Managing Menus

This chapter contains the following sections:

Menu Navigation

Cisco UCS Director uses menu navigation to determine what reports and forms to display in the UI. For more information on the subject of report locations, refer to Specifying the Report Location.

The leftNavType field specifies the type of navigation to be used in your menu item.

The value none means that:

- No navigation is required.
- The context map rule associated with the menu item will use type = 10, name = "global_admin". (Important!)

Tip When the leftNavType is set to none, the type value and name value for the context map rule associated with the menu item comes in handy when you need to register your reports to this menu location.

If the leftNavType is backend_provided, you must provide an implementation of com.cloupia.model.cIM.AbstractTreeNodesProviderIf that populates the left hand navigation tree.

Each node of the navigation tree must provide the following elements:

- A label
- The path to an icon to show in the UI (optional)

Note The size of the icon should be 24 x 24 pixels and the format of the icon should be PNG.

- The context type (for more details, see the section about registering report contexts)
- The context ID (this will become the report context ID that you may use when generating tables)

The navigation tree must be associated with a menu ID. When registering the tree provider, use the corresponding menu ID.

Table 1: System Menu ID for Virtual Account

Menu	ID
Compute	0
Storage	1
Network	2

Table 2: System Menu ID for Physical Account

Item	ID
Compute	50
Storage	51
Network	52

Defining a Menu Item

Step 1 Option 1: Add a new menu item underneath an existing folder; in this case, the one called **Virtual**.

When adding a menu item into an existing menu category, you first have to locate the menuid of the category to which you want to add the item. In the example, we add the new menu item under "Virtual", which has the menuid of 1000. Take note of the parent menu item with just the menuid filled in: this is all you need in order to signal that you are placing your menu item into an existing category. The new menu item is placed into the children field.

Example:

```
<menu>
 <!-- this shows you how to add a new menu item underneath virtual -->
 <menuitem>
   <menuid>1000</menuid>
     <children>
       <menuitem>
          <menuid>12000</menuid>
          <label>Dummy Menu 1</label>
          <path>dummy_menu_1/</path>
          <op>no_check</op>
          <url>modules/GenericModule.swf</url>
          <leftNavType>backend_provided</leftNavType>
       </menuitem>
   </children>
 </menuitem>
```

Step 2 Option 2: Add an entirely new menu item into the UI.

If you are defining an entirely new menu item, provide all the details as shown in the example. First provide all the details for the menu category, then add all the child menu items underneath it. The example here shows a menu two levels deep, but in theory you can go as deep as you want. The best practice is to create menus no more than three levels deep.

Example:

```
<!-- entirely new menu -->
<menu>
  <menuitem>
     <menuid>11000</menuid>
     <label>Sample Category</label>
     <path>sample/</path>
     <op>no_check</op>
     <children>
       <menuitem>
         <menuid>11001</menuid>
         <label>Sample Menu 1</label>
         <path>Sample_menu_1/</path>
         <op>no_check</op>
         <url>modules/GenericModule.swf</url>
         <leftNavType>backend_provided</leftNavType>
       </menuitem>
     </children>
  </menuitem>
<menu>
```

What to do next

Register the menus.

Registering a Menu Item

For Open Automation, menu registration is handled automatically. As a developer, you only need to name the xml file of your menu as `menu.xml`, then package it as part of your module. Ensure that the `menu.xml` file is at the top level of the module jar file.

Before you begin

Define a new menu item under either a new or an existing folder.

Registering Report Contexts

This topic focuses on adding new report contexts. When developing new menu items, new report contexts are crucial: you must register new unique contexts, you CANNOT use existing contexts.

The Open Automation documentation about defining menu navigation briefly mentions that you need to provide a report context type when building your left hand navigation tree provider.

Report contexts are used by the system to determine which reports can be displayed at any point in the UI. For more background information, refer to the documentation on specifying report location: Specifying the Report Location. See also the list of existing report context data in Appendix B.

For open automation, there are APIs in place to auto-generate a new report context. Refer to `com.cloupia.feature.foo.FooModule` for examples on registering report contexts and menu providers.

Tip Auto generated report contexts are not portable. This means that if you deploy your module in one instance of UCSD and the same module in another instance of UCSD, the auto-generated report context you get in each instance may have different values. Thus, any code you write that uses those duplicate values will not necessarily work! To avoid such problems, use the ReportContextRegistry to register report contexts and retrieve them.

Use com.cloupia.model.cIM.ReportContextRegistry.register(String name, String label), and take a look at the javadocs and sample code for more detail.

Refer to code samples and the Specifying Report Location document to see how these report contexts ultimately end up being used.

Before you begin

Open Automation developers who need to register report contexts should first talk to a UCSD lead. The UCSD lead can provide you with a block of integers reserved exclusively for your use. This will guarantee that any report contexts you define are unique. When you have your block, you can use ReportContextRegistry.register(int type, String name, String label) to register the new context.