

# **The GSSC Standard Operating Environment**

Prepared by: Robert Schaefer, GSSC Software Manager

GSSC Standard Operating Environment.....	1
1. Purpose .....	3
2. Introduction.....	3
3. Accounts .....	3
3.1. Operational Requirements.....	3
3.2. Options.....	4
3.2.1. Group Account.....	4
3.2.2. Individual Accounts .....	4
4. Directory Structure .....	5
4.1. Executables .....	6
4.2. Libraries .....	6
4.3. Data Directories.....	6
4.3.1. Log Files.....	6
4.3.2. Database Files.....	6
5. Environment Variables .....	6
6. Configuration files.....	7
7. Special Environments .....	7
7.1. Database Clusters .....	7
7.2. Planning backup machine .....	7

# 1. Purpose

This document specifies the Operational Environment of GSSC programs for doing day-to-day operations, including data ingest, system monitoring, planning, and scheduling. This document specifies the environment in which programs will run including computer accounts, directory structure, environment variables, configuration files, log files and database files.

## 2. Introduction

The GLAST SSC is planning to create a center where operations are highly automated. To create such a system, the operating environment needs to be defined to facilitate daily operations and to allow efficient development of software. The operating environment must satisfy all of the requirements derived from those on the software. This document specifies the aspects of such an environment. We will call this the GSSC standard operating environment and it will be described below.

## 3. Accounts

The first order of business is to define the LHEA accounts that will run the programs. The needs of these accounts are described, followed by discussion of various options.

### 3.1. *Operational Requirements*

The requirements for the operational accounts whether there is one shared or multiple user based accounts must satisfy the following requirements.

- The accounts must support automated operation
  - Must be able to kill any hung process.
  - Must be able to restart a failed process
  - Must be able to monitor disk capacities, memory usage
  - Must be able to read and write data and auxiliary files
  - Must be able to access a MySQL database for logging.
- Must be able to monitor ingest pipeline progress
  - Must be able to bring up monitoring OPUS GUIs (note only the account that started OPUS can do this.)
  - Must be able to run Process manager
- All data, log files, configuration files, and ancillary data files must be able to be read and written by all operations accounts.
- Must be able to read all incoming mail sent to ssc operations
- Must be able to install, upgrade, and reconfigure a common set of software tools, including a suite of 3rd party tools.

## 3.2. Options

### 3.2.1. Group Account

A group shared account automatically fills all of these requirements. All processes, and files are owned by the group account and therefore there are no permission issues. Group accounts are forbidden by NASA security policy, so the group account would need to be waived by the higher ups.

### 3.2.2. Individual Accounts

Because the group account is against NASA policy, we explore some options for how the requirements can be satisfied.

- It would be best if these accounts would have to be separate from the person's other development/user account.
- All accounts would need the following:
  - be part of a group, e.g. gsscops
  - have the same home directory (not strictly required, but this would sure be nice!)
  - use the same shell, source the same resource file, same default group read and write permissions (umask)
- All accounts need permission to kill processes started by others in the group.
- All accounts need to be able to monitor OPUS

The first two needs are probably easily met in the LHEA framework. The last two are problematic – so we discuss them in some detail.

- 1) **All accounts need permission to kill processes started by others in the group.** Phil Newman has suggested that we use sudo privilege to be able to kill processes started by other users. To do this automatically, one would need to store the password, as I cannot believe there is an approved way to run sudo without a password. We can easily store the password encrypted – is that acceptable to NASA security?
- 2) **All accounts need to be able to monitor OPUS.** This seems more difficult. On the face of it, you cannot display the monitor without being the user. There may be some complicated way of configuring a script to throw the display to some other host involving xhosting the machine running OPUS locally.

The installation processes on third party software often does not allow one to specify group permissions on installed directories. Therefore the best that can be done without a group account is to assign different tool responsibilities to SSC personnel. This may be workable, although far from ideal.

## 4. Directory Structure

Basic structure are two top level directories full of locally produced files: /gsschome - where all software and configuration files live, and /gsscops - where all the data and log files live. A possible third top level directory is under consideration: /gsscdb, where databases would live. Finally a central place where we install external operational tools would be /gssc/local/ - so we can control versions of perl, top, gcc, etc.

```
/gsschome |
          | /bin |
          | /fedora-1 | gcc3.3?
          | /rhee | gcc3.5?
          | /lib |
          | /fedora-1 | gcc3.3?
          | /rhee | gcc3.5?
          | /perl
          | /etc
          | /doc
          | /man?

/gsscops |
         | /staging
         | /log
         | /opus
         | /ephem
         | /sched
         | /receive

/gsscdb |
        | /mysql
        | /backup

/gssc   | /local |
        | /bin
        | /lib
        | /inc
        | /doc
        | /man
```

These partitions will be commonly mounted by all SSC operations machines. One possibility is that the executables may be a locally mounted version of glitch:/builds and that the /gssc/local is glitch:/devtools.

## **4.1. Executables**

All executables binary and scripts will live in the /gsschome/bin directory. We are planning to only use one operating system, but given how long it takes to migrate a bunch of machines to a new OS, I think it is important to allow for the possibility that we will need to support more than one OS so our directories need to support multiple O.Ss. Also given the speed with which gcc versions are being released, we may also have to support multiple versions of gcc during a transition period. It is not clear what we need to do here – we could have different subdirectories of bin/arch/ and lib/arch/ for these executables.

## **4.2. Libraries**

The libraries includes compiled libraries and perl modules. If history is any guide, the perl modules will remain fairly stable even with changing versions of perl. The ingest pipeline will not likely use compiled perl modules (wrappers for compiled libraries) and is unlikely to change when development switches gcc versions.

## **4.3. Data Directories**

All data will be kept in the /gsscops partition. This partition will be in constant flux as data moves in the SSC and out to databases.

### **4.3.1. Log Files**

A central area for log files is provides in /gsscops/log

### **4.3.2. Database Files.**

All database files will be kept under the /gsscdb directory.

## **5. Environment Variables**

Operations accounts will source a script provided which will set the environment variables for GSSC programs. There will be two versions of this script  
.ENVIRONMENT.csh (to set the environment for command line programs)  
.ENVIRONMENT.sh (to set the environment for cron jobs.)  
which live in the accounts home directory.

The following environment variables will be used. The environment variables will then facilitate easy testing – just change the variables to fit the test environment.

- GSSCHOME – the root of software and config files (will point to /gsschome)
- GSSCOPS – the root of operational data directories.
- ARCH – the operating system – how to get fedora version
- PATH – usual paths but will prepend \$GSSCHOME/bin/\$ARCH and /gssc/local/bin
- LD\_LIBRARY\_PATH – usual LD\_LIBRARY\_PATH with \$GSSCHOME/lib/\$ARCH and /gssc/local/lib prepended
- PERLLIB - \$GSSCHOME/lib/perl/

## 6. Configuration files

All configuration files will be kept in \$GSSCHOME/etc/. These files will be backed up daily. When possible information such as directories and changeable data information should be kept in configuration files, so that we only need change the files, not the code.

## 7. Special Environments

### 7.1. *Database Clusters*

The data base clusters are on a different subnet and run in a different environment. So they will have a local environment.

### 7.2. *Planning backup machine*

As much as possible this machine will keep the same directory/file structure as the main operations environment – but out of necessity these will be copies of the main GSSC environment.