

# Building omniORB for GLAST

## To obtain the source or binaries:

<http://omniorb.sourceforge.net/>

## Building omniORB

This is just an example, always refer to the README files for the version you download to check to see if the instructions have changed.

- create a directory such as rh9\_gcc32
- cd into that directory
- run configure, for example: `../configure --prefix=$myDir PYTHON=$GLAST_EXT/python/2.5.1/bin/python --disable-ipv6`
- make
- make install

We provide the include and lib directories with our external distributions for GlastRelease.

## Preparing to use the new omniORB in GlastRelease

Update IExternal/omniORB's requirements file as necessary to point to the current set of OmniORB libraries.

## Regenerating files using omniidl

<http://omniorb.sourceforge.net/omni41/omniORB/omniORB005.html>

In our HepRepCorba package, we save two \*.idl files: HepRep.idl and HepEventServer.idl. We use these as input to the omniidl binary that is created when we compile omniORB. These files must be regenerated each time we upgrade our version of OmniORB.

```
omniidl -bcxx -Wba HepRep.idl
producing
HepRep.hh HepRepSK.cc, and HepRepDynSK.cc
```

and

```
omniidl -bcxx HepEventServer.idl
```

The product is the files: HepEventServer.hh and HepEventServerSK.cc

Rebuild HepRepCorba

In the HepRepCorba cmt requirements file there is a line  
#Set some OmniORB env variables  
#This is useful for debugging  
set ORBtraceLevel 0

Documentation pertaining to this env variable is available here: <http://omniorb.sourceforge.net/omni41/omniORB/omniORB004.html#toc20&nbsp>

*The following options control debugging trace output.*

`traceLevel default = 1`

*omniORB can output tracing and diagnostic messages to the standard error stream. The following levels are defined:*

level 0	critical errors only
level 1	informational messages only
level 2	configuration information and warnings
level 5	notifications when server threads are created and communication endpoints are shutdown
level 10	execution and exception traces
level 25	trace each send or receive of a giop message
level 30	dump up to 128 bytes of each giop message
level 40	dump complete contents of each giop message

*The trace level is cumulative, so at level 40, all trace messages are output.*

## Troubleshooting

While upgrading to 4.1.0, ran into an error on Linux Unable to create an endpoint:

<http://www.omniorb-support.com/pipermail/omniorb-list/2006-December/028267.html>