

# ISOC Software

## Overview

The LICOS system is designed to record, package and transfer LAT data in a flight-like manner. This design goal is achieved by incorporating data-handling software developed for the Instrument Science Operations Center. This software implements the conventions for data transfer documented in the *GLAST Operations Data Products ICD*.

## External Dependencies

The ISOC software, like LICOS, is implemented primarily as a set of Python modules, and therefore has the same dependencies on the Python interpreter as LICOS. As of 30 August 2005, both packages are targeted to run on Python 2.4.1. For the Mobile Rack system, we assume that Python 2.4.1 is installed under /usr/local. The following is the complete list of supporting packages, in addition to Python, required for proper operation of the ISOC software. I've indicated which dependencies, to the best of my knowledge, are shared between LICOS and the ISOC software.

## RedHat Enterprise Linux 4 Packages

The following packages from the RHEL4 distribution must be installed:

- postgresql-7.4.7-2.RHEL4.1
- postgresql-server-7.4.7-2.RHEL4.1
- postgresql-devel-7.4.7-2.RHEL4.1
- postgresql-libs-7.4.7-2.RHEL4.1
- jpackage-utils-1.6.0-2jpp\_3rh

## Locally-built packages

The following packages are built locally, generally from source downloaded from the Web, but using a customized RPM specfile and potentially with some patching of the source distro during the build process. All the specfiles for these packages are maintained in the SPECS module of the Online CVS repository at :ext:centaurusa.slac.stanford.edu:/nfs/slac/g/glast/online/cvsroot. For Web-available tarballs, the source distro location is documented in the corresponding specfile. For distros that are not directly web-accessible (e.g. FASTCopy, Sun JDK), the source distributions are maintained in the SOURCES module of the Online CVS repository. All locally-created distribution patches are also stored in the SOURCES module.

Note that the build-numbers of these packages may change if the custom installed configurations of the software is modified in some way. The latest builds of these RPMS should be found in /nfs/online/rpms/ on the bastion host.

Package	Source	LICOS?	Patched?	Description
cfitsio-2.510-3.glast	Web	Y	N	FITS interface library
Numeric-23.8-1.glast	Web	Y	N	Array class for Python
pycfitsio-2.510-3.glast	CVS	Y	N	Python interface to CFITSIO
4Suite-Base-1.0b1-1.glast 4Suite-Xml-1.0b1-1.glast	Web	N	N	Python XML libraries
java-1.5.0-sun-1.5.0.04-1jpp java-1.5.0-sun-devel-1.5.0.04-1jpp java-1.5.0-sun-fonts-1.5.0.04-1jpp java-1.5.0-sun-plugin-1.5.0.04-1jpp	CVS	N	N	Sun JDK
jakarta-tomcat-5.5.11-1.glast	CVS	N	Y	Web-Application container
psycopg-2.0b3-1.glast	Web	N	N	Python interface module for PostgreSQL
FASTCopy-2.6-1.glast	CVS	N	Y	Secure data-transfer software

## Building Distribution RPM's

The ISOC software is packaged as two RPMS: The ISOC RPM contains the core data-handling code, and the ISOC-Web RPM contains a JSP-based Web application that is used to monitor the data-transfer system. Since I envision that this code will evolve much more rapidly than the underlying dependencies, I've set up a two-stage process for generating RPM's from the source code in CVS. Step 1 produces a source tarball based on a specified CVS tag in the repository; the tarball includes a specfile with the appropriate version information. Step 2 uses this tarball to produce an installable RPM, using the rpmbuild command's ability to build from a specfile included in a tarball.

## Creating the source distribution

The following commands retrieve the necessary helper tools from the ISOC CVS repository and create source distributions for the ISOC and ISOC-Web packages:

```
$ export CVSROOT=:ext:centaurusa.slac.stanford.edu:/nfs/slac/g/glast/online/cvsroot
$ export CVS_RSH=ssh
$ cvs co -d DistTools ISOC/DistTools
$ cd DistTools/ISOC
$ ./makedist.sh HEAD 1.2.0 1.glast
$ cd ../ISOC-Web
$ ./makedist.sh HEAD 1.0.0 1.glast
$ cd ..
```

## Creating the installable RPM's

The following commands will use the source distributions created above to make installable RPM's of the software. These operations assume that the user has set up a private RPM build area by configuring their `~/rpmmacros` file appropriately.

```
$ rpmbuild -ta ISOC/ISOC-1.2.0.tar.gz
$ rpmbuild -ta ISOC-Web/ISOC-Web-1.0.0.tar.gz
```

The above will create RPMs with filenames of the form:

```
ISOC-1.2.0-1.glast.i386.rpm
ISOC-Web-1.0.0-1.glast.i386.rpm
```

Once the RPM's are built, they can be installed with ``rpm -ivh``.

## Configuring the Installed Software

Once the ISOC packages and their dependencies are installed on a fresh system, some installation-specific setup of both the underlying services and the ISOC software itself is necessary.

## Starting daemons and services

Issue the following commands to bring up the supporting services and ensure that they will start up when the system is rebooted:

```
$ su
# /sbin/service postgresql start
# /sbin/service tomcat55 start
# /sbin/service flogidc start
# /sbin/service xinetd restart
# /sbin/chkconfig postgresql on
# /sbin/chkconfig tomcat55 on
# /sbin/chkconfig flogidc on
# /sbin/chkconfig xinetd on
```

## Configure PostgreSQL

The FASTCopy automation software (module `ISOC.FASTCopyDB.FASTCopyDB`) uses a set of database tables to perform and track incoming and outgoing FASTCopy transfers. On a fresh installation, a PostgreSQL user and database must be created, and the PostgreSQL server must be configured to allow local and network access to the database. Note that creating a PostgreSQL user does not create a system user account. The `'createuser'` command will prompt for a password used to authenticate the user to the database. Examine the file `/usr/local/lib/python2.4/site-packages/ISOC/sitedep/sitedep-default.ini`, and use the string specified in the `'password'` entry in the `'dsn'` parameter of the `'FASTCopyDB'` stanza.

```
$ su
# su postgres
$ createuser -A -D -P GLAST_ISOC
$ createdb -O GLAST_ISOC GLAST_ISOC
$ exit
# exit
$
```

Once the PostgreSQL user and database are created, edit the file `/var/lib/pgsql/data/postgresql.conf` and change the line `"tcpip_socket = false"` near the top of the file to `"tcpip_socket = true"`. Edit the file `/var/lib/pgsql/data/pg_hba.conf` and add the following two lines immediately above the line beginning `"local all all..."`:

```
local    GLAST_ISOC      GLAST_ISOC                               md5
host     GLAST_ISOC      GLAST_ISOC      0.0.0.0 0.0.0.0 md5
```

After these configuration files are modified, restart the database server using the following commands:

```
$ su
# /sbin/service postgresql restart
```

## Load the FASTCopyDB schema

After PostgreSQL is reconfigured and restarted, use the following commands to load the FASTCopyDB schema into the GLAST\_ISOC database as the GLAST\_ISOC user (note that the `psql` command will prompt for the GLAST\_ISOC user's password):

```
$ cd /usr/local/lib/python2.4/site-packages/ISOC/FASTCopyDB/dbase
$ psql -U GLAST_ISOC -f build_postgresql.sql GLAST_ISOC
```

Once the basic schema is loaded, the name of the installation host must be added to a table in the database. Determine the fully-qualified hostname of the machine, e.g. `"lat-hobbit4.slac.stanford.edu"` and execute the following command:

```
$ psql -U GLAST_ISOC -d GLAST_ISOC -c "insert into fcopy_submithost values( (select max(submithost_pk)+1 from fcopy_submithost), 'lat-hobbit4.slac.stanford.edu');"
```

Similarly, the Unix usernames of any accounts from which outgoing files will be submitted must also be added to a table in the database. Determine the desired list of accounts, e.g. `"ric jim selim"` and execute the following command:

```
$ for u in ric jim selim ; do
> psql -U GLAST_ISOC -d GLAST_ISOC -c "insert into fcopy_submitter values( (select max(submitter_pk)+1 from fcopy_submitter), '$u');"
```

## Enable the FASTCopy-automation cron jobs

The ISOC RPM installation creates a user account named `'glastops'` under which the FASTCopyDB software and the underlying FASTCopy operations take place. Cron jobs executing under this account are used to scan the database tables for work to be performed. Although the installation adds these entries to the `glastops` crontab file, they are initially commented out. Use the following command to bring up the `glastops` crontab in the `vi` editor; uncomment all the job entries and save and close the file to activate the FASTCopy cron jobs:

```
$ su
# VISUAL=vi crontab -u glastops -e
# exit
$
```

