

Problem

Can I put many xml files in the same directory to specify the deployment?

Solution

Since ACS 4.1.0, it is possible to put multiple xml files in the same directory, each with a number of Components inside, by using the recent [XInclude](#) and [XPointer](#) XML specification.

See [CDBProblems#Multiple_Xml_Files_In_Cdb_Direct](#) for a detailed discussion about usage and implementation of XInclude and Xpointer in xerces-2 java, our parser (and the patches we have implemented).

- We can put Component specifications in separate files to be included in a main `Components.xml`
- Each of these external files must be a well formed Components description file=
- The Components described in such files are included in the main `Components.xml` using the following syntax: (2)

```
<xi:include href="relative path/MyIncludeFile.xml" xpointer="element(/1)" />
```

See also:

- [FAQHierarchicalComponentsAndCDBStructure](#)
- [FAQDynamicComponentsAndCDBStructure](#)

for more details of usage and examples

-- [GianlucaChiozzi](#) - 06 Sep 2005

ACS < 4.1.0

With versions of ACS < 4.1.0, this was not possible. Here is the old explanation

The design of the ACS CDB is driven by the need to provide users with an "XML Abstraction" for the configuration database, i.e. from the user point of view every Node in the hierarchical structure of the Configuration Database shall appear as a valid XML document.

This is independent from the actual implementation on the backend.

We are now providing a Java implementation (called `cdbjDAL`) and we will provide an implementation of the same interfaces using the ALMA Archive.

The Java implementation could have used a single XML file for the whole configuration database, but we have preferred to make it more modular using the hierarchical structure of the file system.

In doing this we have introduced the mapping:

- **Directory <=> XML node**
 - Each directory in the filesystem corresponds to a node in the virtual XML document constructed by loading all files.
 - Inside each directory we can put both:
 - other sub-nodes as directories
 - one XML file with the same name of the directory
This file contains the XML description for the node, being its root node the very same node identified by the directory. This must be a complete and well-formed XML file, so that it is possible to validate it when loading it.

If we put two XML files in the directory, it is not possible any more to build a valid XML document by loading both of them!

This is because we would have two root nodes and this is not allowed by the XML specification

Consider the following example:

- `/Components` - Is a directory
 - `SomeComponents.xml` - Is an XML file with the following contents:

```
<?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="PBEND_B_01" Code="acsexmplPowerSupplyImpl"
    Type="IDL:alma/PS/PowerSupply:1.0"
    Container="bilboContainer" />

  <_ Name="PBUMP_B_01" Code="acsexmplPowerSupplyImpl"
    Type="IDL:alma/PS/PowerSupply:1.0"
    Container="bilboContainer" />

</Components>
```

- **OtherComponents.xml** - Is another XML file with the following contents:

```
<?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="TEST_PS_1" Code="acsexmplPowerSupplyImpl"
    Type="IDL:alma/PS/PowerSupply:1.0"
    Container="bilboContainer" />

  <_ Name="TEST_PS_2" Code="acsexmplPowerSupplyImpl"
    Type="IDL:alma/PS/PowerSupply:1.0"
    Container="bilboContainer" />

</Components>
```

We can make the following considerations:

- Both XML files are valid, but concatenating them we obtain an invalid XML document with two root nodes.
- We could remove the external node and generate one wrapper node while loading them, called as the containing directory. But this would leave the two "fragment XML" files invalid.
- We could enforce the convention that the the root node of all files is identical, i.e. same name and same type, and while merging the files keeping only one root node as the container for all sub-nodes in all files. But this is not easily maintainable and is error prone.

We think that none of these solutions would be acceptable and therefore we prefer to stick to the rule that only one XML file per directory is allowed and suggest to use the hierarchical directory structure.

A clean solution for the problem of splitting information over multiple XML files in the same directory would come from the usage of the [XInclude specification](#), but there are currently implementation problems discussed in the topic [CdbXInclude](#).

-- [GianlucaChiozzi](#) - 22 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?](#)