

**Oracle® Agile Product Lifecycle Management for Process**  
Navigation Configuration Guide

Extensibility Pack 2.7  
**E37244-01**

September 2012

**ORACLE®**

# Copyrights and Trademarks

Agile Product Lifecycle Management for Process

Copyright © 1995, 2012, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

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

## Contents

<b>EXTENDING NAVIGATION PANELS .....</b>	<b>4</b>
Overview .....	4
Platform Navigation .....	4
Portal Navigation .....	4
Action Navigation.....	5
Adding/Adjusting Navigation Menu Items.....	5
Portal and Platform Navigation .....	5
Action Navigation Menu Items .....	8
Action Navigation Icons .....	12
Icon Examples .....	12
<b>CONTROLLING VISIBILITY AND SECURITY .....</b>	<b>14</b>
Security Profile (“SecurityProfileRef” Attribute).....	14
<SecurityProfile> Attributes.....	14
Security Profile Mode .....	15
<SecurityProfile> Evaluators .....	16
Security Profile Examples.....	17
Visibility (“Visible” Attribute).....	18
Visibility Examples.....	19
<b>AVAILABLE SERVER SIDE VARIABLES .....</b>	<b>19</b>
<b>REFERENCING THE ACTIVE OBJECT .....</b>	<b>20</b>
<b>SORT ORDER .....</b>	<b>20</b>
SiteMap-DisplayOrder.xml <MenuItem> Attributes.....	21
Sort Order Examples .....	21
<b>TRANSLATION CACHES .....</b>	<b>21</b>
Add a New Translation.....	22
Adjusting an Existing Translation .....	23

## Extending Navigation Panels

### Overview

You can extend all navigation panels throughout the application suite. There are three primary navigation areas:

### Platform Navigation

Platform Navigation is the navigation menu available in the top right of the browser window inside the suite header.

This menu can be adjusted in the following ways:

1. Add Items
2. Remove Items
3. Re-arrange Items
4. Apply Visibility and Security Controls



### Portal Navigation

Portal Navigation is available on the portal homepage listing in the left navigation panel.

This menu can be adjusted in the following ways:

1. Add Items
2. Remove Items
3. Re-arrange Items
4. Apply Visibility and Security Controls



## Action Navigation

Action Navigation is available in the top left corner of all objects. This navigation also includes the quick access icons.

This menu can be adjusted in the following ways:

1. Add Menu Items
2. Remove Menu Items
3. Add Quick Access Icons
4. Remove Quick Access Icons
5. Adjust Hot Keys
6. Re-arrange Items
7. Apply Visibility and Security Controls



## Adding/Adjusting Navigation Menu Items

All navigation is defined by site map configurations inside the core config directory. Each application has its own corresponding site map configuration file. For example, SiteMap-GSM.xml controls all GSM Action Navigation Menus, SiteMap-SCRM.xml controls SCRM, etc. These files are encrypted in the release, but you can find reference copies in the following directory: /ReferenceImplementations/Navigation

With the SiteMap-Extensions.xml file, located in the \Config\Extensions directory you can override existing menu items or add new menu items that are not part of the core product.

## Portal and Platform Navigation

Menu Items are shared between these two menus. Portal and Platform Navigation uses the SiteMap-[App].xml core configuration files for each Application Menu. For example, the GSM Application Menu can be found in the SiteMap-GSM.xml file. To adjust existing items, the existing menu item nodes need to be copied from the SiteMap-[App].xml file and added to the SiteMap-Extensions.xml file using the same node location. To add new items, new menu item nodes need to be added to the SiteMap-Extensions.xml file and placed in the same node location where you want it to appear.

*All Portal and Platform navigation individual application menus are inside the following nodes:*

```
<MenuItem ID="ApplicationWideMenu" DisplayText="**Application Wide Menu" >

    <MenuItem ID="Applications" DisplayText="**Applications">

        INSERT MENU ITEM NODE OVERRIDES HERE

    </MenuItem>

</MenuItem>
```

*Example of the GSM Parent Menu Item:*

```
<MenuItem ID="GSMSideNav" DisplayText="**GSM" SecurityProfileRef="GSMUser" SkinRef="DefaultWithChildren"
TranslationRef="" Target="" NavigateUrl="{dc:GSM}" DefaultItemTextWrap="true" SortOrder="1"
```

```
ToolTip="translate:Portal/PortalGSM/IblGlobalSpecMgmt" Visible="eval:({fc:Portal.SideNav.GSM.Enabled})">
```

### *Adjusting an Existing Node Example:*

To override a node and apply different attributes (like a security profile), the following steps need to be performed:

*Step 1: Open SiteMap-GSM.xml and locate the node you would like to override. In this example, it is the Action Items Node.*

```
<MenuItem ID="IblType5000" DisplayText="***Action Items" SecurityProfileRef="" SkinRef="DefaultWithoutChildren"
TranslationRef="GSMSideBar" Target="" ClientSideCommand="{ConfirmationJsStart,
document.location.href=}'${dc:GSM}/baseforms/default.aspx?ContentType=5000'${ConfirmationJsEnd,;}" />
```

*Step 2: Copy the node, including all parent nodes*

**PARENT NODE –** [`<MenuItem ID="GSMSideNav" DisplayText="***GSM" SecurityProfileRef="GSMUser"
SkinRef="DefaultWithChildren" TranslationRef="" Target="" NavigateUrl="{dc:GSM}" DefaultItemTextWrap="true" SortOrder="1"
ToolTip="translate:Portal/PortalGSM/IblGlobalSpecMgmt" Visible="eval:({fc:Portal.SideNav.GSM.Enabled})"> ]`]

**NODE TO ADJUST –** [`<MenuItem ID="IblType5000" DisplayText="***Action Items" SecurityProfileRef=""
SkinRef="DefaultWithoutChildren" TranslationRef="GSMSideBar" Target="" ClientSideCommand="{ConfirmationJsStart,
document.location.href=}'${dc:GSM}/baseforms/default.aspx?ContentType=5000'${ConfirmationJsEnd,;}" /> ]`]

**PARENT NODE CLOSED –** [`</MenuItem>`]

*Step 3: Paste the node block into SiteMap-Extensions.xml and adjust the attributes of the node you would like to override. Add configChildKey="ID" to the parent menu items added.*

```
<MenuItem ID="ApplicationWideMenu" DisplayText="***Application Wide Menu" >
```

```
<MenuItem ID="Applications" DisplayText="***Applications">
```

```
<MenuItem ID="GSMSideNav" DisplayText="***GSM" SecurityProfileRef="GSMUser"
SkinRef="DefaultWithChildren" TranslationRef="" Target="" NavigateUrl="{dc:GSM}"
DefaultItemTextWrap="true" SortOrder="1"
ToolTip="translate:Portal/PortalGSM/IblGlobalSpecMgmt"
Visible="eval:({fc:Portal.SideNav.GSM.Enabled})" configChildKey="ID" >
```

```
<MenuItem ID="IblType5000" DisplayText="***Action Items"
SecurityProfileRef="NEWSECURITYPROFILE" SkinRef="DefaultWithoutChildren"
TranslationRef="GSMSideBar" Target="" ClientSideCommand="{ConfirmationJsStart,
document.location.href=}'${dc:GSM}/baseforms/default.aspx?ContentType=5000'${Confir
mationJsEnd,;}" />
```

```
</MenuItem>
```

```
</MenuItem>
```

```
</MenuItem>
```

### Adding a New Node Example:

To add an additional node, perform the following steps:

*Step 1: Open SiteMap-GSM.xml and locate the parent nodes where you want to add your new item.*

*Step 2: Copy all parent nodes where the new item will reside.*

*Step 3: Paste the node block into SiteMap-Extensions.xml and add the new menu item node.*

*Step 4: Adjust the Sort Order of the node block using the SiteMap-DisplayOrder.xml file. See Section 6 Sort Order for more information. Note: If you don't see the node block you are adjusting inside SiteMap-DisplayOrder.xml, you will need to copy the entire block and add sort order attributes.*

### <MenuItem> Attributes

Attribute	Description	Comment
ID	Specify ID of the menu item.	Override the settings of existing menu by given the existing menu ID. Add a new menu item by choosing a new ID.
configChildKey	Specify identify key of child node.	Always set to ID
DefaultItemTextWrap	Specify whether text should be allowed to wrap	Values: true/false
DisplayText	Specify text note for the menu item.	This will be used as the caption for the menu item if the node has no translation ID.
NavigateUrl	Specify the link URL of the menu item.	
SecurityProfileRef	Specify Security Profile which be configured in MenuConfig.	Can append more than one security profile. Split by ","
SkinRef	Used by the system to style the menu item	
SortOrder	Specify an order number.	
Sortable	Specify if the child nodes is sortable	Values: true/false
Target	Specify the link target.	
ToolTip	Specify the translation cache to use for the navigation item tooltip	Example: translate:Portal/PortalGSM/IblGlobalSpecMgmt

Attribute	Description	Comment
Visible	Specify variables to evaluate the visibility of the menu item.	

### Nested Panels

Nested panels are used and can be added (maximum 3 levels). An example follows:

Display	Menu Nodes
A	<menu item ID="A">
B1	<menu item ID="B1">
C1	<menu item ID="C1"/>
C2	<menu item ID="C2"/>
B2	</menu item>
C1	<menu item ID="B2">
	<menu item ID="C1"/>
	</menu item>
	</menu item>

### Action Navigation Menu Items

Action Navigation uses the SiteMap-[App].xml core configuration files for the menus in each application. For example, the GSM Action Menu can be found in the SiteMap-GSM.xml file. To add or adjust existing items, the existing menu item nodes need to be matched inside the SiteMap-Extensions.xml file.

Action Menus are in stand-alone nodes. For example, the following nodes are used for GSM Action Navigation:

Node ID	Where Used
GSMNavSpec	GSM Specifications
GSMNavActionItems	Action Items
GSMNavEQT	Specification Search
GSMNavTestingProtocol	Testing Protocols
GSMNavTSA	Temporary Signature Authority



Node ID	Where Used
GSMNavLIO	LIO Profiles
GSMNavSmartIssueEQT	Smart Issue Search
GSMNavSmartIssue	Smart Issue Request
GSMNavFicTermEQT	Component Catalog Search
GSMNavFicTerm	Component Catalog Term
GSMNavGlobalSuccessionEQT	Global Succession Search
GSMNavSignatureDoc	GSM Signature Document

Example of the GSM > New > Material Menu Item:

```
<MenuItem ID="IbTypeNav1004" CssClass="hasIcon cssXlco1004-S" DisplayText="***Material Specifications"
SecurityProfileRef="NewSpecCreator1004,NewSpecCreator_Action_1004" Visible="{fc:GSM.SideNav.MaterialSpec.Enabled}"
NavigateUrl="{dc:GSM}/baseforms/createNew.aspx?ContentType=1004" >
```

### <MenuItem> Attributes

Attribute	Description	Required	Comment
ID	Specify ID of the menu item.	No	Override the settings of existing menu by given the existing menu ID. Add a new menu item by choosing a new ID.
Action	Specify variables to evaluate the action enable status of the menu item.	No	
AutoPostBackOnSelect	Specify if this is a auto post back menu item.	No	Values: true/false
ClientSideCommand	Specify JavaScript command to be executed when clicking on the menu item.	No	
configChildKey	Specify identify key of child node.	No	Always set to ID

Attribute	Description	Required	Comment
CssClass	Specify css class of the menu item	No	Add "hasIcon" to CssClass attribute to specify whether an icon appears to the left of the menu item inside the panel. See menu item example above.
DisplayText	Specify text note for the menu item.	No	This will be used as caption for the menu item if the node has no translation ID.
HotKey	Specify the hot key for the menu item.	No	The following words/characters can be used: Ctrl, Alt, Shift, 0-9, a-z.
InferredVisibility	Specify the mode of inferred visibility. Only for menu split and menu item which has child.	No	<p>For split, Value should be "group".</p> <p>This will evaluate the visibility of menu items between two splits. And use that result to decide the menu-split be visible or not.</p> <p>For a menu item that has a child, Value should be "child".</p> <p>This will evaluate visibility of the child items. And use that result to decide the current node be visible or not.</p>
IsInEditMode	Specify a variable to let the menu know if current object is in edit mode.	No	This property only can be applied to root node.
MenuType	Specify type of the menu item	No	<p>Default is a normal menu item.</p> <p>The flowing types(in low case) can be used:</p> <p>split: specify a split</p> <p>icon: specify a tool button</p> <p>icon-split: specify a tool bar split</p> <p>title: specify a text item</p>
NavigateUrl	Specify the link URL of the menu item.	No	

Attribute	Description	Required	Comment
SecurityProfileRef	Specify Security Profile to be configured in MenuConfig.	No	Can append more than one security profile. Split by “,”
Target	Specify the link target.	No	
Visible	Specify variables to evaluate the visibility of the menu item.	No	Can add more than one by using &&

**<MenuItemReference>**

When you want to reuse an existing Menu Item Node, you can use the MenuItemReference node. Use this to define a reference point to an existing menu item.

Attribute	Description	Required	Comment
path	Specify path point to an existing menu item	Yes	Example: MenuID1/ MenuID2

**<MenuItemDynamic>**

Define a placeholder for a list of menu items that would be dynamically created by code.

Attribute	Description	Required	Comment
Factory	Specify the path of the factory class which implemented IYNavMenuDynamicItemsFactory	Yes	

## Nested Panels

Nested panels are used and can be added (maximum 3 levels). An example follows:

Display	Menu Nodes
A	<menu item ID="A">
B1	<menu item ID="B1">
C1	<menu item ID="C1"/>
C2	<menu item ID="C2"/>
B2	</menu item>
C1	<menu item ID="B2">
	<menu item ID="C1"/>
	</menu item>
	</menu item>

## Action Navigation Icons

Action Navigation can display quick access icons to the right of the action menu arrow. Menu Items with the MenuType of icon should be added to the specific menu nodes where you would like the icon to appear.

Example of GSM Edit Icon:

```
<MenuItem MenuType="icon" ID="lblEdit" DisplayText="**Edit" SecurityProfileRef="" Visible="{ShowEditButton}"
ClientSideCommand="execCmd('Edit');" />
```

Note: The translation will be used for the tooltip that appears when the user hovers over the icon.

## Icon Examples

### Example 1: Nav A will appear as an icon

```
<MenuItem ID="GSMNavSpec" configChildKey="ID">
  <MenuItem ID="NavA" MenuType="icon" DisplayText="NavA" ClientSideCommand="alert('NavA');" />
</MenuItem>
```

We can add a style to the icons.css file to specify the icon image. Other navigation icons are in the `/*ACTION Icons*/` section.

Example:

```
#ynavMenuToolbar #ynavToolButton-NavA
{
    background-image: url(../images/navmenu/NavA.gif);
}
```

We also can add a property `CssClass` to specify a class to specify the icon image.

```
<MenuItem ID="GSMNavSpec" configChildKey="ID">
    <MenuItem ID="NavA" MenuType="icon" CssClass="cssxIco2147-S" DisplayText="NavA"
    ClientSideCommand="alert('NavA');" />
</MenuItem>
```

In this example, the css class `"cssxIco2147-S"` looks like the following:

```
.cssxIco2147-S
{
    background-image: url(../images/icons/objectTypes/2147-S.png);
    background-repeat: no-repeat;
}
```

**Example 2: Item History should appear as an icon on all specification types that support it**

```
<MenuItem ID="GSMNavSpec" configChildKey="ID">
    <MenuItem ID="IbItemHistory" MenuType="icon" DisplayText="**IbItemHistory"
    Visible="${ShowItemHistoryButton}" ClientSideCommand="ShowItemHistory();" />
</MenuItem>
```

## Controlling Visibility and Security

There are two ways to control the visibility of a menu item: “SecurityProfileRef” and “Visible” attribute. Most of them can determine the visibility of a menu item. But there is still a little difference from a business perspective. “SecurityProfileRef” is used when you are evaluating session specific attributes, like user role, user group, spec permissions etc. The “Visible” attribute is used when you are evaluating static attributes, like configurations, spec types, spec mode etc.

### Security Profile (“SecurityProfileRef” Attribute)

For a menu item, the access rules should be defined. The Security Profile is a group of rules to define application, role or group based permissions. Security profiles control access to functionality within the application. For example, only users with the role of [Spec\_Creator\_1004] can create blank material specifications, so only those users can see the New > Material Specification > Blank menu item.

Existing Security Profiles are inside the Core MenuConfig.xml config. This file is encrypted in the release, but you can find a reference copy in the following directory: /ReferenceImplementations/Navigation

You can add additional Security Profiles inside MenuConfig-Extensions.xml. Security Profiles should be added within the following node:

```
<Providers configChildKey="ID">
    <Provider ID="SecurityProvider">
        <SecurityProfiles>
            INSERT SECURITY PROFILES HERE
        </SecurityProfiles>
    </Provider>
```

Example of the Material Template Creator Security Profile:

```
<SecurityProfile ID="CreateFromTemplate1004" Mode="Visibility">
    <Allow Roles="[CREATE_FROM_TEMPLATE_1004]"/>
</SecurityProfile>
```

### <SecurityProfile> Attributes

Attribute	Description	Comment
ID	Specify ID of the security profile.	This is the ID referenced on the menu item
Mode	Defines whether the security profile will just control visibility alone or if it will also perform an action.	Values: Visibility/Action

Attribute	Description	Comment
Application	Evaluates the users UGM Site Access	Application = "GSM"  Will check if the user has the site access GSM checkbox selected on their user profile.

### Security Profile Mode

This is used on the New > Material Specification menu item. The user can hover over "Material Specification" and it will open the From Blank, From Template panel. The user can also click on the label Material Specification and it will automatically create a blank Material Specification.

The Material Specification menu item uses two security profiles:

SecurityProfileRef="NewSpecCreator1004,NewSpecCreator\_Action\_1004" Multiple security profiles are separated by a comma ",".

NewSpecCreator1004 (shown below) allows users with the following roles [SPEC\_CREATOR],[SPEC\_ADMIN],[SPEC\_CREATOR\_1004],[CREATE\_FROM\_TEMPLATE\_1004] to see the menu item Material Specification.

```
<SecurityProfile ID="NewSpecCreator1004" Mode="Visibility">
```

```
<Allow Roles="[SPEC_CREATOR],[SPEC_ADMIN],[SPEC_CREATOR_1004],[CREATE_FROM_TEMPLATE_1004]"/>
```

```
</SecurityProfile>
```

NewSpecCreator\_Action\_1004 (shown below) allows only users with the following roles [SPEC\_CREATOR],[SPEC\_ADMIN],[SPEC\_CREATOR\_1004] to create a spec by clicking on the label Material Specification. So users with [CREATE\_FROM\_TEMPLATE\_1004] will see the menu item Material Specification but they will not be allowed to click on the item to create a new blank specification.

```
<SecurityProfile ID="NewSpecCreator_Action_1004" Mode="Action">
```

```
<Allow Roles="[SPEC_CREATOR],[SPEC_ADMIN],[SPEC_CREATOR_1004]"/>
```

```
</SecurityProfile>
```

**<SecurityProfile> Evaluators**

Some of the box evaluators exist that allow you to evaluate user role and user group. You can also specify a custom evaluator class to evaluate more complex rules.

Attribute	Description
<Allow Roles="[ROLE]"/>	<p>List the user roles that should be allowed. Example:</p> <pre>&lt;Allow Roles="[SPEC_CREATOR],[SPEC_ADMIN]"/&gt;</pre> <p>Roles separated by commas will be evaluated using an OR condition. If multiple Allow Roles evaluators are listed the system will evaluate each line as an AND.</p> <p>The example below will make sure the user has SPEC CREATOR OR SPEC ADMIN AND CREATE FROM TEMPLATE 2280 before they will see the menu item.</p> <pre>&lt;SecurityProfile ID="SpecBlankCreator2280" Mode="Visibility"&gt; &lt;Allow Roles="[SPEC_CREATOR],[SPEC_ADMIN]"/&gt; &lt;Allow Roles="[CREATE_FROM_TEMPLATE_2280]"/&gt; &lt;/SecurityProfile&gt;</pre>
<Deny Roles="[ROLE]"/>	<p>List the user roles that should be denied. Example:</p> <pre>&lt;Deny Roles="[NPD_PARTNER]"/&gt;</pre> <p>Roles separated by commas will be evaluated using an OR condition. If multiple Deny Roles evaluators are listed the system will evaluate each line as an AND.</p>
<Allow Groups=" PKID OF UGM GROUP" />	<p>List the user groups that should be allowed by specifying the pkid of the UGM User Group.</p> <p>Groups separated by commas will be evaluated using an OR condition. If multiple Allow Groups evaluators are listed the system will evaluate each line as an AND.</p>
<Deny Groups="PKID of UGM"/>	<p>List the user groups that should be denied by specifying the pkid of the UGM User Group.</p> <p>Groups separated by commas will be evaluated using an OR condition. If multiple Deny Groups evaluators are listed the system will evaluate each line as an AND.</p>
<Allow	Use a custom class to evaluate security. Example:



Attribute	Description
EvaluatorClassURL="ClassLocation"/>	<Allow EvaluatorClassURL="Class:prodika.MyEvaluator,GSMWebUI"/>

Note: Wildcards can be used to designate that all groups or roles are allowed or denied. The "\*" is used as the wild card.

## Security Profile Examples

### Example 1: Nav A should only appear on the material specification for users who are in GroupA

Step 1. Add a new security profile in MenuConfig-Extensions.xml

```
<SecurityProfile ID="MySecurityProfile1" Mode="Visibility" Application="GSM">
    <Allow Groups="Group A"/>
</SecurityProfile>
```

Step 2. Add menu item node to SiteMap-Extensions.xml

```
<MenuItem ID="GSMNavSpec" configChildKey="ID">
    <MenuItem ID="NavA" DisplayText="NavA" ClientSideCommand="alert('NavA');"
    SecurityProfileRef="MySecurityProfile1"
    Visible="eval:${ObjectType}==1004 &amp; & ${IsInReadMode}" />
</MenuItem>
```

### Example 2: NavB should only appear when the specification is in a workflow status with the name of "Approved"

Step 1. Add a new security profile in MenuConfig-Extensions.xml

```
<SecurityProfile ID="MyEvaluator" Mode="Visibility" Application="GSM">
    <Allow EvaluatorClassURL="Class:prodika.MyEvaluator,GSMWebUI"/>
</SecurityProfile>
```

Step 2. Create a plugin named MyEvaluator and compile.

```
Imports Xeno.Prodika.Application
```

```
Imports Xeno.Prodika.Services
```

```
Imports Xeno.Web.UI.Common.Controls.ObjectHeader
```

```
Imports Xeno.Web.UI.Providers.Security
```

```

Public Class MyEvaluator : Implements ICustomSecurityEvaluator
    Private ReadOnly Property SpecService() As ISpecificationService
        Get
            Return AppPlatformHelper.ServiceManager(Constants.GetInstance()).SPECIFICATION_SERVICE
        End Get
    End Property

    Public Function Evaluate() As Boolean Implements ICustomSecurityEvaluator.Evaluate
        If (SpecService.Current Is Nothing) Then
            Return False
        End If

        //<Your codes here>

        Return //<Your result here>
    End Function
End Class

```

Step 3. Add menu item node to SiteMap-Extensions.xml

```

<MenuItem ID="GSMNavSpec" configChildKey="ID">
    <MenuItem ID="NavA" DisplayText="NavA" ClientSideCommand="alert('NavA');"
    SecurityProfileRef="MyEvaluator"

    Visible="eval:${ObjectType}==1004 & & ${IsInReadMode}" />
</MenuItem>

```

### Visibility (“Visible” Attribute)

You can also use the menu item attribute “Visible” to control the menu item visibility. There are several out of the box server variables you can leverage (see Section 4 for a listing of server variables).

Example of the Save menu item using Visible:

```

<MenuItem ID="lblSave" DisplayText="***Save" Visible="${IsInEditMode}" AutoPostBackOnSelect="true" />

```

The previous example will only show the Save menu item if the specification is in edit mode.

## Visibility Examples

### Example 1: Add NavA to a material spec

```
<MenuItem ID="GSMNavSpec" configChildKey="ID">

  <MenuItem ID="NavA" DisplayText="NavA" ClientSideCommand="alert('NavA');"
  Visible="eval:${ObjectType}==1004" />

</MenuItem>
```

### Example 2: NavA will only appear on a material specification in read mode

```
<MenuItem ID="GSMNavSpec" configChildKey="ID">

  <MenuItem ID="NavA" DisplayText="NavA" ClientSideCommand="alert('NavA');"
  Visible="eval:${ObjectType}==1004 & & ${IsInReadMode}" />

</MenuItem>
```

## Available Server Side Variables

Certain properties can leverage server side values. The following syntax should be used: `${variableName[,defaultValue]}`.

Function	Description
<code>\${dc:APP_NAME}</code>	Used to insert the URL set in the environment settings config. For example, <code>\${dc:GSM}</code> will pull in the GSM URL set inside the environment variables config.
<code>fc:CONFIGURATION_NAME</code>	Used to evaluate the state of a feature configuration For example, <code>eval:\${fc:GSM.SideNav.SpecComparisonReport.Enabled}</code> will pull true if Spec Comparison is enabled and false if it's not.
<code>\${IsInEditMode}</code>	Used to evaluate whether the object is in Edit Mode
<code>\${IsInReadMode}</code>	Used to evaluate whether the object is in Read Mode
<code>\${CanEdit}</code>	Used to evaluate whether the user can edit the object
<code>\${ObjectType}==TYPE#</code>	Used to evaluate what object type is active.  For example, <code>\${ObjectType}==1004</code> will only show the menu item when the active object is a material specification.

Note: If a variable is used and does not existing in that specific application, the system will display a stack trace mentioning that the system is unable to parse a Boolean.

## Referencing the Active Object

Some out of the box JavaScript functions are available so you can pull the active object's identity information. These can be used to directly attach to a URL or even to supplement code extensions. For example, you can grab the object's pkid and use it inside custom code to pull additional data from the specification.

Function	Description
GetObjectHeader_ObjectPKID()	Pulls the active object's pkid
GetObjectHeader_ObjectTitle()	Pulls the active object's title. This can be up to 256 characters.
GetObjectHeader_ObjectType()	Pulls the active object's type and returns it in number format. For example 1004 is a material specification

### Example 1: Opening a popup window and appending the pkid to a URL

```
<MenuItem ID="lblNewItem" DisplayText="NavB" TranslationRef="ApplicationMenu"
    ClientSideCommand="window.open('${dc:GSM}/getspec.aspx?SpecID=' +
    GetObjectHeader_ObjectPKID(),
    'mywindow','height=350,width=400,status=no,toolbar=no,menubar=no,location=no,dependent=yes,scrollbars=yes');"/>
```

## Sort Order

You can use the SiteMap-DisplayOrder.xml file to change the sort order for platform, portal and application navigation items. All the menu items have a SortOrder property that is evaluated and ordered in ascending order.

For example:

```
<MenuItem ID="Menu1" SortOrder="10" configChildKey="ID"/>
<MenuItem ID="Menu2" SortOrder="20" configChildKey="ID"/>
```

The above menus are displayed as in the order Menu1, Menu2. If the sort order is transposed, then Menu2 is displayed before Menu1. The actual number or the spacing between these tags does not matter. In the above example, 10 menu items could be inserted between Menu1 and Menu2 and yet be sorted deterministically. If two menus have the same sort order, then the display order is not guaranteed between those two elements.

**SiteMap-DisplayOrder.xml <MenuItem> Attributes**

Attribute	Description	Required	Comment
ID	Specify ID of the menu item.	Yes	Should be the existing menu ID
configChildKey	Specify identify key of child node.	Yes	Should be "ID"
SortOrder	Specify an order number.	Yes	
Sortable	Specify if the child nodes is sortable	Yes (if has child nodes)	Values: true/false

**Sort Order Examples****Example 1-1: Re-arrange the "New" object menu and Re-arrange "New" and "Open"**

Step 1: Add the following settings to SiteMap-DisplayOrder.xml

```

<MenuItem ID="GSMSideNav" SortOrder="1" Sortable="true" configChildKey="ID">
  <MenuItem ID="lblNew" SortOrder="2" Sortable="true" configChildKey="ID">
    <MenuItem ID="lblTypeNav2147" SortOrder="10" configChildKey="ID"/>
    <MenuItem ID="lblTypeNav5816" SortOrder="20" configChildKey="ID"/>
    <MenuItem ID="lblTypeNav1004" SortOrder="35" configChildKey="ID"/>
  </MenuItem>
  <MenuItem ID="lblOpen" SortOrder="1" configChildKey="ID" />
</SiteMap>

```

**Translation Caches**

All menu items use translatable labels. The following click streams explain how to update an existing label or add a new label to use with a new menu item.

## Add a New Translation

Step 1: Add the new menu item to the SiteMap-Extensions.xml config specifying the menu item ID you are going to use.

Step 2: Identify which translation cache is being used by that menu using the following list. In the table, you will find a list of cache paths that are used, depending on the menu's location in the application suite.

Navigation Menu	Cache Path
Platform Navigation	ApplicationMenu/TopMenu
Portal Navigation	ApplicationMenu/TopMenu
GSM Action Navigation	frmMaster/ctlNavMenu1
SCRM Action Navigation	frmMaster/ctlNavMenu1
NPD Action Navigation	frmNPD/ctlSidebar
NSM Action Navigation	frmNSM/ctlSidebar
eQ Action Navigation	frmEQuestionnaire/ctlSidebar
PQS Action Navigation	frmPQS/ctlSidebar
CSS Action Navigation	frmCss/ctlSidebar
SPA Action Navigation	frmSupplierPortalAdmin/ctlSidebar
ADMN Action Navigation	Portal/DataAdmin
UGM Action Navigation	frmPrincipalManagement
RPT Action Navigation	frmReporting/ctlSidebar

Step 3: Confirm that this translation path has been added to the commonXLAExtensionCache database table. If it has not been added, you will need to add the cache path to this table.

Example script below:

```
insert into commonXLAExtensionCache ( pkid, Name, InheritFrom, Category )
```

```
SELECT '105862C1FABE-E347-4E10-90EC-A2418B893380', N'CACHÉ PATH', N'CACHÉ PATH', N'Nav Extension' FROM DUAL WHERE NOT EXISTS(SELECT 1 FROM commonXLAExtensionCache WHERE pkid = '105862C1FABE-E347-4E10-90EC-A2418B893380');
```

Verify that the pkid that is being inserted is unique per insert. It must start with 1058.

Step 4: Add the new menu item translation item to actual menu to the commonXLAExtensionCacheItem database table. This new menu item needs to be linked to the commonXLAExtensionCache entry you created above.

## Example Script

```
insert into commonXLAExtensionCacheItem ( pkid, fkParent, langID, Id, Value )
```

```
SELECT '1059FBECD5B-7B39-46A2-8C36-4A35EECCFED', 'PKID of translation path from commonXLAExtensionCache', 0, N'NEW  
MENU ITEM ID', N'NEW MENU ITEM LABEL' FROM DUAL WHERE NOT EXISTS(SELECT 1 FROM commonXLAExtensionCacheItem  
WHERE pkid = '1059FBECD5B-7B39-46A2-8C36-4A35EECCFED' and langID = 0);
```

Verify that the pkid that is being inserted is unique per insert. It must start with 1059.

## Adjusting an Existing Translation

Step 1: Identify which translation cache is being used by that menu using the list below. Below you will find a list of cache paths that are used depending on the menu's location in the application suite.

Navigation Menu	Cache Path
Platform Navigation	ApplicationMenu/TopMenu
Portal Navigation	ApplicationMenu/TopMenu
GSM Action Navigation	frmMaster/ctlNavMenu1
SCRM Action Navigation	frmMaster/ctlNavMenu1
NPD Action Navigation	frmNPD/ctlSidebar
NSM Action Navigation	frmNSM/ctlSidebar
eQ Action Navigation	frmEQuestionnaire/ctlSidebar
PQS Action Navigation	frmPQS/ctlSidebar
CSS Action Navigation	frmCss/ctlSidebar
SPA Action Navigation	frmSupplierPortalAdmin/ctlSidebar
ADMN Action Navigation	Portal/DataAdmin
UGM Action Navigation	frmPrincipalManagement
RPT Action Navigation	frmReporting/ctlSidebar

Step 2: Use the Customer Translations toolkit widget. Find the translation cache and update the existing translation item.

More information on this widget can be found in the *Agile Product Lifecycle Management for Process Data Administration Toolkit Guide*.

