

# Building Gaudi

## Building Gaudi v21r7 against ROOT v5.34

### redhat5-i686-32bit-gcc41

```
setenv CMTBASE $userSpace/gaudi_v21r4/CMT
setenv CMTVERSION v1r20p20090520
source $CMTBASE/$CMTVERSION/mgr/setup.csh
setenv CMTCONFIG i686-slc5-gcc41-opt
setenv CMTPATH $userSpace/gaudi-rhel5-32-ROOTv5.34/GAUDI/GAUDI_v21r7:$userSpace/gaudi-rhel5-32-ROOTv5.34/external:$userSpace/gaudi-rhel5-32-ROOTv5.34/external/LCGCMT/LCGCMT_58:$userSpace/gaudi_v21r4
```

Removed HistogramSvc from the GaudiSvc requirements due to complete re-write to THistSvc in later Gaudis to comply with ROOT v5.34.

Gaudi uses CMT, so building goes something like this

We choose to limit the Gaudi packages we build to (in this order): GaudiKernel, GaudiSvc, GaudiUtil, GaudiAlg, GaudiAud

For each package, enter the cmt directory and do:

```
cmt broadcast "cmt config"
source setup.sh
make
```

## Building Gaudi v21r7

### Requirements

- CMT v1r20p20090520 available on SLAC Linux on /afs/slac/g/glast/users/heather/gaudi\_v21r4/CMT and through the download section of the [CM T web site](#).
- Source Code modified for GLAST to limit the externals required by Gaudi see /afs/slac/g/glast/ground/GLAST\_EXT/srcExtLibs/GAUDI\_v21r7-gl1\_src.tar.gz
- LCGCMT\_58 zip file of externals and interface CMT packages for Gaudi
- An updated version of ROOT v5.26.00a-gl1, the binaries are available in GLAST\_EXT at SLAC and the source is in compiled against gccxml 0.9

### Windows

1. Unzip LCGCMT\_58 into your local directory
2. Unzip Gaudi source code into your local directory
3. Copy in our version of ROOT v5.26.00a-gl1 into the external/ROOT directory
4. Copy in our version of Python 2.5.1 into the external/Python directory
5. Update external/LCGCMT\_58/LCG\_Configuration/cmt/requirements  
set ROOT\_config\_version to 5.26.00a-gl1  
set Python\_config\_version to 2.5.1-gl1
6. Update external/LCGCMT/LCGCMT\_58/LCG\_Interfaces/ROOT/cmt/requirements and comment out  
## apply\_pattern use\_optional pkg=Qt  
## apply\_pattern use\_optional pkg=dcache\_client  
## apply\_pattern use\_optional pkg=gfal  
and add:  
macro ROOT\_cppflags "" \  
target-winxp "-DROOT\_w32pragma "  
7. Update external/LCGCMT/LCGCMT\_58/LCG\_Settings/cmt/requirements  
Update LCG\_home to the top level location of the Gaudi build to something like:  
macro LCG\_home "\${myPathDir}/buildingGaudi\_v21r7/"  
Update LCG\_Releases to  
macro LCG\_releases "\${LCG\_home}/external"  
8. Update GaudiKernel's requirements file to remove use of CppUnit  
9. Update GaudiSvc's requirements file to remove use of CLHEP  
10. Set up to use CMT v1r20p20090520, for example:  
CMTBIN=VisualC  
CMTCONFIG=win32\_vc71\_dbg  
CMTPATH=C:\heather\gaudiUpgrade\GAUDI\GAUDI\_v21r7;C:\heather\gaudiUpgrade\external;C:\heather\gaudiUpgrade\external\LCGCMT\LCGCMT\_58;C:\heather\glst\tools  
CMTROOT=C:\heather\glst\tools\CMTv1r20p20090520 (make sure there is no space at the end of that definition!)  
CMTVERSION=v1r20p20090520  
11. Add CMT to PATH  
12. Execute vsvars32.bat in the Microsoft VC++ 7.1 distribution  
13. Gaudi uses CMT, so building goes something like this  
We choose to limit the Gaudi packages we build to (in this order): GaudiKernel, GaudiSvc, GaudiUtil, GaudiAlg, GaudiAud  
For each package, enter the cmt directory and do:  
cmt broadcast "cmt config"  
setup.bat  
nmake /f nmake  
14. Extra Notes.. a number of the boost includes had to be renamed from \*.hp to \*.hpp

### VC 2008 notes

1. GaudiKernel's Debugger.cpp required a modification where windows.h was included outside the Win namespace
2. GaudiSvc's manifest file for genconf.exe needs to be copied over into the InstallArea by hand

## redhat4-i686-32bit-gcc34

1. Obtain the source from /afs/slac/g/glast/ground/GLAST\_EXT/srcExtLibs/GAUDI\_v21r7-gl1\_src.tar.gz
2. Setup up the required Gaudi externals: AIDA, Boost, gccxml, python 2.5.1, ROOT v5.26.00-gl1, uuid, XercesC 2.8.0

# Building Gaudi v21r4

## Requirements

- CMT v1r20p20090520 available on SLAC Linux on /afs/slac/g/glast/users/heather/gaudi\_v21r4/CMT and through the download section of the [CM T web site](#).
- Source Code modified for GLAST to limit the externals required by Gaudi
- LCGCMT\_56c zip file of externals and interface CMT packages for Gaudi
- An updated version of ROOT v5.22.00e-gl1, the binaries are available in GLAST\_EXT at SLAC and the source is in compiled against gccxml 0.9

## redhat4-i686-32bit-gcc34

- Copy in our version of ROOT v5.22.00e-gl1 in the external/ROOT directory
- Copy in our version of python 2.5.1-gl1 into the external/python directory
- Update external/LCGCMT\_56c/LCG\_Configuration/cmt/requirements and set ROOT\_config\_version to 5.22.00e-gl1  
set Python\_config\_version to 2.5.1-gl1
- Update external/LCGCMT\_56c/LCG\_Interfaces/ROOT/cmt/requirements and comment out  
## apply\_pattern use\_optional pkg=Qt  
## apply\_pattern use\_optional pkg=dcache\_client  
## apply\_pattern use\_optional pkg=gfal  
and add:  
macro ROOT\_cppflags "" \  
    target-winxp "-DROOT\_w32pragma "  
• Update external/LCGCMT\_56c/LCG\_Settings/cmt/requirements  
Update LCG\_home to the top level location of the Gaudi build to something like:  
macro LCG\_home "\${myPathDir}/buildingGaudi\_v21r4/"  
Update LCG\_Releases to  
macro LCG\_releases "\${(LCG\_home)/external}"  
• Set up to use CMT v1r20p20090520, for example:  
setenv CMTBASE /afs/slac/g/glast/users/heather/gaudi\_v21r4/CMT  
setenv CMTVERSION v1r20p20090520  
source \$CMTBASE/\$CMTVERSION/mgr/setup.csh  
setenv CMTCONFIG slc4\_ia32\_gcc34  
setenv CMTPATH /nfs/slac/g/glast/users/glgound/heather/buildingGaudi\_v21r4/GAUDI/GAUDI\_v21r4:/nfs/slac/g/glast/users/glgound/heather/buildingGaudi\_v21r4/external:/nfs/slac/g/glast/users/glgound/heather/buildingGaudi\_v21r4/external/LCGCMT/LCGCMT\_56c:/afs/slac/g/glast/users/heather/gaudi\_v21r4  
• Gaudi uses CMT, so building goes something like this  
We choose to limit the Gaudi packages we build to (in this order): GaudiKernel, GaudiSvc, GaudiUtil, GaudiAlg, GaudiAud  
For each package, enter the cmt directory and do:  
cmt broadcast "cmt config"  
source setup.sh  
make

# Building Gaudi v18r1

The official Gaudi Home Page: <http://proj-gaudi.web.cern.ch/proj-gaudi/>

Our outdated GLAST Gaudi home page: <http://www-glast.slac.stanford.edu/software/gaudi/>

A good point of contact if having trouble: Pere Mato (Pere.Mato@cern.ch).

## Obtaining the source

The Gaudi home Page sees modification from time to time, especially on the timescale of our upgrades - so this information is as of June,2006.

Access the Releases page off the Gaudi home page: <http://proj-gaudi.web.cern.ch/proj-gaudi/releases/>

Click on the Doxygen link for the version you are interested in.

The Doxygen page for a particular release will provide a Download for the Gaudi source.

## Obtaining the external libraries that Gaudi uses

Accessing the Gaudi external libraries is separate from downloading Gaudi itself. Again, the Gaudi Home Page does see reorganization from time to time, this information is current as of June, 2006.

Access the Releases page off the Gaudi Home Page: <http://proj-gaudi.web.cern.ch/proj-gaudi/releases/>

Click on the Doxygen link for the version you are interested in.

The Doxygen page will include a link called "Related External Libraries"

For example for Gaudi v18r1: <http://proj-gaudi.web.cern.ch/proj-gaudi/releases/v18r1/doxygen/externaldocs.html> Note the version of LCGCMT, in this case LCGCMT\_40a.

Also note that the list of external libraries is not complete exhaustive! This page lists the highlights, but some of these externals depend on other external libraries that must be obtained.

This page is helpful in that it provides the version of all Gaudi external, however, I have found that it does not necessarily provide useful links for obtaining the binaries for all platforms. I found the following link much more helpful in that all the binaries are available in one location:

<http://service-spi.web.cern.ch/service-spi/external/distribution/>

This page does indeed include ALL externals you might require. For example, for Gaudi v18r1 we require:

AIDA 3.2.1  
BOOST 1.32  
CLHEP 1.9.2.2  
CppUnit 1.10.2  
expat 1.95.8  
gccxml\_0.6.0.patch3  
GSL 1.5  
pcre 4.4  
POOL 2.2.7  
Python 2.4.2  
SEAL 1.7.9  
uuid 1.38  
xerces 2.3.0

Yes, a long list, this will allow you to get started - after compiling Gaudi and determining which Gaudi packages we actually need and require, one can trim down the list of external libraries. Currently, GLAST Offline uses the following Gaudi packages: GaudiAud, GaudiAlg, GaudiKernel, GaudiPI, GaudiSvc. We are interested in using GaudiPython as well.

On the windows side - many libraries are only provided in debug mode. The Gaudi team does not have the time or resources to provide non-debug optimized versions - they use Windows as a development environment and not for production.

**In addition to downloading the libraries, also download the appropriate version of LCGCMT\_\*, this package acts as their IExternal package and is necessary to compile Gaudi.**

## Compilation

Unpack all the source and binaries you have downloaded.

Once LCGCMT has been unpacked you may have a directory tree that looks like:

LGCCMT/LCG\_CMT40a/LCG\_Interfaces - contains packages that behave like our IExternal package where the subpackages determine how to find the external libraries

LCGCMT/LCG\_CMT40a/LCG\_Settings - requirements file sets up global environment to find externals - in particular it defines the tags for the various operating systems and compilers

LCGCMT/LCG\_CMT40a/LCG\_Policy - like our GlastPolicy

There are some environment variables that should be set before compiling Gaudi:

CMTBIN  
CMTROOT  
CMTVERSION  
CMTPATH  
CMTCONFIG  
SITEROOT

In addition some of the requirements in LCGCMT should be modified:

Update LCG\_Settings/cmt/requirements:

set LCG\_home=C:\heather

set LCG\_releases - determines path to find POOL and SEAL

set LCG\_external - determines path to find other externals like AIDA... haven't experimented with making LCG\_external == LCG\_releases

Update LCG\_Interfaces/CLHEP/cmt/requirements

CLHEP\_version - version number of CLHEP for Gaudi v18r1 it is 1.9.2.2

CLHEP\_home - points to our CLHEP external example: \$(GLAST\_EXT)/CLHEP/\$(CLHEP\_native\_version)

Also add python.exe to the PATH if it isn't already.

Gaudi uses CMT, so building goes something like this - (Windows version in parentheses):

From within the Gaudi cmt directory

```
cmt broadcast "cmt config"
source setup.sh (setup.bat)
make (nmake /f nmake)
```

## Linux Notes

Might check if there is a .cmtrc file in your \$HOME directory to be sure it doesn't override your CMTPATH

## Windows Notes

Some examples env var settings:

```
set CMTBIN = VisualC
set CMTROOT=C:\heather\glast\tools\CMT\v1r16p20040901
set CMTVERSION=v1r16p20040901
set CMTPATH=C:\heather\gaudi\GAUDI\GAUDI_v18r1;C:\heather\gaudi\LCGCMT;C:\heather\gaudi\LCGCMT\LCGCMT_40a;C:\heather\glast\tools

set CMTCONFIG=win32_vc71_dbg
set SITEROOT=C:\heather
```

## Reorganization of includes and libraries

When Gaudi is done compiling, you will find the libraries distributed under each of its packages: GaudiAlg, GaudiAud, GaudiKernel, GaudiPI, GaudiSvc,... the libraries and includes may be reorganized under one common gaudi directory for distribution with the GLAST Offline software.