

Problem

Why does the container emit a "Failed to create Python object" message when I try activating my component?

Solution

The simple answer is because for some reason or another the Python container is unable to create the component based on the information found in \$ACS_CDB/MACI/Components/Components.xml (or from a dynamic component client).

Some of these reasons include but are not limited to:

- the *Code* attribute specifying the Python package and class for the component defined in Components.xml is incorrect
- the *Type* attribute specifying the interface repository location of the component's IDL interface defined in Components.xml is incorrect
- the constructor of your Python implementation of the IDL interface takes parameters when it should not
- the constructor of your Python implementation of the IDL interface **raises any exception**
- the Python CORBA stubs for your IDL interface are not in your \$PYTHONPATH environment variable. There is a very high probability this is the case if the IDL module (i.e., namespace) your IDL interface is defined in is reopened in other ALMA CVS modules. See [ALMASW2002083](#) in the [ALMA SPR system](#) for more details.
- the component baseclass(es) and/or container cannot find a definition for your component's IDL type within the CORBA IFR. If you see an error message complaining about some derivation of the word **interface** such as:

```
Traceback (most recent call last):
  File "../lib/python/site-packages/Acspy/Container.py", line 238, in activate_component
  File "../lib/python/site-packages/IERSImpl/IERS.py", line 60, in initialize
    addProperty(self, "dUT")
  File "/alma/ACS-4.1/ACSSW/lib/python/site-packages/Acspy/Util/BaciHelper.py", line 128, in addProperty
    interf = interf._narrow(CORBA.InterfaceDef)
```

this is most definitely the case. Typical causes of this are failure to install your module into an \$INTROOT **before** issuing the *acsStart* command. Additionally, it's possible that some bad IDL (i.e., interfaces declared twice, enum values declared twice in the same namespace, etc) has corrupted the Interface Repository. Please pay very close attention to the output coming from *acsStart* if you think this may be the case. To verify that this type of error is occurring, simply start ACS and run the *irbrowser* command to see if your IDL interface is really loaded into the CORBA interface repository.

Also, if you're running the Python container from the *acscommandcenter* you're not seeing the entire output of the container (temporarily at least) and more importantly the underlying error message from the Python interpreter. Try starting the container from the command-line instead. See [FAQ on Delayed Output](#) for more details.

-- [DavidFugate](#) - 06 Aug 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?](#)