

Business Center™

Customization Guide

Release 9.0

Last Updated in HF1

May 2010

Sterling Commerce
An IBM Company

© Copyright 2010 Sterling Commerce, Inc. All rights reserved.

Additional copyright information is located on the documentation library:
<http://www.sterlingcommerce.com/Documentation/MCSF90/CopyrightPage.htm>

Contents

Overview of Business Center Customization	5
Setting Up the Customization Environment	6
Customize Business Center Using Web UI Framework	7
Differential Screen Extensions	7
Override Screen Extensions	7
Mashups	8
Struts	8
Customizing the web.xml File for Development	8
Customizing the web.xml File for Deployment	8
Resource Bundles	9
Customize Advanced Search Criteria for Business Center	10
Add or Modify a Search Attribute	10
Adding a Search Attribute - An Example	15
Modifying a Search Attribute - An Example	16
Remove a Search Attribute	17
Removing a Search Attribute - An Example	17
Customize Related Tasks for Business Center	19
Add a Related Task Category	20
Modify a Related Task Category	21
Add a Related Task	21
Modify a Related Task	21
Delete a Related Task	22
Delete a Related Task Category	22
Customize the Conditions in an Approval Rule for Business Center	23
Add a New Entity Attribute to a Domain	23
Modify an Entity Attribute in a Domain	27
Add a New Query Type to an Entity Attribute	27
Delete an Existing Query Type from an Entity Attribute	28
Delete an Entity Attribute from a Domain	28
Deploying Web UI Framework Customization	29

Index

30

Overview of Business Center Customization

The Business Center™ application can be customized based on the business requirements of users. A user may want to make changes in the way information is displayed in the user interface (UI) without changing the functionality. This topic provides an overview of the types of customizations possible in the Business Center application.

You can customize the following in Business Center:

- Search criteria used in the Advanced Search screens in Business Center. For more information about customizing the advanced search criteria, refer to the topic, *Customize Advanced Search Criteria for Business Center*.
- Related tasks and the categories under which the related tasks are grouped in the **Related Tasks** panel pertaining to a screen. For more information about customizing related tasks, refer to the topic, *Customize Related Tasks for Business Center*.
- Conditions displayed in the Approval Rule Summary screen. For more information about customizing conditions, refer to the topic, *Customize the Conditions in an Approval Rule for Business Center*.

In addition to the advanced search criteria, related tasks, and conditions, the following Web UI Framework components can also be customized:

- **Override Screen extensions:** You can customize a screen by completely replacing it.
- **Differential Screen extensions:** You can add a new field or hide an existing field in the UI. Moreover, you can also modify the properties of the out-of-the-box fields.
- **New UI screens:** You can use the canvas in the Designer Workbench to create the actual user interface that will be used by the Business Center application.
- **New menu entry:** You can use the Configurator to create a new menu entry.
- **Mashup extensions:** You can customize the input XML and the output template of an API call. In addition to this, you can define new mashups also.
- **Resource bundles:** You can define new bundle entries and override the out-of-the-box bundle entries.
- **Struts extensions:** You can define new Struts and override the existing Struts.
- **Cascading Style Sheets (CSS):** You can create new CSS entries.
- **Data types:** You can define new data types and modify the existing data types.
- **Login screen:** You can define your own Login screen and use it to log in to Business Center.

After you customize an existing screen, you must deploy your changes in the application. For more information about customizing the Web UI Framework components and deploying the extensions, refer to the *Selling and Fulfillment Foundation: Customizing the Web UI Framework Guide*.

Note that the Business Center application is built on the Ext JS JavaScript framework, version 2.0.

You can also extend the Business Center database. For example, you can add a column to a standard tables by modifying the entity database extension XML files. For more information about extending the database, refer to the *Selling and Fulfillment Foundation: Extending the Database Guide*.

Note: .To monitor the execution of a program that is run to trap errors when customizing Business Center, you can use the debugging tools provided with the application. For more information about the debugging tools, refer to the *Selling and Fulfillment Foundation: Customizing the Web UI Framework Guide*.

Setting Up the Customization Environment

The Business Center™ application can be customized based on the business requirements of users. This topic provides information about setting up the customization environment.

Business Center must be customized using the `sbcdev.war` file, and not the `sbc.war` file. The `sbcdev.war` file contains additional folders and `web.xml` entries that are required for customization. You must deploy the `sbc.war` file in the production server with the required extensions that are created using the `sbcdev.war` file.

To set up the customization environment:

1. Run the following command to create the `sbcdev.war` file:

- ⋮ For Linux/UNIX:

```
.\buildwar.sh -Dappserver=<application server>  
-Dwarfiles=sbcdev
```

- ⋮ For Windows:

```
.\buildwar.cmd -Dappserver=<application server>  
-Dwarfiles=sbcdev
```

2. Deploy the `sbcdev.war` file.

3. Start the application server by passing the following argument:

```
-Dwufdevmode=true
```

Customize Business Center Using Web UI Framework

You can customize the user interface of an existing out-of-the-box installation of the Business Center™ application using the Extensibility Workbench. The Extensibility Workbench allows you to customize the existing installation at runtime without recompiling or changing the original source code.

The Business Center application is built on the Ext JS JavaScript framework, version 2.0. The JavaScript API Documentation contains information about the JavaScript utility and classes used by the application. The JavaScript API Documentation can be accessed using the following URL:

```
http://<server>:<port>/<context_root>/yfscommon/jsdocs/20/index.html
```

where `context_root` is the context root of the `smcfsdocs.war` file, which is used for the context-sensitive help. For more information about deploying the context-sensitive help, refer to the *Selling and Fulfillment Foundation: Installation Guide*.

You can perform the following customizations:

- † [Differential Screen Extensions](#)
- † [Override Screen Extensions](#)
- † [Mashups](#)
- † [Struts](#)
- † [Customizing the web.xml File for Development](#)
- † [Customizing the web.xml File for Deployment](#)
- † [Resource Bundles](#)

For more information about customizing the Web UI Framework components and deploying the extensions, refer to the *Selling and Fulfillment Foundation: Customizing the Web UI Framework Guide*.

Differential Screen Extensions

Differential extensibility enables you to customize parts of a screen. You can add a new field, or hide an existing field in the user interface. Additionally, you can modify the properties of the out-of-the-box fields. The Java Script Builder (JSB) files are located in the `<INSTALL_DIR>/repository/eardata/sbc/war/builder/sbc` directory.

For more information about differential extensibility, refer to the *Selling and Fulfillment Foundation: Customizing the Web UI Framework Guide*.

Override Screen Extensions

Override extensibility enables you to replace an existing screen with an entirely new screen.

For more information about override extensibility, refer to the *Selling and Fulfillment Foundation: Customizing the Web UI Framework Guide*.

Mashups

You can customize the input XML template, and the output XML template of an API call. Additionally, you can define new mashup layers. The Business Center mashups are located in the `<INSTALL_DIR>/repository/eardata/sbc/war/mashupxmls/sbc` directory. You must place the extended mashups in the `<INSTALL_DIR>/extensions/sbc/webpages` directory.

The Business Center mashups use the following mashup types:

- XAPI (for XAPI calls)
- AggregateXAPI (for multiple mashups)
- SBCAPI (for application-specific mashups)

For more information about extending mashups, refer to the *Selling and Fulfillment Foundation: Customizing the Web UI Framework Guide*.

Struts

You can define new Struts and override the existing Struts. The Business Center struts files are located in the `<INSTALL_DIR>/repository/eardata/sbc/war/WEB-INF/classes` directory. The `struts.xml.sample` and `struts.properties.sample` files are located in the `<INSTALL_DIR>/repository/eardata/sbc/extn` directory.

For more information about creating and extending a Struts XML file, refer to the *Selling and Fulfillment Foundation: Customizing the Web UI Framework Guide*.

Customizing the web.xml File for Development

During development, you can customize the `web.xml` file to view your customizations.

To customize the `web.xml` file during development:

1. Run the `buildwar` utility to create the `sbcdev.war` file.
2. Copy the `web.xml.sample` file from the `<INSTALL_DIR>/repository/eardata/sbcdev/extn` directory to the same directory with the file name "web.xml".
3. Modify the newly created `web.xml` files as needed.
4. If you need to add a new servlet or filter, package it in a `.jar` file, and run the `<INSTALL_DIR>/bin/install3rdParty.sh` script to include this jar file in a classpath.
5. Run the `buildwar` utility to create the `sbcdev.war` file.

Customizing the web.xml File for Deployment

After performing the customizations, you must customize the `web.xml` file for deployment.

To customize the `web.xml` file for deployment:

1. Run the `buildear` or `buildwar` utility to create the `sbc.war` file.

2. Copy the `web.xml.sample` file from the `<INSTALL_DIR>/repository/eardata/sbc/extn` directory to the same directory with the file name "web.xml".
3. Modify the newly created `web.xml` files as needed.
4. If you need to add a new servlet or filter, package it in a `.jar` file, and run the `<INSTALL_DIR>/bin/install3rdParty.sh` script to include this jar file in a classpath.
5. Run the `buildear` or `buildwar` utility to create the `sbc.war` file.

Resource Bundles

You can define new bundle entries and override the out-of-the-box bundle entries. The client-side bundle files are located in the `<INSTALL_DIR>/repository/eardata/sbc/war/sbc/resources` directory. To load the client-side bundles, the `sbcBundle` target is used. This target is defined in the `sbcCore.jsb` file located in the `<INSTALL_DIR>/repository/eardata/sbc/war/builder/sbc` directory.

For more information about changing bundle files, refer to the *Selling and Fulfillment Foundation: Customizing the Web UI Framework Guide*.

Customize Advanced Search Criteria for Business Center

Business Center™ enables users to perform basic and advanced searches. You, as an implementer, can customize the advanced search criteria. The advanced search criteria can be modified by either removing a few attributes or adding a few attributes to the corresponding Advanced Search metadata file. All the customizations for the Advanced Search screens must be performed by defining an Advanced Search metadata file in the

<INSTALL_DIR>/extensions/sbc/webpages/sbc/metadata/advancedsearch folder. An Advanced Search metadata file must be created to specify the search attributes, along with details such as the controls to be displayed in the user interface (UI), query types to be used, and so on.

The following table lists the Advanced Search metadata files that must be created to customize the corresponding Advanced Search screens:

Screen Name	Advanced Search Metadata File Name
Coupon Search	Coupon.xml
Entitlement Search	Entitlement.xml
Item Search	Item.xml
Price List Search	Pricelist.xml
Pricing Rule Search	PricingRule.xml
Service Search	Service.xml
Service Option Search	ServiceOption.xml
Manual Pricing Rule Search	ManualPricingRule.xml
Approval Rule Search	ApprovalRule.xml

Add or Modify a Search Attribute

You can add a search attribute to a list of existing search attributes in an Advanced Search screen. In addition, you can also modify an existing search criteria.

To add or modify a search attribute:

1. Create a corresponding Advanced Search metadata file if it does not already exist in the <INSTALL_DIR>/extensions/sbc/webpages/sbc/metadata/advancedsearch folder.
2. Create an element, Attribute, as a child element of the root element, Attributes.
3. Set the value of the XML attribute, Action, to ADD.

4. Define the values of the other XML attributes.

Refer to the following table for more information about defining the values of the otherXML attributes:

Component	Component Type	Component Location	Description
Name	Attribute	Attributes/Attribute	Indicates the XML name of the search attribute. This component, along with the XPath, is used to set the source and target binding for the control that is displayed in the UI.
DataType	Attribute	Attributes/Attribute	Indicates the data type of the search attribute. If data Type is not provided, it is derived using the XPath and the Name attribute values. You can use the DataType attribute to override the default data types. Each data type is mapped to String, Date, or Number. If query types are not explicitly mentioned, this mapping is used to display the query types in the UI.
DisplayName	Attribute	Attributes/Attribute	Indicates the bundle key of a label that is displayed in a drop-down list for search attributes. If the label is not mentioned, the Name attribute is considered as the bundle key to display the label.
XPath	Attribute	Attributes/Attribute	Indicates the XML path of the element to which the search attribute belongs.
DefaultQueryTypes	Attribute	Attributes/Attribute	Indicates the default query type for the search attribute. If DefaultQueryTypeDesc is not set for an attribute, set it by performing this procedure: Get the supported query types by either reading the overridden query types or using the out-of-the-box query types for the data type of the attribute. Because Equal is the most commonly used query type, verify if it is one of the supported query types. If yes, use Equal as the default. Otherwise, use the first query type, which is displayed in the list of supported query types.
StringQueryTypes	Element	Attributes/Attribute	Indicates that out-of-the-box string query types are being overridden with the child elements of the StringQueryTypes element. If all the possible StringQueryTypes are not required, create one QueryType child element of StringQueryType for each query type you want to support.
DateQueryTypes	Element	Attributes/Attribute	Indicates that out-of-the-box date query types are being overridden with the child elements of the DateQueryTypes element. If all the possible date query types are not required, create one QueryType child element of DateQueryType for each query type you want to support.
NumericQueryTypes	Element	Attributes/Attribute	Indicates that out-of-the-box numeric query types are being overridden with the child elements of the NumericQueryTypes element. If all the possible NumericQueryTypes are not required, create one QueryType child element of NumericQueryType for each query type you want to support.

Component	Component Type	Component Location	Description
-----------	----------------	--------------------	-------------

QueryType Element Attributes/Attribute/StringQueryTypes OR Attributes/Attribute/DateQueryTypes OR Attributes/Attribute/NumericQueryTypes

Indicates the query types that are supported for the search attribute. Following are the query types that can be supported:

◆ Query types for strings

QueryType	QueryTypeDesc	Description
EQ	is	is
FLIKE	starts_with	starts with
LIKE	contains	contains

◆ Query types for numbers

Query Type	QueryTypeDesc	Description
EQ	is	is
GE	greater_than_or_equal_to	greater than or equal to
LE	less_than_or_equal_to	less than or equal to
NE	not_equal_to	not equal to
LT	less_than	less than
GT	greater_than	greater than

◆ Query types for dates

Query Type	QueryTypeDesc	Description
EQ	is	is
GE	greater_than_or_equal_to	greater than or equal to
LE	less_than_or_equal_to	less than or equal to
NE	not_equal_to	not equal to
LT	less_than	less than
GT	greater_than	greater than
BETWEEN	between_from_and_to	between

Component	Component Type	Component Location	Description
QueryType	Attribute	Attributes/Attribute/ StringQueryTypes/ QueryType OR Attributes/Attribute/ DateQueryTypes/ QueryType OR Attributes/Attribute/ NumericQueryTypes/ QueryType	Indicates the query type. For more information about the supported query types, refer to the description of the QueryType Element.
QueryType Desc	Attribute	Attributes/Attribute/ StringQueryTypes/ QueryType OR Attributes/Attribute/ DateQueryTypes/ QueryType OR Attributes/Attribute/ NumericQueryTypes/ QueryType	Indicates the bundle key of the label that is displayed for the query type in the UI. If the value of QueryTypeDesc is not set, the QueryTypeDesc attribute value corresponding to the query type is considered as the bundle key that will display the query type label. For more information about the supported query types, refer to the description of the QueryType Element.
Configuration	Element	Attributes/Attribute	Indicates that the search attribute will have certain configuration for UI control and possible values of the search attribute.
ActionURL	Attribute	Attributes/Attribute/ Configuration	Indicates the action that returns the list of possible values for the search attribute. This action may either return the static list of predefined values or call the corresponding API to get them, and then return them. For example, the ActionURL can return the values pertaining to the common codes, currency, and countries.
UIControlConfiguration	Element	Attributes/Attribute/ Configuration	Indicates that the search attribute will have certain configurations for UI control.

Component	Component Type	Component Location	Description																
xtype	Attribute	Attributes/Attribute/Configuration/UIControlConfiguration	Indicates the type of the control that is displayed in the Advanced Search screen for the search attribute. Following are the possible xtypes for an attribute: <table border="1" data-bbox="727 432 1390 1098"> <thead> <tr> <th>xtype</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>xadvancedtextcontrol</td> <td>Used to display a text box in the UI.</td> </tr> <tr> <td>xadvancedradiocontrol</td> <td>Used to display radio buttons in the UI.</td> </tr> <tr> <td>xadvanceddatecontrol</td> <td>Used to display a field for entering date in the UI.</td> </tr> <tr> <td>xadvancedcomboboxcontrol</td> <td>Used to display a drop-down list in the UI.</td> </tr> <tr> <td>xadvancedtriggercontrol</td> <td>Used to display a trigger field in the UI.</td> </tr> <tr> <td>xadvattributelookup</td> <td>Used to display an attribute lookup widget in the UI.</td> </tr> <tr> <td>xadvancednumbercontrol</td> <td>Used to display a number field in the UI.</td> </tr> </tbody> </table>	xtype	Description	xadvancedtextcontrol	Used to display a text box in the UI.	xadvancedradiocontrol	Used to display radio buttons in the UI.	xadvanceddatecontrol	Used to display a field for entering date in the UI.	xadvancedcomboboxcontrol	Used to display a drop-down list in the UI.	xadvancedtriggercontrol	Used to display a trigger field in the UI.	xadvattributelookup	Used to display an attribute lookup widget in the UI.	xadvancednumbercontrol	Used to display a number field in the UI.
xtype	Description																		
xadvancedtextcontrol	Used to display a text box in the UI.																		
xadvancedradiocontrol	Used to display radio buttons in the UI.																		
xadvanceddatecontrol	Used to display a field for entering date in the UI.																		
xadvancedcomboboxcontrol	Used to display a drop-down list in the UI.																		
xadvancedtriggercontrol	Used to display a trigger field in the UI.																		
xadvattributelookup	Used to display an attribute lookup widget in the UI.																		
xadvancednumbercontrol	Used to display a number field in the UI.																		
optionsBinding	Attribute	Attributes/Attribute/Configuration/UIControlConfiguration	The optionsBinding attribute is used to specify the repeating elements of the list that constitutes one value of either combo box values or radio button values. Note: This is applicable only for a combo box and a radio button.																
displayField	Attribute	Attributes/Attribute/Configuration/UIControlConfiguration	The displayField attribute specifies the attribute whose value is used as the label. Note: This is applicable only for a combo box and a radio button.																
valueField	Attribute	Attributes/Attribute/Configuration/UIControlConfiguration	The valueField attribute is used to uniquely identify the repeating element in order to display the correct value of the selected option from either a combo box or a radio button.																

Note: Ensure that you rebuild the Enterprise ARchive (EAR) after adding a new search attribute.

For more information about building an EAR, refer to the *Selling and Fulfillment Foundation: Installation Guide*.

Adding a Search Attribute - An Example

You may want to enable users to search the child price lists of a master price list. In such a scenario, you can add the master price list ID attribute to the list of advanced price list search attributes. Because master price list IDs are string values, you can allow the query types, starts with or is, and use a text box as the input control.

To add the master price list ID as a search criteria:

1. Create the Pricelist.xml file in the `<INSTALL_DIR>/extensions/sbc/webpages/sbc/metadata/advancedsearch` folder if the file does not exist in the folder. Ensure that the root element of the XML is `Attributes`.
2. Create an element, `Attribute`, as a child element of the root element, `Attributes`.
3. Set the value of the XML attributes, `Action`, to `ADD`.
4. Set the value of the XML attribute, `Name`, to `PricelistName` in the element, `Attribute`. This is because the XML attribute that is set with the master price list ID in the input XML is `PricelistName`.
5. Set the value of the XML attribute, `XPath`, to `PricelistHeader\InheritFromPricelistHeader` in the element, `Attribute`. This is because the XML path of the `PricelistName` attribute for the master price list ID is `PricelistHeader\InheritFromPricelistHeader`.
6. To define the user interface (UI) label in the search attribute drop-down list, set the XML attribute, `DisplayName`, to any key. Ensure that you add a bundle entry as the key for this attribute.
7. To use a text field for the attribute, set the XML attribute, `xtype`, to `xadvancedtextcontrol`.
8. Create an element, `Configuration`, as the child element of the element, `Attribute`.
9. Create an element, `UIControlConfiguration`, as the child element of the element, `Configuration`.
10. To use only a few string query types, create the element, `StringQueryTypes`, as the child element of the element, `Attribute`.

11. To support the query types, is and starts with, perform the following steps:
 - a. Add the QueryType elements with the value of the QueryType XML attribute as EQ and FLIKE.
 - b. To customize the query type description, set the value of the QueryTypeDesc XML attribute to the correct bundle key.

The search attribute is added.

Following is the newly added attribute element for the master price list ID field:

```
<Attribute Action="ADD" Name="PricelistName"
DisplayName="b_Price_List_Name"
XPath="PricelistHeader/InheritFromPricelistHeader">
<Configuration>
<UIControlConfiguration xtype="xadvancedtextcontrol"/>
</Configuration>
<StringQueryTypes>
<QueryType QueryType="EQ" QueryTypeDesc="is"/>
<QueryType QueryType="FLIKE"/>
</StringQueryTypes>
</Attribute>
```

Modifying a Search Attribute - An Example

Users may want to allow only the is query type and starts with query type for the Item ID field in the Advanced Search Item screen in order to improve the performance of the search task. In such a scenario, you can modify the existing query types for the search attribute.

To modify the XML attribute, QueryType:

1. Create the Item.xml file if it does not exist in the `<INSTALL_DIR>/extensions/sbc/webpages/sbc/metadata/advancedsearch` folder.
2. Create an element, Attribute, as a child element of the root element, Attributes.
3. Set the value of the attribute, Name, to ItemID.

Following is the XML attribute that has to be modified:

```
<Attribute Name="ItemID" DisplayName="b_ItemSearch_Item_ID" XPath="Item">
<Configuration>
<UIControlConfiguration xtype="xadvancedtextcontrol"/>
</Configuration>
</Attribute>
```

4. Set the value of the attribute, Action, to ADD.
5. Add the QueryType elements with the value of the QueryType XML attribute as EQ and FLIKE.

6. To customize the query type description, set the value of the QueryTypeDesc XML attribute to the correct bundle key.

The XML attribute is modified.

Following is the attribute element that is modified:

```
<Attribute Name="ItemID" DisplayName="b_ItemSearch_Item_ID" XPath="Item"
Action="ADD">
<Configuration>
<UIControlConfiguration xtype="xadvancedtextcontrol"/>
</Configuration>
<StringQueryTypes>
<QueryType QueryType="EQ" QueryTypeDesc="is"/>
<QueryType QueryType="FLIKE" QueryTypeDesc="starts_with"/>
</StringQueryTypes>
</Attribute>
```

Remove a Search Attribute

You may want users to exclude a search attribute when performing an advanced search. In such a scenario, you can remove the corresponding attribute from the search criteria.

To remove a search attribute:

1. Create a corresponding Advanced Search metadata file if it does not already exist in the `<INSTALL_DIR>/extensions/sbc/webpages/sbc/metadata/advancedsearch` folder.
2. Create an element, Attribute, as a child element of the root element, Attributes.
3. Set the value of the attribute, Name, to the name of the search attribute you want to remove.
4. Set the value of the corresponding XML attribute, Action, to DELETE.

The search attribute is removed.

Note: Ensure that you rebuild the Enterprise ARchive (EAR) after removing a search attribute.

For more information about building the EAR, refer to the *Selling and Fulfillment Foundation: Installation Guide*.

Removing a Search Attribute - An Example

You may, for example, want users to exclude the attribute, target, as a search criteria from the Advanced Pricing Rule Search screen. In such a scenario, you can remove the corresponding XML attribute from the `PricingRule.xml` file.

To remove the attribute, target:

1. Create the `PricingRule.xml` file if it does not exist in the `<INSTALL_DIR>/extensions/sbc/webpages/sbc/metadata/advancedsearch` folder.
2. Create an element, Attribute, as a child element of the root element, Attributes.
3. Set the value of the attribute, Name, to `TargetAttributeValue`.

4. Set the value of the corresponding XML attribute, Action, to DELETE.

The search attribute is removed.

Following is the attribute element that is removed:

```
<Attribute Name="TargetAttributeValue" Action="DELETE">  
</Attribute>
```

Customize Related Tasks for Business Center

Business Center™ enables users to efficiently set up and maintain their organization’s catalog, and define the peripheral components of items, such as assets, classifications, and so on, after their organization has downloaded all the items’ details into the Sterling database. This can be done by performing tasks and related tasks using Business Center’s task-based user interface. The **Related Tasks** panel contains related task categories and related tasks under a given category pertaining to a screen. The **Related Tasks** panel can be customized by removing a related task or a related task category, adding a related task or a related task category, and redirecting a related task to a new screen in the corresponding Related Task metadata file. All the customizations pertaining to the related tasks must be performed by defining the corresponding Related Task metadata file located at the

<INSTALL_DIR>/extensions/sbc/webpages/sbc/metadata/relatedtasks folder.

You can locate a Related Task metadata file in the following ways:

- For certain screens, refer to the following table for the corresponding Related Task metadata file:

Screen Name	Related Task Metadata File Name
Customer Entitlement	buyingentitledetailscreen.xml
Seller Entitlement	sellingentitledetailscreen.xml
Customer Entitlement Search	buyingentitlementsearch.xml
Seller Entitlement Search	sellingentitlementsearch.xml
Delivery Service Associations	sbcItemDetailsManageServices.xml
Return Pickup Service Associations	sbcItemDetailsManageServices.xml
Coupon Search	couponSearch.xml
Manual Pricing Rule Search	manualPricingRuleSearch.xml
Price List Search	seapricelistsearch.xml
Pricing Rule Search	pricingRuleSearch.xml

- For the remaining screens, the corresponding Related Task metadata file is named as <sciId>.xml, where *sciId* is the unique identifier of the screen. For example, *itemSearch* is the *sciId* property of the Item Search screen and the Related Task metadata file for this screen is *itemSearch.xml*. For more information about the *sciId* property, refer to the *Selling and Fulfillment Foundation: Customizing the Web UI Framework* Guide.

A Related Task metadata file contains the following elements:

- **Category:** You can add, delete, or modify a related task category by defining the *Action* attribute in the *Category* element in the Related Task metadata file.

You can perform the following tasks pertaining to a related task category in the corresponding Related Task metadata file:

- Add a new related task category in the corresponding Related Task metadata file: The new related task category is added at the end of the list in the corresponding **Related Tasks** panel.
 - Delete a related task category from the corresponding Related Tasks XML: All the related tasks under a given category are deleted from the corresponding **Related Tasks** panel.
 - Modify a related task category in the Related Task metadata file corresponding to a related task category: You can add a new related task, remove an existing related task, and modify an existing related task. Modifying a related task includes modifying only the action that occurs when a user clicks a related task under a given category.
 - IncludeCategory: If a Related Task metadata file contains this element, it indicates that this XML consists of multiple Related Task metadata files corresponding to various categories. The path to these Related Task metadata files are defined in the corresponding File attribute of the IncludeCategory element. You can add or delete the IncludeCategory element, which in turn adds or deletes a category if it is present in the File attribute. However, you cannot set the value of the Action attribute to MODIFY in the IncludeCategory element.
 - Task: You can add, delete, or modify a related task by defining the Action attribute in the Task element. However, you need to first set the Action attribute to Modify in the corresponding Category element.
- Note:** You cannot customize the Dynamic Tasks.

Add a Related Task Category

You can add a related task category to the list of existing categories in a **Related Task** panel. For example, if you have created a new category with new tasks, you can add a new related task category containing new related tasks in the **Related Tasks** panel.

To add a related task category:

1. Create a corresponding Related Task metadata file if it does not already exist in the `<INSTALL_DIR>/extensions/sbc/webpages/sbc/metadata/relatedtasks` folder.
2. Create an element, Category, as a child element of the root element, Categories.
3. Set the value of the XML attribute, Action, to ADD.
4. Set the value of the XML attribute, CategoryID, as required. This value acts as a unique identifier for the related task category.
5. Set the value of the XML attribute, CategoryName, as required. This value indicates the bundle key of the related task category that is displayed in the corresponding **Related Tasks** panel.

The related task category is added at the end of the list of categories in the corresponding **Related Tasks** panel.

Note: Ensure that you rebuild the Enterprise ARchive (EAR) after adding a new related task category or related task.

For more information about building an EAR, refer to the *Selling and Fulfillment Foundation: Installation Guide*.

Modify a Related Task Category

You may want to modify a related task category in order to add, delete, or modify a task under it.

To modify a related task category:

1. Create a corresponding Related Task metadata file if it does not already exist in the `<INSTALL_DIR>/extensions/sbc/webpages/sbc/metadata/relatedtasks` folder.
2. Create an element, `Category`, as a child element of the root element, `Categories`.
3. Set the value of the corresponding XML attribute, `Action`, to `MODIFY`.
4. Set the value of the XML attribute, `CategoryID`, as required. This value acts as a unique identifier for the related task category.
5. Perform any or all of the following tasks:
 - [Add a Related Task](#)
 - [Modify a Related Task](#)
 - [Delete a Related Task](#)

Add a Related Task

If you have created a new screen, you can add a new related task in the **Related Tasks** panel, which, when clicked, displays the corresponding screen.

1. Create an element, `Tasks`, as a child element of the root element, `Category`.
2. Create an element `Task`, as a child element of the element, `Tasks`.
3. Set the value of the corresponding XML attribute, `Action`, to `ADD`.
4. Set the value of the XML attribute, `TaskID`, as required. This value acts as a unique identifier and resource identifier for the related task.
5. Set the value of the XML attribute, `TaskName`, as required. This value indicates the bundle key of the related task that is displayed in the corresponding **Related Tasks** panel.
6. Set the value of the XML attribute, `PermissionID`, as required. This indicates the resource that controls the permission to this task.
7. Set the value of the XML attribute, `ActionID`, as required. This value indicates the action that will return the corresponding screen in which the related task can be performed.

The related task is added to the related task category in the corresponding **Related Tasks** panel.

Modify a Related Task

You can modify the `ActionID` attribute of the `Task` element so that when the corresponding related task is clicked in the **Related Tasks** panel, a new screen linked to the modified `ActionID` attribute is displayed. For example, if you have created a new screen, you can modify the existing related task in the **Related Tasks** panel, which, when clicked, displays the corresponding screen.

1. Create an element, `Tasks`, as a child element of the root element, `Category`.
2. Create an element, `Task`, as a child element of the element, `Tasks`.

3. Set the value of the XML attribute, TaskID, as required. This value acts as a unique identifier and resource identifier for the related task.
4. Define the value of the corresponding XML attribute, ActionID, as required. This value indicates the action that will return the corresponding screen in which the related task can be performed.

The related task is modified. When a user clicks the related task in the corresponding **Related Tasks** panel, the screen to which the value in the ActionID attribute refers, is displayed.

Delete a Related Task

1. Create an element, Tasks, as a child element of the root element, Category.
2. Create an element, Task, as a child element of the root element, Tasks.
3. Set the value of the XML attribute, TaskID, as required. This value acts as a unique identifier and resource identifier for the related task.
4. Set the value of the corresponding XML attribute, Action, to DELETE.

The related task is deleted from the related task category in the corresponding **Related Tasks** panel.

Delete a Related Task Category

You may not want users to perform related tasks under a given category when viewing a screen. In such a scenario, you can delete the corresponding related task category from the Related Tasks panel.

To delete a related task category:

1. Create a corresponding Related Task metadata file if it does not already exist in the `<INSTALL_DIR>/extensions/sbc/webpages/sbc/metadata/relatedtasks` folder.
2. Create an element, Category, as a child element of the root element, Categories.
3. Set the value of the XML attribute, CategoryID, as required. This value acts as a unique identifier for the related task category.
4. Set the value of the corresponding XML attribute, Action, to DELETE.

The related task category along with the related tasks under that category are deleted from the **Related Tasks** panel.

Note: Ensure that you rebuild the Enterprise ARchive (EAR) after deleting a related task category or related task.

For more information about building an EAR, refer to the *Selling and Fulfillment Foundation: Installation Guide*.

Customize the Conditions in an Approval Rule for Business Center

Business Center™ enables enterprise users to create approval rules that will be used to set and enforce discount policies in the Sterling Field Sales application. Enterprise users can also create the conditions under which an approval rule will become applicable. You, as an implementer, can customize the conditions displayed in the Approval Rule Summary screen. A condition pertains to a domain that is displayed in a tree hierarchy containing various entity attributes. Depending on the entity attribute selected, fields are displayed in the dynamic row of the **Conditions** panel.

You can customize the **Conditions** panel in the Approval Rule Summary screen by configuring the `ApprovalRuleConfig.xml` file in the `<INSTALL_DIR>/extensions/sbc/webpages/sbc/metadata/approval` folder. You must create the `ApprovalRuleConfig.xml` file to specify the entity attributes, along with details such as the controls to be displayed in the user interface (UI), query types to be used, and so on.

Add a New Entity Attribute to a Domain

You can add a new entity attribute to a domain that is displayed in the tree hierarchy from which an enterprise user can select the newly added entity attribute to create a condition. However, you cannot add a new dynamic entity attribute like the Item attribute to a domain. This is because the dynamic entity attributes are read dynamically from the database, and depending on the entity attribute the XML attributes such as query types, x paths and so on are displayed. You can add new entity attributes to domains, Order and OrderLine.

To add a new entity attribute to a domain:

1. Create `ApprovalRuleConfig.xml` file if it does not already exist in the `<INSTALL_DIR>/extensions/sbc/webpages/sbc/metadata/approval` folder.
2. Create an element, `Attribute`, as a child element of the element, `Attributes`. Ensure that you follow the structure of the following sample XML file when creating elements.

3. Set the value of the XML attribute, Action, to ADD.

```

<ApprovalConfig>
  <Domain Name="Order">
    <Attributes>
      <Attribute Action="ADD" Name="OrderDate"
        DisplayName="b_Effective_Start_Date"
        XPath="Order" DefaultQueryType="DATEGREATER">
        <QueryTypes>
          <QueryType QueryType="DATEGREATER" QueryTypeDesc="b_date_greater">
            <GreexFunction Name="dateGreater" />
            <Configuration>
              <UIControlConfiguration xtype="xapprovaldatecontrol" />
            </Configuration>
          </QueryType>
        </QueryTypes>
      </Attribute>
    </Attributes>
  </Domain>
</ApprovalConfig>

```

4. Define the values of the other XML attributes.

Refer to the following table for more information about the other XML attributes:

Component	Component Type	Component Location	Description
Name	Attribute	ApprovalConfig/Domain/Attributes/Attribute	Indicates the XML name of the entity attribute.
DisplayName	Attribute	ApprovalConfig/Domain/Attributes/Attribute	Indicates the bundle key of a label that is displayed in a drop-down list for entity attributes. If the label is not mentioned, the Name attribute is considered as the bundle key to display the label.
XPath	Attribute	ApprovalConfig/Domain/Attributes/Attribute	Indicates the XML path of the element to which the entity attribute belongs. This XML path is also used by the corresponding Greex Rule XML to locate the entity attribute when the approval rule is being evaluated.
DefaultQueryTypes	Attribute	ApprovalConfig/Domain/Attributes/Attribute	Indicates the default query type for the entity attribute.
compareAny	Attribute	ApprovalConfig/Domain/Attributes/Attribute	Indicates that the Greex engine compares any of the values of the corresponding XPath.
QueryTypes	Element	ApprovalConfig/Domain/Attributes/Attribute/QueryTypes	Indicates the parent element of the supported query types for the entity attribute.

Component	Component Type	Component Location	Description
QueryType	Element	ApprovalConfig/Domain/Attributes/Attribute/QueryTypes/QueryType	Indicates the query types that are supported for the entity attribute. Ensure that you set the value of the corresponding GreexFunction child element.
QueryType	Attribute	ApprovalConfig/Domain/Attributes/Attribute/QueryTypes/QueryType	Indicates the query type.
QueryTypeDesc	Attribute	ApprovalConfig/Domain/Attributes/Attribute/QueryTypes/QueryType	Indicates the bundle key of the label that is displayed for the query type in the UI.
GreexFunction	Element	ApprovalConfig/Domain/Attributes/Attribute/QueryTypes/QueryType/GreexFunction	Indicates the Greex Function associated to a query type.
Name	Attribute	ApprovalConfig/Domain/Attributes/Attribute/QueryTypes/QueryType/GreexFunction	Indicates the Greex Function associated to a query type so that the condition using this query type can be transformed in a Greex Rule XML.
Configuration	Element	ApprovalConfig/Domain/Attributes/Attribute	Indicates that the entity attribute will have certain configurations for UI control and possible values for the entity attribute.
ActionURL	Attribute	ApprovalConfig/Domain/Attributes/Attribute/QueryTypes/QueryType/Configuration	Indicates the action that returns the list of possible values for the entity attribute. This action may either return the static list of predefined values or call the corresponding API to get them, and then return them. For example, ActionURL can return the values pertaining to the common codes, currency, and countries.
UIControlConfiguration	Element	ApprovalConfig/Domain/Attributes/Attribute/QueryTypes/QueryType/Configuration/UIControlConfiguration	Indicates that the entity attribute will have certain configurations for UI control.

Component	Component Type	Component Location	Description																
xtype	Attribute	ApprovalConfig/Domain/Attributes/Attribute/QueryTypes/QueryType/Configuration/UIControlConfiguration	<p>Indicates the type of control that is displayed in the Conditions panel in the Approval Rule Summary screen for the entity attribute. Following are the possible xtypes for an attribute:</p> <table border="1"> <thead> <tr> <th>xtype</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>xapprovaltextcontrol</td> <td>Used to display a text box in the UI.</td> </tr> <tr> <td>xapprovalradiocontrol</td> <td>Used to display radio buttons in the UI.</td> </tr> <tr> <td>xapprovaldatecontrol</td> <td>Used to display a field for entering date in the UI.</td> </tr> <tr> <td>xapprovalcombocontrol</td> <td>Used to display a drop-down list in the UI.</td> </tr> <tr> <td>xapprovaltriggercontrol</td> <td>Used to display a trigger field in the UI.</td> </tr> <tr> <td>xapprovalattributelookup</td> <td>Used to display the Item attribute lookup widget in the UI.</td> </tr> <tr> <td>xapprovalnumbercontrol</td> <td>Used to display a number field in the UI.</td> </tr> </tbody> </table>	xtype	Description	xapprovaltextcontrol	Used to display a text box in the UI.	xapprovalradiocontrol	Used to display radio buttons in the UI.	xapprovaldatecontrol	Used to display a field for entering date in the UI.	xapprovalcombocontrol	Used to display a drop-down list in the UI.	xapprovaltriggercontrol	Used to display a trigger field in the UI.	xapprovalattributelookup	Used to display the Item attribute lookup widget in the UI.	xapprovalnumbercontrol	Used to display a number field in the UI.
xtype	Description																		
xapprovaltextcontrol	Used to display a text box in the UI.																		
xapprovalradiocontrol	Used to display radio buttons in the UI.																		
xapprovaldatecontrol	Used to display a field for entering date in the UI.																		
xapprovalcombocontrol	Used to display a drop-down list in the UI.																		
xapprovaltriggercontrol	Used to display a trigger field in the UI.																		
xapprovalattributelookup	Used to display the Item attribute lookup widget in the UI.																		
xapprovalnumbercontrol	Used to display a number field in the UI.																		
decimalPrecision	Attribute	ApprovalConfig/Domain/Attributes/Attribute/QueryTypes/QueryType/Configuration/UIControlConfiguration	Indicates the number of digits allowed after a decimal separator of a decimal number in a number field.																
allowNegative	Attribute	ApprovalConfig/Domain/Attributes/Attribute/QueryTypes/QueryType/Configuration/UIControlConfiguration	Indicates if negative values are allowed as an input in a number field.																
maxValue	Attribute	ApprovalConfig/Domain/Attributes/Attribute/QueryTypes/QueryType/Configuration/UIControlConfiguration	Indicates the maximum value that is allowed as an input in a number field.																
optionsBinding	Attribute	ApprovalConfig/Domain/Attributes/Attribute/QueryTypes/QueryType/Configuration/UIControlConfiguration	<p>The optionsBinding attribute is used to specify the repeating elements of the list that constitutes one value of either combo box values or radio button values.</p> <p>Note: This is applicable only for a combo box and a radio button.</p>																

Component	Component Type	Component Location	Description
displayField	Attribute	ApprovalConfig/Domain/Attributes/Attribute/QueryTypes/QueryType/Configuration/UIControlConfiguration	The displayField attribute specifies the attribute whose value is used as the label. Note: This is applicable only for a combo box and a radio button.
valueField	Attribute	ApprovalConfig/Domain/Attributes/Attribute/QueryTypes/QueryType/Configuration/UIControlConfiguration	The valueField attribute is used to uniquely identify the repeating element in order to display the correct value of the selected option from either a combo box or a radio button.

Note: Ensure that you rebuild the Enterprise ARchive (EAR) after adding an entity attribute.

For more information about building an EAR, refer to the *Selling and Fulfillment Foundation: Installation Guide*.

Modify an Entity Attribute in a Domain

You can modify an entity attribute in a domain to add a new query type or delete an existing query type. The entity attribute is displayed in the tree hierarchy from which an enterprise user can select the modified entity attribute to create a condition. However, you cannot modify an existing dynamic entity attribute such as ItemAttribute in a domain.

To modify an entity attribute in a domain:

1. Create the ApprovalRuleConfig.xml file if it does not already exist in the <INSTALL_DIR>/extensions/sbc/webpages/sbc/metadata/approval folder.
2. Create an element, Attribute, as a child element of the element, Attributes.
3. Set the value of the XML attribute, Action, to MODIFY.
4. Modify the values of the other XML attributes.

Add a New Query Type to an Entity Attribute

To add a new query type to an entity attribute:

1. Create the ApprovalRuleConfig.xml file if it does not already exist in the <INSTALL_DIR>/extensions/sbc/webpages/sbc/metadata/approval folder.
2. Create an element, Attribute, as a child element of the element, Attributes.
3. Set the value of the XML attribute, QueryType, to ADD.
4. Modify the values of the other XML attributes.

Delete an Existing Query Type from an Entity Attribute

If you delete a query type and the corresponding entity attribute is used by a condition in an approval rule, none of the conditions will be displayed in the Approval Summary screen corresponding to the approval rule.

To delete an existing query type from an entity attribute:

1. Create the `ApprovalRuleConfig.xml` file if it does not already exist in the `<INSTALL_DIR>/extensions/sbc/webpages/sbc/metadata/Approval` folder.
2. Create an element, `Attribute`, as a child element of the element, `Attributes`.
3. Set the value of the XML attribute, `QueryType`, to `DELETE`.

Note: Ensure that you rebuild an Enterprise ARchive (EAR) after modifying an entity attribute.

For more information about building the EAR, refer to the *Selling and Fulfillment Foundation: Installation Guide*.

Delete an Entity Attribute from a Domain

You can delete an entity attribute from a domain. If you delete an entity attribute and the entity attribute is used by a condition in an approval rule, none of the conditions will be displayed in the Approval Summary screen corresponding to the approval rule.

To delete an entity attribute from a domain:

1. Create the `ApprovalRuleConfig.xml` file if it does not already exist in the `<INSTALL_DIR>/extensions/sbc/webpages/sbc/metadata/approval` folder.
2. Create an element, `Attribute`, as a child element of the element, `Attributes`.
3. Set the value of the XML attribute, `Action`, to `DELETE`.

The entity attribute is deleted from the domain.

Note: Ensure that you rebuild an Enterprise ARchive (EAR) after deleting an entity attribute.

For more information about building the EAR, refer to the *Selling and Fulfillment Foundation: Installation Guide*.

Deploying Web UI Framework Customization

After you perform the customizations in the Business Center™ application, you must place all the extensions in the <INSTALL_DIR>/extensions/sbc/webpages directory. You must place all the Java Script files in the <INSTALL_DIR>/extensions/sbc/webpages/extn directory.

For more information about deploying the Web UI Framework extensions, refer to the *Selling and Fulfillment Foundation: Customizing the Web UI Framework Guide*.

A

advanced search metadata file name
approval rule search 10
coupon search 10
entitlement search 10
item search 10
manual pricing rule search 10
price list search 10
pricing rule search 10
service option search 10
service search 10

B

basic and advanced searches 10
Business Center customization
an overview 5

C

Customization Environment
setting up 6
customize advance search criteria
add or modify a search attribute 10
an example 15
defining XML attributes 11
remove a search attribute 17
an example 17
customize conditions
add a new entity attribute 23
define XML attributes 24
delete an entity attribute 28
modify an entity attribute 27
add a new query type 27
delete a query type 28
customize related tasks
add a related task category 20
delete a related task category 22
modify a related task category 21
add a related task 21

delete a related task 22
modify a related task 21

customizing Business Center
advanced search criteria 10
basic differential screen extensions 5
conditions in an approval rule 23
CSS 5
data types 5
deploying the Web UI Framework customizations 29
login screen 5
mashup extensions 5
new meny entry 5
new UI screens 5
override screen extensions 5
related tasks 19
resource bundles 5
struts extensions 5
using Web UI Framework 7

R

related task metadata file name
coupon search 19
customer entitlement 19
customer entitlement search 19
delivery service association 19
manual pricing rule search 19
price list search 19
pricing rule search 19
return pickup association 19
seller entitlement 19
seller entitlement search 19