

Problem

What is the '#pragma prefix' in IDL?

Solution

Each IDL file written for ACS shall have a line like this:

```
#pragma prefix "alma"
```

FOLLOWING ALL #include DIRECTIVES!

Use the "alma" prefix and not "ALMA". The IDL #pragma prefix is used to:

- generate Java package names as a prefix package
- generate IDL Interface Repository identifiers

The IDL compilers we are using have all limitations in the usage of the #pragma prefix.

To avoid problems with these limitations we have adopted the following conventions:

- always use "alma" prefix unless you have very specific reasons to do otherwise. This will make your life much simpler.
- in any case, do not use hierarchical prefixes like "ALMA.mysystem".

Support from the IDL compilers is broken and not coherent.

Updated with ACS 6.0

This positioning of the pragma prefix has always been the rule, but the ACS/TAO version distributed with ACS 6.0 has re-introduced an old bug that makes this even more important.

You will have problems contacting C++ Components from Java and Python if you do not to this.

A symptom of this problem is a message like the following logged by the Manager while activating a component:

```
Activated component 'ERRORCOMP_CPP' does not implement specified type 'IDL:alma/acsexmplErrorComponent/
ErrorComponent:1.0'.
```

Another symptom of this, in your client code, will be a failure (exception) when trying to narrow the component after doing a GetComponent call:

```
2007-04-12T23:23:20.434 SEVERE [LogStress] Client application failure
org.omg.CORBA.BAD_PARAM: Narrow failed vmcid: 0x0 minor code: 0 completed: No
```

If you have the following IDL:

```
#pragma prefix "alma"

#include <acscomponent.idl>
#include <ACSErrTypeCommon.idl>

module acsexmplErrorComponent
{
    interface ErrorComponent : ACS::ACSComponent
    .....
}
```

TAO IDL compiler generates askeleton that declares the object as implementing the IDL interface:

- **IDL:acsexmplErrorComponent/ErrorComponent:1.0**



This is wrong!

If instead you have the following IDL:

```
#include <acscomponent.idl>
#include <ACSErrTypeCommon.idl>

#pragma prefix "alma"

module acsexmplErrorComponent
{
    interface ErrorComponent : ACS::ACSComponent
    .....
}

you get the correct value:

    • IDL:alma/acsexmplErrorComponent/ErrorComponent:1.0
```

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?](#)