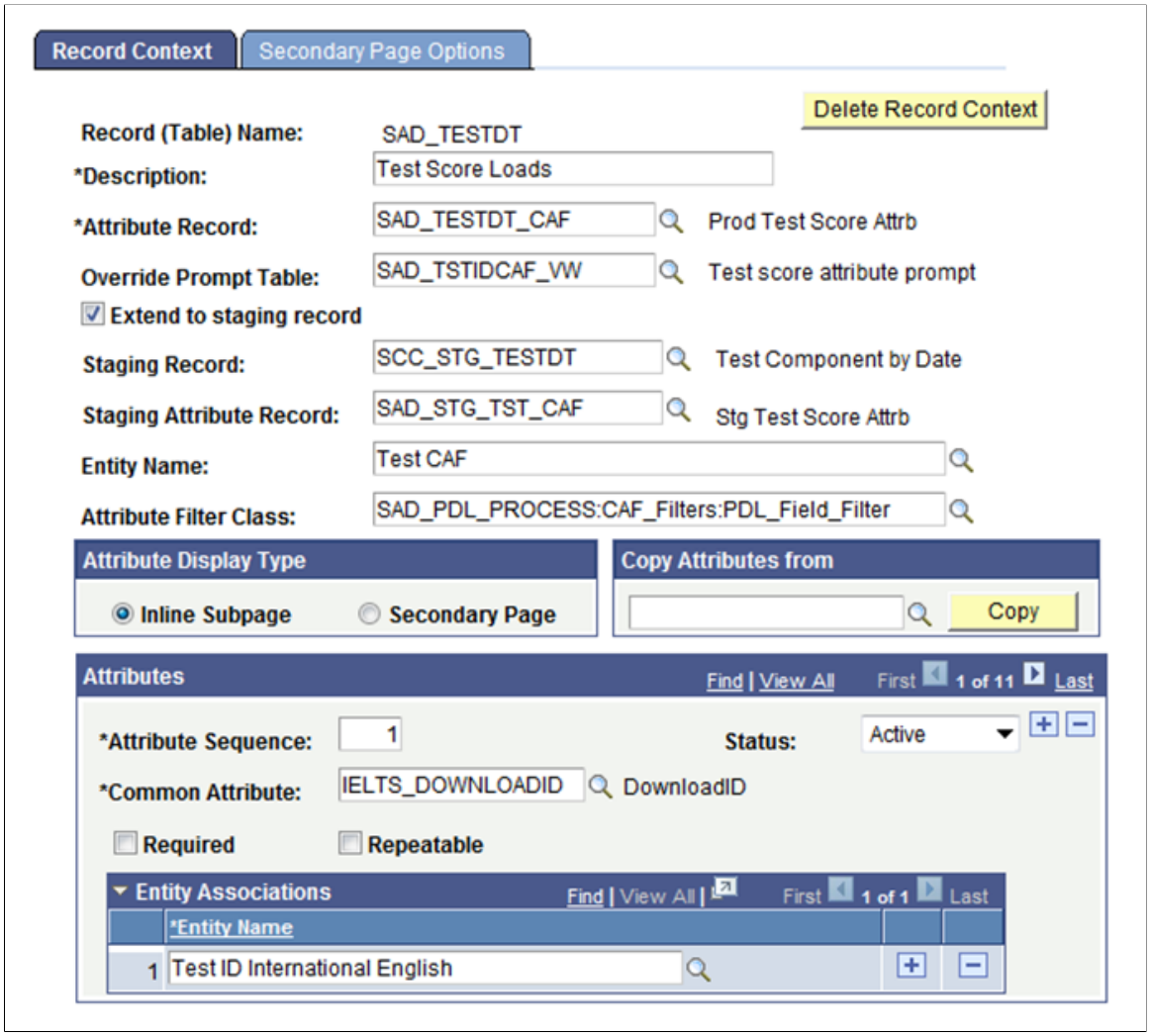
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Associating a Common Attribute to a Record

Access the Record Context page (Set Up SACR, Common Definitions, Common Attributes Setup, Record Context).

Image: Record Context page

This example illustrates the fields and controls on the Record Context page. You can find definitions for the fields and controls later on this page.



Continuing with the GENL_MATERIALS record example that was discussed in the Application Designer section: After we use the Common Attribute page to create the common attributes—Passport Number and Primary Institution—we use the Record Context page to associate the GENL_MATERIALS record with these two attributes.

Note: Extending a record with an attribute is similar to adding a new field to that record. The advantage of extending a record with an attribute is that there is no impact on any existing functionality in the system that uses this record. Additionally, it is possible to change the number and order of attributes at runtime dynamically without having any impact on existing extensions to this record.

Warning! For performance reasons, Oracle recommends that you limit the number of attributes for a given record context to no more than 25.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Record (Table) Name

Displays the record that you are extending (that is, displays the record in context).

You specify this record when you access the page in Add mode.

Delete Record Context

Available only after you save a record in context.

Description

This field is informational only and is not used in any processing.

Attribute Record

Specify the record that contains the SCC_CAF_SBR subrecord. The framework uses this mapping to store the attribute data, entered at runtime, in this record. This record should be a child record of the record in context.

Override Prompt Table

This field enables consumers of Common Attribute Framework who choose to deploy their own user interface (UI) – rather than the delivered subpage or secondary page – to use a regular edit box rather than the default behavior of a drop-down list box that lists all attribute values. This enables consumers to filter the available values using a custom view to implement their own filtering logic. For example, assume that you have an ABC list box. For user M, the ABC list box should display a different list of values. For user N, the ABC list box should display a different list of values. In such a case, you should use the Override Prompt Table field.

Extend to Staging Record

Select to indicate that any common attributes assigned to this record context is valid for its staging record. This enables consumers to share attributes rather than having to keep separate record context definitions in sync. When you select this check box, the Staging Record and Staging Attribute Record fields become available for edit.

A staging record defines a table to store data that is brought into the system through some kind of import process or interface, before the data is moved into actual production tables.

Here is an example of when you need to select this check box and use its two associated fields (Staging Record and Staging Attribute Record): Assume that user 1 uses 123 attribute with record context ABC. Record ABC is populated via a data import into staging record XYZ. In such a case, attribute 123 would need to be associated with both record contexts — ABC and XYZ. By selecting this check box and providing the staging record name (XYZ), user 1 can enable the attribute 123 (and all other attributes defined for record context ABC) for the staging record without creating a separate record context definition.

Entity Name

Use this field to map a record context definition to its corresponding common attributes entity in the Entity Registry. The Entity Registry framework uses this mapping internally, to create, retrieve, update and delete data in the attribute record (SAD_TESTDT_CAF is the attribute record in the example

 Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

graphic titled *Record Context page*). This field is used when attributes are exposed as properties of associated entities in the Entity Registry.

Attribute Filter Class

Specify a class that implements SCC_COMMON_ATTRIBUTE _FW:Interfaces:iAttributeFilter. This class provides an entity ID at runtime which is used by the framework to limit (filter) the available attributes on the transaction page at runtime. For information about creating an Attribute Filter class, see "Attribute Filter Class."

Attribute Display Type

Inline Subpage

When you save the page and more than twenty five attributes are present in the Attributes scroll area, the system displays a warning message. Select the Inline Subpage option if you do not want the system to display this message.

The inline subpage (SCC_CAF_SBP) can display any number of attributes in the transaction component. So, this limit of twenty five attributes is not applicable to the inline subpage.

Secondary Page

When you select this option:

- The fields on the Secondary Page Options page become available for edit.
- The system displays a warning message when you save the Record Context page and more than twenty five attributes are present in the Attributes scroll area.
- The Attributes scroll area does not display the Repeatable check box. Attributes cannot be repeated in a secondary page.

Copy Attributes from

To populate the list of attributes in the Attributes scroll area from another record context, specify the record name and click the Copy button. This clears any existing data in the Attributes scroll area and inserts all the attributes from the other record context.

Attributes

Use this scroll area to specify a list of attributes that you would like the record to be extended with. Remember, this is similar to adding fields to the record.

Attribute Sequence Determines the order of display of the attributes.

You can mark the attributes as required and repeatable. There is a small variation in the way the framework implements this feature in the two delivered subpages, as described in the following table:

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

SCC_CAF_LAUNCH_SBP	***SCC_CAF_SBP***
Implication if you select the Required check box: The user must enter a value for this attribute before save. If the user does not enter a value and clicks the Save button, the system displays an error message: *Field is required.*	Implication if you select the Required check box: With this subpage, the user selects the attribute from a down-down list and then enters the value (refer to the example screenshot titled *Attributes on the primary page* in the "Using Application Designer to Configure the Record" section). The system inserts the attributes identified as required into the subpage at run time and the user must enter a value in order to save the component. A user cannot delete a required attribute because the system disables the delete button for the required rows. Also, the user cannot change the required attribute to a non-required attribute.
Implication if you select the Repeatable check box: This feature is not applicable here.	Implication if you select the Repeatable check box: The user can enter multiple instances of this attribute and its corresponding values in the scroll area. If you do not select the Repeatable check box and the user tries to enter multiple instances of this attribute in the scroll area, the system displays an error message.

Entity Association

Enter the entities for which the attribute is valid. If the Attribute filter class provides an entity at runtime that matches an entity entered here, then the system displays this attribute on the runtime page. Conversely, if the Attribute filter class provides an entity at runtime that does *not* match an entity entered here, then the system does not display the attribute on the runtime page.

If an attribute's entity association grid is blank, then the system does not filter the attribute and the attribute is always available on the runtime page irrespective of the entity provided by the attribute filter class at runtime.

Related Links

Setting Up Entity Registry

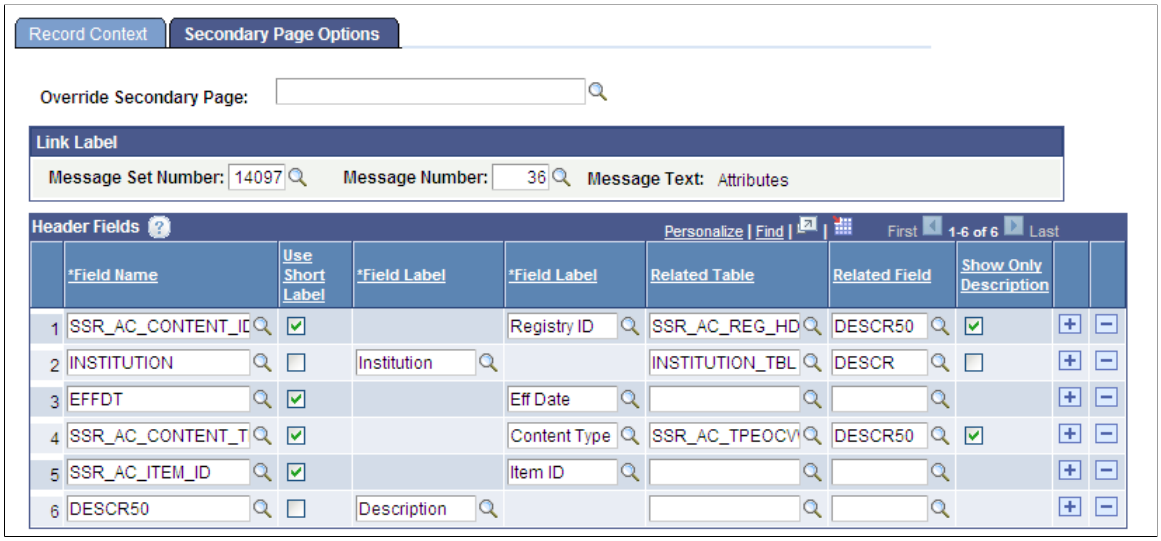
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setting Up Secondary Page Options

Access the Secondary Page Options page (Set Up SACR, Common Definitions, Common Attributes Setup, Record Context, Secondary Page Options).

Image: Secondary Page Options page

This example illustrates the fields and controls on the Secondary Page Options page. You can find definitions for the fields and controls later on this page.



Note: The fields on this page are available for edit only if you have selected the Secondary Page option on the Record Context page.

Link Label

The default hyperlink label in the subpage SCC_CAF_LAUNCH_SBP is *Attributes*. If you want a different label, specify the message catalog entry of the required text in this group box and that label is displayed on the page.

See *PeopleTools: System and Server Administration*, Message Catalog.

Header Fields

Optionally, you can use this grid to define header fields for the CAF_LAUNCH_SBP subpage. The fields you enter here display as read-only fields on the header region of the subpage.

Field Name Select the field you want to display on the header region of CAF_LAUNCH_SBP. The prompt lists all of the fields in the record context (GENL_MATERIALS in the example).

Use Short Label Select this check box to use the default short name as the field label.

Field Label Indicates the field label that the system displays on the header region of CAF_LAUNCH_SBP.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

When you select a Field Name, the Field Label field is automatically populated with the default long name. If you select the Use Short Label check box, the Field Label field is automatically populated with the default short name.

If more names are available and you do not want to use the default value, you can select a different value as the field label.

Note: If the Header Fields grid contains rows where Use Short Label is selected for at least one but not all field names, then the grid displays two Field Label columns. The left column lists the long names, while the right column lists the short names.

Related Table

The prompt lists all tables where the selected field (Field Name) exists. This enables you to add a related display field to the header. For example, if the field INSTITUTION is added to the header, there might be a need to display the description from the INSTITUTION_TBL.

Related Field

Prompts against the selected related table. The field you select here appears on the header. Continuing with the previous example, if you select INSTITUTION_TBL as the related table, then to display the institution's description you must select DESCR, DESCRFORMAL, or DESCRSHORT as the related field.

Show Only Description

This check box is available only when you select a related field.

When this check box is selected, the subpage header displays only the related field value's description (rather than both the field value and the related field value's description). For example, suppose you select this check box for the EMPLID field name, select ID as the field label, select PERSONAL_DATA as related table, and select FIRST_NAME as related field. In such a case, if you select the Show Only Description check box, only the first name is displayed next to the ID label (for example, the header displays *ID: Arlene*). If you do not select the Show Only Description check box, both the ID number and first name are displayed (for example, *ID: 3765 Arlene*).

Writing Custom Logic for Validation and Formatting

On the Common Attribute page, it is possible to associate formatting and validation logic with an attribute. If the delivered list of attribute format options is not sufficient, you can write your own application class with custom logic and associate that class with the attribute on the Common Attribute page.

The custom class must extend the interface SCC_COMM_ATTRIBUTE_FW:Interfaces:iAttributeModifier.

 Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

This interface has two methods: *Validate* and *Format.*

Validate method:

Write PeopleCode in this method that validates the data in the attribute. The framework invokes this method when the user tabs out of the attribute field after entering a value or clicks the Save button after entering the attribute value. If you want to display a message, populate the *&p_msg* object with the required message. The framework shows this message when the user tabs out of the attribute field after entering a value or clicks the Save button after entering the attribute value. The framework picks the severity of the message from the message catalog entry's severity.

Here is the sample code for populating the *&p_msg* object.

```
method validate

   /+ &p_value as Any, +/

   /+ &p_msg as SCC_COMMON:ENTITY:LOG:MessageLogBase out +/

   /+ Returns Boolean +/

   /+ Extends/implements

SCC_COMMON_ATTRIBUTE_FW:Interfaces:iAttributeModifier.validate +/

   Local SCC_COMMON:ENTITY:LOG:MessageLogBase &msg =
create SCC_COMMON:ENTITY:LOG:MessageLogBase();

   Local SCC_COMMON:ENTITY:LOG:MessageEntry &me;

   If Len(String(&p_value)) >= 3 Then

      Local Record &rec = CreateRecord(Record.COUNTRY_TBL);

      &rec.COUNTRY.Value = String(Substring(&p_value, 1, 3));

      If &rec.SelectByKey() Then

         Return True;

      Else

         &me = create SCC_COMMON:ENTITY:LOG:MessageEntry();

         &me.MsgSet = 14098;

         &me.MsgID = 10;

         &me.Severity = &me.Severity_Error;

         &msg.writeEntry(&me);

         &p_msg = &msg;

         Return False;

      End-If;

   Else

      &me = create SCC_COMMON:ENTITY:LOG:MessageEntry();

      &me.MsgSet = 14098;

      &me.MsgID = 11;

      &me.Severity = &me.Severity_Error;
```

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

```
      &msg.writeEntry(&me);

      &p_msg = &msg;

      Return False;

   End-If;

   end-method;
```

The return value of this method indicates if the validation passed or failed. If the return value is *True,* then the framework assumes that the validation has passed and continues processing. But if the return value is *False,* the framework halts processing and displays the messages if any in the *&p_msg* object. Based on the severity of the message, the user is either forced to make changes to the attribute data (severity error) or just dismiss the message and continue with other attributes on the page.

Format method

The framework invokes this method on *fieldchange* event. Write PeopleCode in this method that formats the attribute value. The framework then displays the formatted value on the page.

Here is an example of formatting an attribute value to appear as a US zip code.

```
method format

   /+ &p_value as Any +/

   /+ Returns String +/

   /+ Extends/implements
SCC_COMMON_ATTRIBUTE_FW:Interfaces:iAttributeModifier.format +/

   Local string &str = String(&p_value);

      If Len(&str) = 9 Then

      Return (Substring(&str, 1, 5) | "-" | Substring(&str, 5, 4));

   Else

      Return &str;

   End-If;

end-method;
```

Understanding the Common Attribute API

The Common Attribute Application Programming Interface (API) is intended to simplify the access and manipulation of attributes through PeopleCode. If attributes need to be accessed and manipulated in a different way compared to the delivered generic subpages, the framework provides some utility methods and classes that have a simple interface for manipulating attribute data.

The API supports retrieving attributes based on record context definitions. An attribute object received through this API can then be used to set the data, validate, format, get list of values, and save to backend. The CAF API is internally used by the delivered generic subpages and subrecord.

The API is contained in the application package: *SCC_COMMON_ATTRIBUTE_FW.* The *SCC_COMMON_ATTRIBUTE_FW:AttributeManager* application class provides a set of utility methods

 Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

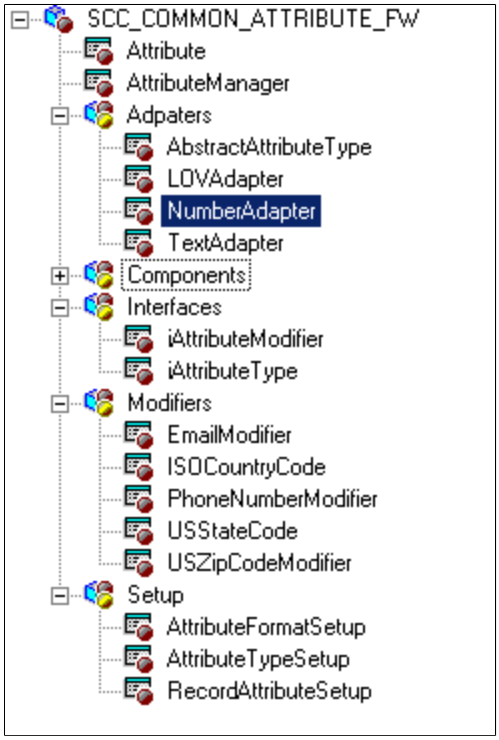
that can be used to fetch relevant attributes. This class behaves as a sort of object factory that provides attribute objects.

Attribute objects are instances of the *SCC_COMMON_ATTRIBUTE_FW:Attribute* class. This class encapsulates all the basic properties and behaviors of an attribute. It is possible to manipulate an attribute using its various methods. For the complete list of available methods in this class, see Common Attribute Framework Application Class Reference.

The following diagram shows the application packages and classes in the Common Attribute Framework API:

Image: Common Attribute Framework API

This example illustrates the fields and controls on the Common Attribute Framework API. You can find definitions for the fields and controls later on this page.



Using and Creating an Attribute Filter Class

An attribute filter class can be used to control the availability of attributes in a transaction page. You can specify the attribute filter class in the Record Context page.

The Common Attribute Framework creates an instance of this class at runtime and accesses its EntityID property. All the attributes that are mapped with this entity ID in the Record Context page are available in the transaction page. Any attribute that does not have an entity association is also available.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Creating an Attribute Filter Class

Attribute Filter classes must implement the interface SCC_COMMON_ATTRIBUTE_FW:Interfaces:iAttributeFilter. Implementing teams can then code the getter method of EntityID property to return a value based on any runtime data like component buffer values or external table values.

Here is an example of an attribute filter class:

```
import SCC_COMMON_ATTRIBUTE_FW:Interfaces:iAttributeFilter;

class AIAFilter implements
SCC_COMMON_ATTRIBUTE_FW:Interfaces:iAttributeFilter

   property Record ParentRecord;

   property string EntityID get;

end-class;

get EntityID

   /+ Returns String +/

   /+ Extends/implements
SCC_COMMON_ATTRIBUTE_FW:Interfaces:iAttributeFilter.EntityID +/

   rem Local Rowset &rs = GetLevel0()(1).GetRowset(Scroll.SSR_AIR_HDR);

   Local Record &rec = CreateRecord(Record.SSR_ITEM_TYPE);

   rem &rec.SSR_ITEM_TYPE.Value =
&rs(CurrentRowNumber(&rs.Level)).SSR_AIR_HDR.SSR_ITEM_TYPE.Value;

   &rec.SSR_ITEM_TYPE.Value = &ParentRecord.SSR_ITEM_TYPE.Value;

   &rec.SelectByKey();

   Return &rec.SCC_ENTITY_ID.Value;

end-get;
```

The property ParentRecord is populated by the framework with the runtime component buffer object of the parent record. This object can then be used to access other records, rows, rowsets, and so on from the component buffer thereby giving access to any field value in the runtime component.

Common Attribute Framework Application Class Reference

This section discusses the application classes that you can use for accessing and manipulation of attributes through PeopleCode.

Attribute Manager

class SCC_COMMON_ATTRIBUTE_FW:AttributeManager

This class provides a set of utility methods that can be used to fetch relevant attributes. This class behaves as a sort of object factory that provides attribute objects.

 Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Summary

Constructor Summary

public	AttributeManager ()

Method Summary

public boolean	deleteAllAttributes (Record p_parentRec) Deletes all attributes associated with a record
public array of Attribute	getActiveRecordAttributes (Record p_parentRec) Returns all active attributes associated with a record
public Attribute	getAttributeFromChildRecord (Record p_childRec) Returns an Attribute object based on the attribute record
public any	getAttributeValue (Record p_rec, string p_AttributeName) Returns the value of an attribute from the child record based on the name of the attribute provided
public array of Attribute	getFilteredRecordAttributes(Record p_parentRec, string p_EntityID) Returns all attributes associated with a record filtered by Entity ID
public Attribute	getRecordAttribute (Record p_parentRec, string p_AttributeName) Returns attribute associated with a record
public array of Attribute	getRecordAttributes (Record p_parentRec) Returns all attributes associated with a record
public array of Attribute	getRecordAttributesV2 (Record p_parentRec) Returns all attributes associated with a record
public void	setAttributeToChildRecord (Record p_childRec out, Attribute p_Attribute) Sets the correct field in the record with value in the Attribute Object

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

public void	setAttributeValue (Record p_rec out, string p_AttributeName, any p_AttributeValue) Sets the value of an attribute from the child record based on the name of the attribute provided

Detail

Constructor Detail

public	AttributeManager ()

Method Detail

deleteAllAttributes	public boolean deleteAllAttributes (Record p_parentRec) Deletes all attributes associated with a record. Parameters: p_parentRec - The record which has been extended with attributes. Returns: boolean
getActiveRecordAttributes	public boolean deleteAllAttributes (Record p_parentRec) Returns all active attributes associated with a record Parameters: p_parentRec - The record which has been extended with attributes. Returns: Array of Attribute

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

getAttributeFromChildRecord	public Attribute getAttributeFromChildRecord (Record p_childRec) Returns an Attribute object based on the attribute record. Parameters: p_childRec - The record populated with data from a row in the attribute child record containing the subrecord SCC_CAF_SBR. Returns: Attribute
getAttributeValue	public any getAttributeValue (Record p_rec, string p_AttributeName) Returns the value of an attribute from the child record based on the name of the attribute provided. Parameters: • p_rec - The record populated with data from a row in the attribute child record containing the subrecord SCC_CAF_SBR. • p_AttributeName - Name of the attribute as string. Returns: any
getFilteredRecordAttributes	public boolean deleteAllAttributes (Record p_parentRec) Returns all attributes associated with a record filtered by entity ID. Parameters: • p_parentRec - The record which has been extended with attributes. • string p_EntityID Returns: Array of Attribute

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

getRecordAttribute	public Attribute getRecordAttribute (Record p_parentRec, string p_AttributeName) Returns attribute associated with a record. Parameters: • p_parentRec - The record which has been extended with attributes. • p_AttributeName - Name of the attribute as string. Returns: Attribute
getRecordAttributes	public array of Attribute getRecordAttributes (Record p_parentRec) Returns all attributes associated with a record. Parameters: p_parentRec - The record which has been extended with attributes. Returns: array of Attribute
getRecordAttributesV2	public array of Attribute getRecordAttributesV2 (Record p_parentRec) Returns all attributes associated with a record. Parameters: p_parentRec - The record which has been extended with attributes. Returns: array of Attribute

 Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

setAttributeToChildRecord	public void setAttributeToChildRecord (Record p_childRec out, Attribute p_Attribute) Sets the correct field in the record with value in the Attribute Object. Parameters: • p_childRec - The record populated with data from a row in the attribute child record containing the subrecord SCC _CAF_SBR. • p_Attribute - Instantiated Attribute Object.
setAttributeValue	public void setAttributeValue (Record p_rec out, string p_ AttributeName, any p_AttributeValue) Sets the value of an attribute from the child record based on the name of the attribute provided. Parameters: • p_rec - The record populated with data from a row in the attribute child record containing the subrecord SCC_CAF _SBR. • p_AttributeName - Name of the attribute as string. • p_AttributeValue - Value of the attribute; data type depends on the attribute type.

Class Attribute

class SCC_COMMON_ATTRIBUTE_FW:Attribute

This class encapsulates all the basic properties and behavior of an attribute. It is possible to manipulate an attribute using its various methods.

Summary

Property Summary

public Array of string	arrRequestKeyField
public iAttributeModifier	AttributeModifier
public iAttributeType	AttributeType
public Record	DataRecord
public string	DataType readonly

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

public string	DESCR
public string	FIELDNAME
public string	FormattedValue
public Boolean	HasLOV
public Boolean	isRepeatable
public Boolean	isRequired
public StringHashMap	LOVKeys
public string	RECNAME
public string	SCC_APPCLASSNAME
public string	SCC_CAF_ALLOW_BLNK
public string	SCC_CAF_ALLOW_MULT
public string	SCC_CAF_ATR_FORMAT
public number	SCC_CAF_ATTR_DEC
public string	SCC_CAF_ATTR_DEF
public number	SCC_CAF_ATTR_INT
public string	SCC_CAF_ATTR_TYPE
public string	SCC_CAF_ATTRIB_NM
public string	SCC_CAF_FIELDNAME
public string	SCC_SL_LOV_CONTEXT
public Boolean	SelfServiceMode
public any	Value

Constructor Summary

public	Attribute (string p_SCC_CAF_ATTRIB_NM, Record p_rec) Constructor

Method Summary

 Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

public void	close ( )
public boolean	delete (MessageLogBase p_msg out) Deletes the attribute from the backend table.
public string	format (any p_value) This method contains logic related to formatting of the attribute data. This is driven by the attribute type associated with the attribute. If a modifier or validator class is associated with this attribute then the format method in that class is invoked instead. Value passed in as parameter is validated in this method. This is useful for validating data in a field in the component buffer.
public StringHashMap	getListOfValues ( ) Returns the list of values (code and description pair) associated with this attribute.
public void	init (Record p_rec) This methods contains logic to initialize properties and variables. This is called by the framework after the object is created.
public boolean	save (MessageLogBase p_msg out) Saves the attribute to the backend table.
public void	setDefault ( ) This methods contains logic to initialize attribute data with default values.
public boolean	validate (MessageLogBase p_msg out) This method contains validation logic. Validations are driven by the attribute type associated with the attribute. If a modifier or validator class is associated with this attribute then the validate method in that class is invoked instead. Value of the attribute is validated in this method.
public boolean	validateValue (any p_value, MessageLogBase p_msg out) This method contains validation logic. Validations are driven by the attribute type associated with the attribute. If a modifier or validator class is associated with this attribute then the validate method in that class is invoked instead. Value passed in as parameter is validated in this method. This is useful for validating data in a field in the component buffer.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Detail

Property Detail

arrRequestKeyField	public Array of string
AttributeModifier	public iAttributeModifier
AttributeType	public iAttributeType
DataRecord	public Record
DataType	readonly public string
DESCR	public string
FIELDNAME	public string
FormattedValue	public string
HasLOV	public boolean
isRepeatable	public boolean
isRequired	public boolean
LOVKeys	public StringHashMap
RECNAME	public string
SCC_APPCLASSNAME	public string
SCC_CAF_ALLOW_BLNK	public string
SCC_CAF_ALLOW_MULT	public string
SCC_CAF_ATR_FORMAT	public string
SCC_CAF_ATTR_DEC	public number
SCC_CAF_ATTR_DEF	public string
SCC_CAF_ATTR_INT	public number
SCC_CAF_ATTR_TYPE	public string
SCC_CAF_ATTRIB_NM	public string
SCC_CAF_FIELDNAME	public string

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

SCC_SL_LOV_CONTEXT	public string
SelfServiceMode	public boolean
Value	public any

Constructor Detail

Attribute	public Attribute (string p_SCC_CAF_ATTRIB_NM, Record p _rec) Constructor Parameters: • p_SCC_CAF_ATTRIB_NM - Name of the attribute. • p_rec - Data record of the attribute containing the SCC_ CAF_SBR subrecord.

Method Detail

delete	public boolean delete (MessageLogBase p_msg out) Deletes the attribute from the backend table. Parameters: p_msg Returns: boolean
format	public string format (any p_value) This method contains logic related to formatting of the attribute data. This is driven by the attribute type associated with the attribute. If a modifier or validator class is associated with this attribute then the format method in that class is invoked instead. Value passed in as parameter is validated in this method. This is useful for validating data in a field in the component buffer. Parameters: p_value - The value to be formatted. Returns: string

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

getListOfValues	public StringHashMap getListOfValues ( ) Returns the list of values (code and description pair) associated with this attribute. Returns: StringHashMap
init	public void init (Record p_rec) This method contains logic to initialize properties and variables. This is called by the framework after the object is created. Parameters: p_rec - Record containing the SCC_CAF_SBR subrecord.
save	public boolean save (MessageLogBase p_msg out) Saves the attribute to the backend table. Parameters: p_msg Returns: boolean
setDefault	public void setDefault ( ) This method contains logic to initialize attribute data with default values.
validate	public boolean validate (MessageLogBase p_msg out) This method contains validation logic. Validations are driven by the attribute type associated with the attribute. If a modifier or validator class is associated with this attribute then the validate method in that class is invoked instead. Value of the attribute is validated in this method. Parameters: p_msg - This object is populated with any validation messages generated in the method. Returns: boolean

 Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

validateValue	public boolean validateValue (any p_value, MessageLogBase p_msg out) This method contains validation logic. Validations are driven by the attribute type associated with the attribute. If a modifier or validator class is associated with this attribute then the validate method in that class is invoked instead. Value passed in as parameter is validated in this method. This is useful for validating data in a field in the component buffer. Parameters: • p_value - The value to be validated. • p_msg - This object is populated with any validation messages generated in the method. Returns: boolean

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.