

Important Infrastructure Links

- [Building External Libraries](#)
- [CMT RM](#)
- [Core Meeting Minutes](#)
- [SCons RM](#)
- [System Tests](#)
- [Workbook](#)

Building External Libraries

A page with links with instructions for [building the externals](#) .

CMT RM

<https://confluence.slac.stanford.edu/display/SAS/CMT+Release+Manager>

Windows builds occur on: glast09, glast10, glast11, and glast-win03, as the "glast" user, where builds and externals reside on the V: drive
Note that the LSF password on windows must match the "glast" password. This is set via a command windows, using the lspasswd command.

RHEL4 builds occur on glastlnx14

Core Meeting Minutes

All of the core meeting minutes are located here: [/afs/slac/www/exp/glast/ground/software/minutes](#) and can be updated by those interested in doing so.

SCons RM

<https://confluence.slac.stanford.edu/display/SAS/SCons+Release+Manager>

Windows builds occur on glast-win04, as the "glast" user" where all data resides on the D: drive
Note that the LSF password on windows must match the "glast" password. This is set via a command windows, using the lspasswd command.

RHEL4 builds occur on glastlnx14

RHEL5-32 bldlnx06, bldlnx10, bldlnx11, bldlnx12 (this is the slow one I think)

RHEL5-64 available in buildq

RHEL6-64 available in buildq

Mac OS Snow Leopard bldmac02 and bldmac01

Mac OS Mountain Lion on PPA-PC90719

[See here for Jenkins](#)

To start up the Jenkins Windows Service:

My Computer (right click)->Manage->Services and Applications->Services->Jenkins Slave->Properties->Log On
Here we will also change the glast password, whenever the glast password is updated.

Managing the areas where the builds occur

[/nfs/farm/g/glast/u52/ReleaseManagerBuilds](#) - GR

[/nfs/farm/g/glast/u35/ReleaseManagerBuilds](#) - ST, TMineRelease

General Build Retention Rules

ScienceTools: All Releases are retained, HEADs for last 10 releases, the last 20 LATEST builds

Finding Lock Files in CVSROOT

`python /u/gl/jchiang/bin/find_cvs_lock_files.py`

Running RM steps by hand

The key bit of information you need is the build ID of the build you want to work with. This is just the number at the end of the URL in the RMII webpages. i.e. in this URL:

<http://glast-ground.slac.stanford.edu/rm2/buildPkg-II.jsp?buildId=20296>

the build ID is 20296.

Once you have this all you need to do is the following:

- 1) Log on to glast-win04
- 2) Open a command prompt
- 3) The command to run the RMII programs is

```
D:\ReleaseManager\tools\grits-cpp\bin\<program> --buildId <NNNNN>
```

where <program> is testBuild.exe or createReleaseBuild.exe (or any of the other ones) and <NNNNN> is the build ID number of the one you want to run.

That's it. It will start off the appropriate process. You just have to let it run and not kill your command prompt window until it finishes (and probably deal with any popup dialogs that appear :().

The one thing you should do, however, is actually create the LATCalibRoot environment variable for the glast account so that it is set properly before you run the testBuild.exe program. Right now it is getting passed in via LFS so as far as I know it isn't set up for interactive logins.

System Tests

<https://confluence.slac.stanford.edu/display/SYSTEST/System+test+overview+and+instructions>

Workbook

The entire workbook resides in CVS (CVSROOT=/nfs/slac/g/glast/ground/cvs), those interested in making modifications can do so. When a commit is made a "build" of the workbook occurs via [Hudson](#) and is posted to the [test site](#). Once we are satisfied and desire to post the updated contents to the [production version](#), we can run the `~glast/releaseWorkbook` script.

Note that the workbook test and prod versions seem to be symlinked at: `/afs/slac/www/exp/glast/wb/`, which points to `/nfs/farm/g/glast/u09/workbook/`

Adding ETags to force pages to reload when changed:

```
<meta http-equiv="ETag" content="$Revision: $" />
```

`/nfs/farm/g/glast/u52/ReleaseManagerBuilds`

`/nfs/farm/g/glast/u35/ReleaseManagerBuilds`