

Determine the DN of an Object Within the APIC GUI



Document ID: 119419

Contributed by Shae Eastman and Gabriel Monroy, Cisco TAC Engineers.

Jan 04, 2016

Contents

Introduction

Determine the DN of an Object Within the APIC GUI

- Use the Browser URI

- Use the Right-Click Menu

- Use the *Show API Inspector* Feature

- Use the *Show Debug Info* Feature

Introduction

This document describes how to obtain the Distinguished Name (DN) of an object within the Cisco Application Policy Infrastructure Controller (APIC) GUI. When you troubleshoot the logical model or attempt to create a script on the APIC GUI, it can be difficult to match the object that you want to interact with on the GUI with the actual class name, or obtain its DN in order to view the Managed Object (MO) attributes in Visore.

Determine the DN of an Object Within the APIC GUI

There are many available methods that can be used in order to determine the DN of an object that you see in the GUI. This section describes four different methods.

Use the Browser URI

While you navigate to the GUI, you will notice that there are many dynamic elements loaded. These elements respond to your clicks as you navigate through the GUI representation of the Object Model. When you click an object, the URL within your browser should update in order to include the DN of the object that you select.

For example, if you click tenant *common*, the URL updates to reflect this:

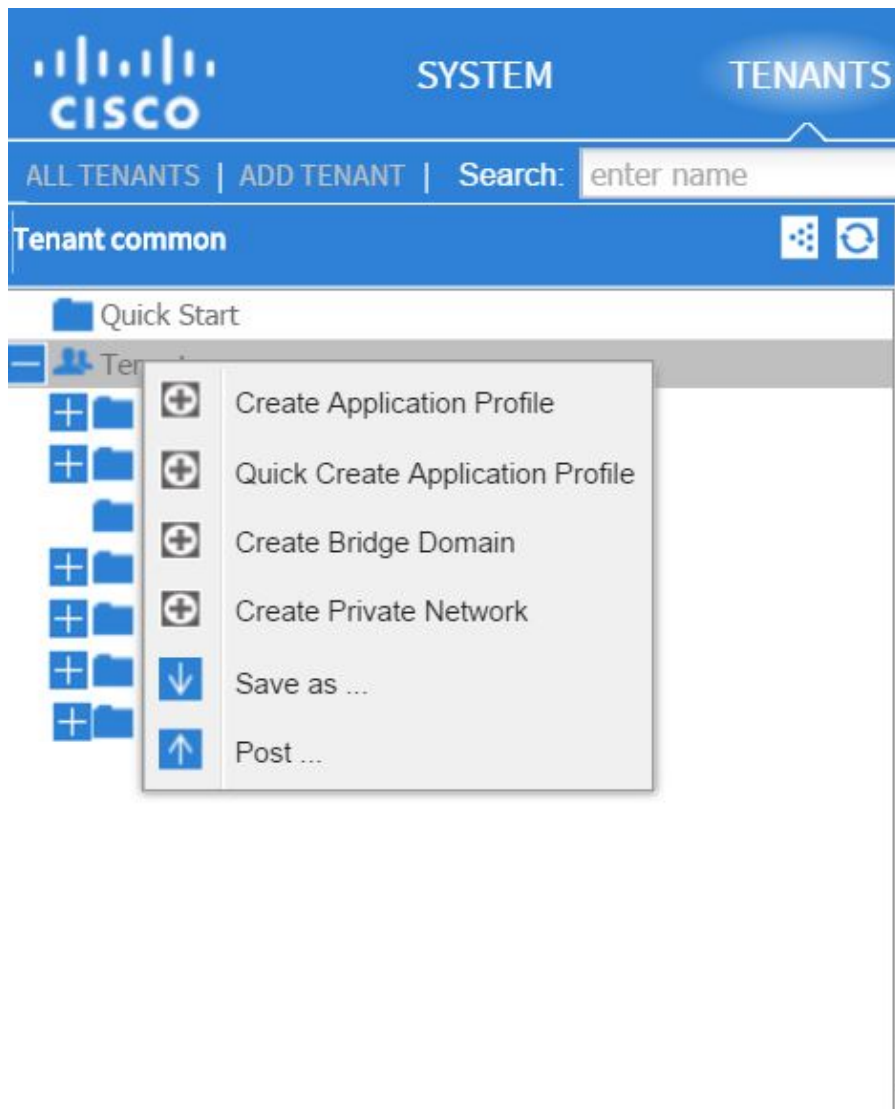
```
http://x.x.x.x/#bTenants:common|uni/tn-common
```

As you read the Uniform Resource Identifier (URI), notice that the DN of the object that you last clicked appears after the pipe (|).

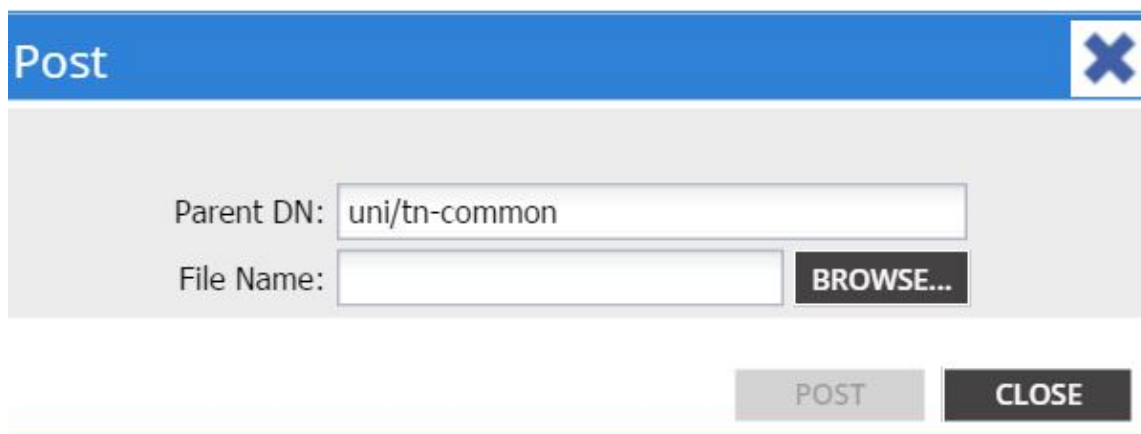
Note: If you select GUI entities to which objects do not correspond (such as folders), then you will not see a DN.

Use the Right-Click Menu

For some of the policies in the GUI, you can right-click on the policy and see a **Post ...** option:

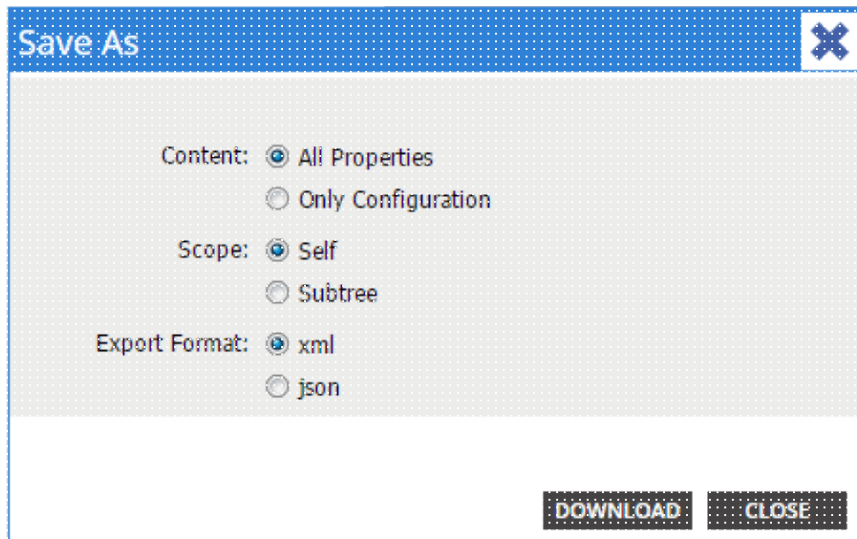


When you click this option, you are presented with the DN for that policy/MO:



Similarly, you can right-click on a policy and choose the **Save as ...** option in order to download either an XML or JSON output of the selected object and various properties (as well as special filters, if needed).

This image shows the available selections when you choose the *Save as...* option:



Here are some notes about these selections:

- **Content**

- ◆ Select the **All Properties** radio button in order to return every attribute that is associated with the object.
- ◆ Select the **Only Configuration** radio button in order to return only the attributes that are user-configurable.

- **Scope**

- ◆ Select the **Self** radio button in order to return only the object that is selected.
- ◆ Select the **Subtree** radio button in order to return the selected object and any children of that object.

- **Export Format**

- ◆ Select the **xml** radio button in order to download the content in XML format.
- ◆ Select the **json** radio button in order to download the content in JSON format.

This example shows the *Save as ...* operation for the tenant *common*, where the *All Properties*, *Self*, and *XML* selections are specified:

```
<?xml version="1.0" encoding="UTF-8"?>
- <imdata totalCount="1">
  <fvTenant uid="0" status="" ownerTag="" ownerKey="" name="common"
    monPolDn="uni/tn-common/monepg-default" modTs="2015-09-
    24T16:51:05.671+00:00" lcOwn="local" dn="uni/tn-common" descr=""
    childAction=""/>
</imdata>
```

This example shows the *Save as ...* operation for the tenant *common*, where the *Only Configuration*, *Self*, and *XML* selections are specified:

```

<?xml version="1.0" encoding="UTF-8"?>
- <imdata totalCount="1">
  <fvTenant ownerTag="" ownerKey="" name="common" dn="uni/tn-common"
    descr=""/>
</imdata>

```

Notice that both versions of the content selections provide the DN of the chosen object. You also receive the name of the class, which in this case is **fvTenant** (shown as the start of the object instance). This is useful for Visore or MOquery.

Tip: If you select the *Subtree* radio button for the scope, you receive information for all of the child objects in the same export. This option allows you to obtain all attributes, which gives you visibility into the object in its entirety.

Use the *Show API Inspector* Feature

You can take the subscription requests directly from the Application Policy Infrastructure (API) inspector in order to obtain the DN of an object. In order to enable this, click **Show API Inspector** under the *Welcome*, `<username>` message near the top-right navigation bar.

Here is an example that shows the health of the tenant *common*:



Here is an example that provides a look inside of the API inspector for tenant *common*:

```

method: GET
url: https://10.122.141.109/api/node/mo/topology/HDfabricOverallHealth5min-0.json
response: {"totalCount": "1", "imdata": [{"fabricOverallHealthHist5min": {"attributes": {"childAction": "", "cnt": "30", "dn
timestamp: 15:36:06 DEBUG
method: GET

```

Use the *Show Debug Info* Feature

If you desire to have a more in-depth analysis, you can enable the browser debugging information feature, which includes the DN of the object that you select.

In order to enable this, click **Show Debug info** under the *Welcome, <username>* message near the top-right navigation bar.

Once this is enabled, there is some fine print near the bottom of the browser window that begins with the words **Current Screen**.

When you click tenant *common*, you will see this output:

```
Current Screen:insieme.stromboli.layout.Tab [fv:infoTenant:center:a ] | Current  
Mo:insieme.stromboli.model.def.fvTenant [uni/tn-common ]
```

Note: The information provided is somewhat verbose, but you also get the class of the object that you have selected. In this case, it is **fvTenant**. The rest of this output is likely only useful if you desire to modify the GUI.