

Problem

Why do I have (for some modules) make both the WS and LCU sides of a module even if I am only using the LCU side?

Solution

The ACS Makefile has been implemented with the idea of avoiding redundancy of files, by keeping in one place only files that are or can be used both on the workstation and LCU side.

We have decided to conventionally place all common files in the workstation side.

We could have done the other way around, but our rationale was that to make simulations and tests we will have setups where we have only the workstation, while there will be no possibility of a setup with only LCUs, since a workstation is needed to build the LCU.

To this purpose, when building an LCU application the ACS Makefile always has the vw directories before the workstation side directories in each search path.

For example, the include path (or the IDL search path) built by the Makefile for my setup contains:

```
-I .  
-I../include  
-I../ws/include  
-I/introot/gchiozzi/vw/include  
-I/introot/gchiozzi/include  
-I/alma/OCT2001/ACSSW/vw/include  
-I/alma/OCT2001/ACSSW/include
```

and so on (use the `MAKE_VERBOSE=1` Makefile command line option to see it).

The advantage is that you cannot have misaligned systems (one file on the WS side and a different one in the LCU side).

The disadvantage is that you have to compile (or in many case just to install) the WS part when you change common files.

All ACS modules have been implemented in this way. No include file (or IDL file) already present in the ws branch is duplicated in the lcu branch. This means that all ACS modules (including enumprop) have to be always built and installed on the ws side before the lcu side.

Another approach is to duplicate all files (or create links) in the LCU part and you will be able to build just the LCU part.

The Makefile would work fine, since the LCU files would be always found on the vw directory branch before looking in the workstation branch. You would pay this with the risk of inconsistencies: what if you change a file and make all install only the LCU side? The workstation side would remain the old one.

As you can see, this is not a limitation of the ACS Makefile, but an explicit implementation choice we have adopted for ACS. For example the tics module is instead explicitly linking the include files from ws/include into lcu/include. In this way you need to build only in one place.

Based on my personal experience, I prefer very much the approach adopted in ACS, because I find it safer: whenever there is any kind of redundancy you can (and will) have misalignment problems. Also, the alternative approach used by some TICS module makes the LCU Makefile more complex, since you have to add the rules for making the links.

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?](#)