

Problem

Run time and compile time problems with IDL enum

Solution

IDL enums have some limitations.

There are two different cases when one can run into problems with IDL enums: at run-time when the "acsStartORBSVC" script is run or at compile-time.

When you see "bad" output from the TAO interface repository at run-time, there are generally two different cases.

The first is that the same enum has been specified in multiple files. So long as the module name is different between two different IDL files, this should not be a problem.

The second case is a little bit trickier: IDL does not allow a second enum to use one of the previous enum's enumerations in the same namespace. This is because enums are NOT in the scope of the enum type, i.e. the following IDL is illegal:

```
module DOM {
  interface Node {
    typedef enum _DocumentOrder {
      PRECEDING,
      FOLLOWING,
      SAME,
      UNORDERED
    } DocumentOrder;

    typedef enum _TreePosition {
      PRECEDING,
      FOLLOWING,
      ANCESTOR,
      DESCENDENT,
      SAME,
      UNORDERED
    } TreePosition;
  }
}
```

because PRECEDING is defined twice.

This produces a compile time error. The error output by the Makefile should be something similar to:

"../idl/crap.idl", line 42: redefinition after use, ::DOM::PRECEDING, ::DOM::PRECEDING

This is a limitation of the IDL specifications needed to support C binding. See http://www.eso.org/projects/alma/faqs/bin/faqs?_highlightWords=enum&file=95 for details.

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?](#)