Problem

What are dynamic components, and how are they used?

Solution

General

There are two different types of components:

1. **static components** whose instances are fixed at deployment time. These typically represent (abstracted) hardware devices, or service components for which we choose a fixed number of instances. One component can be deployed many times under different names, like Pipeline1, Pipeline2.
2. **dynamic components** whose type (and optionally container location) is fixed at deployment time, whereas the number and names of instances is determined only at runtime.

Dynamic components should be used when

1. the number of instances is not known at deployment time. To draw an analogy to object-oriented programming, this would correspond to creating new objects on the stack or heap, as opposed to retrieving pre-existing instances from an object cache.
2. the client needs a "personal" component only for itself, that is, a component with client state. Another component can offer a kind of factory method, inside which a new dynamic component instance is created. The factory method would then return the component ID to the client, and the client could obtain a reference to its personal component using this ID.

Usage (Java)

In the `ZLegacy/ACS.ContainerServices` interface, there are two methods for creating dynamic components:

1. `public org.omg.CORBA.Object getDynamicComponent(ComponentQueryDescriptor compSpec, boolean markAsDefault)` offers safe access to dynamic components.
   The `ComponentQueryDescriptor` allows to specify the type, and optionally the instance name, of the new component. If the name is not given, it will be generated.
2. `public org.omg.CORBA.Object getDynamicComponent(ComponentSpec compSpec, boolean markAsDefault)` is similar, but allows to specify advanced options that may cause unwanted effects if not used carefully. In addition to the component type and name, also the component implementation class and the container name can be specified dynamically, similar to how it's done for static components in the CDB.

When optional values are not specified, the ACS Manager will try to find defaults, blending in CDB config data (see below).

Also see the example application in the module `jcontexmpl` (`./src/alma.demo.dyncomp.JDynAct`), and the test component

Request Format, CDB configuration, Manager logic

The possible requests for a dynamic component fall into one of two categories, depending on whether `componentName` is specified in the call:

1. specified `componentName`
   - if all entries are specified, the dynamic component is activated using the given entries
   - if there is an entry with unspecified parameter (i.e. "*" value):
     - search in the CDB for entry with the same ***componentName***
       - if there are more than one entries matching search criteria, the closest match of ComponentSpec is used, search priority is: ***componentName***, ***componentType***, ***componentCode***, ***containerName***
     - override fields (***componentType***, ***componentCode***, ***containerName***) retrieved from the CDB with the given fields in ***ComponentSpec***
     - if there exists unspecified field throw ***IncompleteComponentSpec*** exception
   - check if there is already a component with ***componentName*** name activated, if it is not compatible with the given ***componentSpec*** th row ***ComponentSpecIncompatibleWithActiveComponent*** exception, otherwise return existing reference or activate the component
2. unspecified `componentName` case (implies that a new component will be activated)
   - precondition for CDB: the component must be deployed w/o a name, i.e. `Name="*"`
   - precondition for call: `componentType` must also be specified, if not an ***IncompleteComponentSpec*** exception is thrown
   - search for ***componentName*** == '*' and right ***componentType*** entry in the CDB (again closest match is taken).
   - override fields (***componentCode***, ***containerName***) retrieved from the CDB with the given fields in ***ComponentSpec***
   - if there exists unspecified entry (w/ exception of ***componentName***) throw ***IncompleteComponentSpec*** exception
   - generate ***componentName*** = ***componentType*** + '_' + number
   - activate the component

For details, refer to the discussion of `ComponentSpec` in NewMaciIdl.

Something else to work in here (from ZLegacy/ACS.ReleaseNotes_ACS_3_0_0):

- NOTE: if a dynamic component is marked as default via get_dynamic_component method, then released, it still remains as a default component, request for it will activate it
- NOTE on get_component_info method: inactive dynamic component are not returned, in other words entries containing '*' are not returned
- NOTE on dynamic component entries: since dynamic components are in the same Component.xml table as static, there is the following restriction: component name must act like a primary key, with exception of "*" entries

-- HeikoSommer - 17 Feb 2004

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?