

Problem

How can I add components to the ACS CDB without modifying \$ACS_CDB/MACI/Components/Components.xml?

Solution

Instead of just having one file, *Components.xml*, in your CDB you now have the option to replace it with a separate directory which includes a single xml file "for each component entry". To allow this ACS has added an additional schema, *Component.xsd* (notice this schema is for a single component), to be used instead of the *Components.xsd* schema that the *Components.xml* file uses currently.

For example:
If you previously had:

\$ACS_CDB/CDB/MACI/Components/Components.xml

Where the Components.xml file looked as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<Components xmlns="urn:schemas-cosylab-com:Components:1.0"
  xmlns:cdb="urn:schemas-cosylab-com:CDB:1.0"
  xmlns:baci="urn:schemas-cosylab-com:BACI:1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >

  <_ Name="MOUNT1"      Code="acsexmplMountImpl"
                                Type="IDL:alma/MOUNT_ACS/Mount:1.0"
                                Container="bilboContainer"/>
  <_ Name="MOUNT2"      Code="acsexmplMountImpl"
                                Type="IDL:alma/MOUNT_ACS/Mount:1.0"
                                Container="bilboContainer"/>
  <_ Name="HELLODEMO1"  Code="alma.demo.HelloDemoImpl.HelloDemoHelper"
                                Type="IDL:alma/demo/HelloDemo:1.0"
                                Container="frodoContainer"/>
  <_ Name="HELLOLAMP1"  Code="alma.demo.LampAccessImpl.LampAccessHelper"
                                Type="IDL:alma/demo/LampAccess:1.0"
                                Container="frodoContainer"/>

</Components>
```

The new CDB structure you would have would look like:

\$ACS_CDB/CDB/MACI/Components/MOUNT1/MOUNT1.xml

\$ACS_CDB/CDB/MACI/Components/MOUNT2/MOUNT2.xml

\$ACS_CDB/CDB/MACI/Components/HELLO1/HELLO1.xml

\$ACS_CDB/CDB/MACI/Components/HELLO2/HELLO2.xml

An example of the xml file would be

(C++ example for the MOUNT1.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<Component xmlns="urn:schemas-cosylab-com:Component:1.0"
  xmlns:cdb="urn:schemas-cosylab-com:CDB:1.0"
  xmlns:baci="urn:schemas-cosylab-com:BACI:1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

  Name="MOUNT1"
  Code="acsexmplMountImpl"
  Type="IDL:alma/MOUNT_ACS/Mount:1.0"
  Container="bilboContainer"
/>
```

(Java example for HELLO1.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<Component xmlns="urn:schemas-cosylab-com:Component:1.0"
  xmlns:cdb="urn:schemas-cosylab-com:CDB:1.0"
  xmlns:baci="urn:schemas-cosylab-com:BACI:1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

  Name="HELLODEMO1"
  Code="alma.demo.HelloDemoImpl.HelloDemoHelper"
  Type="IDL:alma/demo/HelloDemo:1.0"
```

```
Container="frodoContainer"
/>
```

Important things to note:

1. The components xml filename must match the directory name which it is in.
For example, if you create \$ACS_CDB/CDB/MACI/Components/COMP1, the xml file must be called COMP1.xml
2. Some components, particularly C++ components, require instance xml files. This is usually associated with components which use properties. In this case the file hierarchy in the \$ACS_CDB/CDB/MACI/Components directory and the \$ACS_CDB/alma directory must match.

For example if you have

```
$ACS_CDB/CDB/MACI/Components/MOUNT1/MOUNT1.xml
```

then you must have

```
$ACS_CDB/CDB/alma/MOUNT1/MOUNT1.xml
```

If you are not using properties in C++ you do not need to do this because you do not have an entry in \$ACS_CDB/CDB/alma. Likewise for Java components or Python components, which usually do not require properties, you can call the new directory and xml file anything you want.

-- [SohailaLucero](#) - 25 May 2004

-- [GianlucaChiozzi](#) - 03 Feb 2005

Related articles

- [How can more people do development with ACS on the same machine without disturbing each other?](#)
- [Which ports are used by ACS?](#)
- [Problems connecting to ACS servers on a remote machine: bad /etc/hosts](#)
- [Why does the getComponent method of ZLegacy/ACS.ContainerServices return an object of type None?](#)
- [Why are some of my print statements not showing up in the container output section of acscommandcenter?](#)