

Problem

What package should I use for the component implementation classes?

Solution

When the build process ("make all") compiles the IDL file that contains your component's interface, it creates Java classes according to the CORBA spec for IDL->Java mapping, with the package being determined by the pragma prefix and the IDL module name. These classes are packed in `../lib/.jar`.

For example, prefix `"alma"`, module `"demo"`, interface `"XmlComponent"` yield the class `alma.demo.XmlComponent` inside a jar file under `lib`.

Under your module's `src` directory, you should create a corresponding package tree, with a subpackage for the implementation classes of each component. The name should be that of the interface, with `"Impl"` appended. In the above example, we would create the Java package `"alma.demo.XmlComponentImpl"`.

This naming convention is gently enforced by the framework through the generated component helper classes. With `COMPONENT_HELPERS=on` in the Makefile, you'll get the right package created under `src`, and the file `Helper.java.tpl` is put there automatically. When you remove the `".tpl"` ending, you'll notice that the generated method `_createComponentImpl()` attempts to instantiate the (yet missing) component implementation class from that same package. In the example, the implementation class would be `"alma.demo.XmlComponentImpl.XmlComponentImpl"`.

If you wonder about the seemingly silly repetition of `"XmlComponentImpl"` in the class and the package name, please consider that an IDL module can contain many interfaces (`= components`), and that it's favorable to group the classes that make up one component's implementation within one subpackage. Classes that are used in several component implementations should go into a different shared package.

Without trying to infringe on the standard ruling powers of ALMA software engineering, it seems reasonable to group the implementation classes of very similar components into one `Impl` package, that contains several `-Impl` and `-Helper` classes.

Note: 2003-11-25, G.Chiozzi: with ACS 3.0 the standard for ALMA prefix has been changed from upper case `"ALMA"` to lower case `"alma"`.

-- [GianlucaChiozzi](#) - 30 Aug 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`?](#)