INTEGRAL Off-Line Scientific Analysis
Installation Guide

Issue 11.2, March 2022

INTEGRAL Science Data Centre - Chemin d?Écogia 16 – CH-1290 Versoix
https://www.isdc.unige.ch/integral

CONTENTS

1 Welcome

This document describes the installation of the INTEGRAL Off-line Scientific Analysis (OSA) version 11.2 on Linux and the use of the docker image.

Requirements:

As from OSA11, OSA depends on HEASOFT for rebinning the IBIS/ISGRI and JEM-X response matrices. It is thus required to have it installed an initialized (see https://heasarc.nasa.gov/lheasoft/). **Due to non-backwards compatible changes in `rbnrmf`, we require Heasoft at most 6.27. OSA is compatible with Heasoft 6.24, 6.25, 6.26, and 6.27, NOT COMPATIBLE with Heasoft 6.28 onward.**

OSA is composed of the following packages:

Off-line Scientific Analysis Software (OSA_SW)
Instrument Characteristics (OSA_IC)
"High-energy" Catalogue (OSA_CAT)
Test Observations, which consist of a set of data and related scripts (OSA_TESTDATA)

In order to successfully analyse INTEGRAL data, you will have to download and install at least the three packages

OSA_SW
OSA_IC
OSA_CAT

via the following URL

http://www.isdc.unige.ch/integral/analysis#Software

and retrieve observation data from the INTEGRAL archive

http://www.isdc.unige.ch/integral/archive

The download and use of OSA_TESTDATA is optional. It is meant to check the correct installation of OSA by running test scripts on a small data set.

OSA software comes as a binary package or may be installed from the source code.

We strongly recommend that you first install and test the binary package for your operating system(s).

You can use the binary packages to compile and link your own software with OSA libraries. If you have problems doing so, please read Appendix A on 'Troubleshooting'.

This document is part of the INTEGRAL analysis user documentation available at http://www.isdc.unige.ch/integral/analysis. In particular, detailed information about the software is provided in the instrument specific user manuals.

If you have any problems installing OSA please consult the 'Frequently Asked Questions' at http://www.isdc.unige.ch/integral/analysis#FAQ to check for an answer to your question or send an E-mail to the INTEGRAL help-desk at inthelp@sciops.esa.int. See the appendix on 'Troubleshooting' for information you should provide to help us solving the problem you have.

2 Installing the Software

The INTEGRAL OSA software is available via the following URL

http://www.isdc.unige.ch/integral/analysis#Software

Packages containing the OSA binaries for the following Linux distributions are provided for your convenience:

- CentOS 5 – x86_32
- CentOS 6 – x86_64
- CentOS 7 – x86_64
- Ubuntu 16.04 – x86_64
- Ubuntu 20.04 – x86_64

The OSA software requires as run-time library the `libgfortran.so.3` and the `libc.so.6` libraries. These library should be installed before the OSA software can be used. The libc.so library is most likely already installed with recent operating systems. The `libgfortran.so` library can be received by various packages, depending on the Linux distribution. For Ubuntu 20.04 – x86_64, we distribute `libgfortran.so.3` together with OSA, as it is not anymore available using the most common installation methods.

If you are running an operating system for which no binary package is available, you can use the docker image, which is giving full portability to OSA (Section 6). Testers have successfully used the Ubuntu version on Open Suse 42.2 and 42.3. You can also download and install the source code package. Please see Appendix B for the detailed system requirements and instructions to install the OSA software from the source code.

We strongly recommend that you first install and possibly test the binary package for your operating system(s) or use the docker image.

Linux users:
Once you have downloaded the binary package for a Linux operating system,

1. Move it to a directory to which you have write access and
2. unpack the downloaded tar-file using e.g. the following command: `tar xfzv "Binary-file"`

Resulting from the above command(s) you will find the directory `osa11` containing all the OSA software. This directory is referred to as `ISDC_ENV` in the INTEGRAL OSA user documentation.

 You will have to set the environment variable `ISDC_ENV` to the full pathname of this directory to use the just installed OSA software. Details on setting up the run-time environment can be found in the cookbook sections of the instrument specific OSA user manuals.

Mac OS X users: Docker is a system that enables to run a software within its native environment without modification of the users' operating system. The only requirement it to install the docker server and download the docker image for a specific task. We provide OSA and Heasoft in a docker image built on Linux CentOS7.

Mac OS X users are invited to use docker distribution, and are referred to 6 for full instructions.

For any platform, you need to install the 'Instrument Characteristics' and the "Reference Catalogue". Their installation is described in the following two sections. Then, you need to fetch data and perform the analysis.

3 Installing the Instrument Characteristics

Since OSA 8.0, the IC data is no more provided as a pre-packaged tar-file nor via a script to download the files.

1. Define an archive directory to store the instrument characteristics and the reference catalogues. Then execute one of the following commands:

   `setenv REP_BASE_PROD __archive_directory__` -- for `csh` family
   `REP_BASE_PROD=__archive_directory__ ; export REP_BASE_PROD` -- for `sh` family

   (If the `setenv` commands fail with a message like: 'setenv: command not found' or 'setenv: not found' please read theappendix on 'Troubleshooting'.)
2. with the following command you will retrieve the IC data from the ISDC server :

   ```
   rsync -Lzrtv isdcarc.unige.ch::arc/FTP/arc_distr/ic_tree/prod/ $REP_BASE_PROD
   ```

A total disk space of some 3+ GBytes is needed to install the full IC data containing all INTERAL instruments.

 The file names of the IC data are significant and must not be changed. Otherwise internal references in the IC data will be broken.

To learn more about 'ISDC compliant Data Repositories' (`REP_BASE_PROD`) and 'Instrument Characteristics', please refer to the OSA User Manuals (especially the 'Introduction to the INTEGRAL Data Analysis') at the following URL

http://www.isdc.unige.ch/integral/analysis

4 Installing the Reference Catalogues

The INTEGRAL reference catalogue and the OMC reference catalogue are available via the following URL
http://www.isdc.unige.ch/integral/analysis#Software.
A total disk space of 85 MB is needed to install the two catalogues.
Once you have downloaded the OSA_CAT package,

1. cd to a directory to which you have 'write' access.
   Typically, the directory where you unpack the reference and OMC catalogue is as well used for the installation of the instruments characteristics, see above.
2. Unpack the downloaded tar-file using e.g. the following command

   ```
   gzip -dc osa_cat-VV.tar.gz | tar xvf -
   ```

   where `VV` is the available version number, e.g. 43.

To learn more about 'ISDC compliant Data Repositories', 'OSA Catalogues' and DOLs, please refer to the OSA User Manuals (especially the 'Introduction to the INTEGRAL Data Analysis') at the following URL
http://www.isdc.unige.ch/integral/analysis

5 Installing and Using the Test Data

Now you may want to test the installed OSA software. To be able to do so, OSA comes with a test data packages running an analysis on small data sets from Crab observations for 5 instruments. Those packages are available via the following URL:

http://www.isdc.unige.ch/integral/analysis#Software

A total disk space of about 3 GB is needed to install the test data packages. After running the test scripts, the disk space required grows to about 5 GB.

5.1 Installing the Test Data

Once you have downloaded the test data package,

1. Create a directory to which you have write access and `cd` to this directory.
2. Unpack the downloaded tar-file using e.g. the following command
   `gzip -dc osa_testdata-11.0.tar.gz | tar xvf -`
3. `cd` to the `testdata` sub-directory that was created via the above command. This directory will be referred to as `REP_BASE_PROD` for the rest of this section.
4. Create links to the already downloaded OSA_IC package in the archive directory.

   ```
   ln -s __archive_directory__/ic ic
   mkdir idx          - if the idx directory does not exist yet
   cd idx
   ln -s __archive_directory__/idx/ic ic
   ```

5.2 Running the Test Scripts

Once a consistent set of the OSA software, IC, CAT and the test data packages has been successfully installed, the available test scripts can be used to verify the correct installation, set-up and functioning of the INTEGRAL Off-line Scientific Analysis.

To do that, the following command sequence should be followed. Here, we report only the commands for the csh-family, but they can be easily ported for the sh-family by substituing `setenv VAR VALUE` with `setenv VAR=VALUE`.

1. `setenv ISDC_ENV __directory_of_OSA_SW_installation__`
   the "_directory_of_OSA_SW_installation" can be, e.g., "/opt/osa/"
2. `setenv ISDC_REF_CAT __DOL_of_the_reference_catalogue__`

   e.g. "__directory_of_cat_installation__/cat/hec/gnrl_refr_cat_0043.fits[1]"  The double-quotes (") are important if you are using a shell of the csh-family. Otherwise the shell will try to interpret the [1] as a regular expression.
3. `setenv ISDC_OMC_CAT __DOL_of_the_OMC_catalogue__`

   e.g. "__directory_of_cat_installation__/cat/omc/omc_refr_cat_0005.fits[1]"  The double-quotes (") are again important (see above).
4. `setenv REP_BASE_PROD __TEST_DATA_install_dir__/testdata`
5. `source $ISDC_ENV/bin/isdc_init_env.csh`

   for sh-family, use `. $ISDC_ENV/bin/isdc_init_env.sh`  The 'source' or the leading '.' are important!
6. `cd __TEST_DATA_install_dir__/testdata`
7. Make a symbolic link to the `ic` and `idx` locations on your system
8. `ln -s _my_location/ic`
9. `ln -s _my_location/idx`
10. `cd ..`

11. It is always a good idea to remove old parameter files that are stored in your user pfiles directory – usually the pfiles subdirectory in your home directory. You actually only need to remove the entry-level parameter files, ibis_science_analysis.par, jemx_science_analysis.par, omc_science_analysis.par or spi_science_analysis.par
12. `make test INSTR=`*`instr`*
    instr should be either `ibis`, `picsit`, `jemx`, `spi` or `omc` to perform the test for a specific instrument.[1]
13. `make dircmp INSTR=`*`instr`* – optional. See below for details.
    instr should be either `ibis`, `picsit`, `jemx`, `spi` or `omc` to verify the test for a specific instrument.

The "`make test INSTR=`*`instr`*" command will run the data analysis for the test observation. If you have the FTOOL fdiff available on your system, you may run as well the "`make dircmp INSTR=instr`" command. A comparison of the output data of the current test run with the reference output data will be performed. The reference output data is included in the OSA test data package in the directory 'instr_outref'. It was generated running OSA on a 64-bit Linux platform. The `make test` command should exit with a return code of 0. Users may check the return code by typing echo '$?' immediately after the `make test` or `make dircmp` command.

If `make test` terminated with an error and you cannot figure out the problem yourself, please consult the 'Frequently Asked Questions' at http://www.isdc.unige.ch/integral/analysis#FAQ to check for a solution to your problem or send an E-mail to the INTEGRAL help-desk at inthelp@sciops.esa.int. See the appendix on 'Troubleshooting' for information you should provide to help us solving the problem you have.

 We have introduced a test also for running trough the docker container. First make sure that docker is installed on your system (see the following section). Then, you need to define just two enviromental variables

```
setenv REP_BASE_PROD __TEST_DATA_install_dir__/testdata
setenv CURRENT_IC __IC_TREE_install_dir__
```

Then, you can run the scripts: `make test INSTR=`*`instr`*`docker`
instr should be either `ibis`, `jemx`, `spi` or `omc` to verify the test for a specific instrument.

[1] The test runs of the different instruments are independent of each other and can be launched in any order.

6 OSA through Docker

To ensure portability of OSA with evolving operating systems, we provide an OSA docker image, suitable for any platform for which docker is available. We can imagine Docker as a tool in which one can ship an OS with the SW. In our case, it allows us to run OSA in a Linux environment, independently from the user's OS. As Mac OS-X platforms would require significant software development for portability, this has become not feasible with the available resources. Therefore, we discontinued our effort to provide Mac OS binary images. Here, we provide a brief introduction to the use of docker for OSA.

6.1 General installation of OSA docker image to run the analysis

First of all, it is necessary to **install Docker on your platform**. We do not provide support on this, instructions can be retrieved from the docker project to which we refer. Instructions are specific to your OS distribution (see https://docs.docker.com/install/). For a quick use, we suggest to read the first two paragraphs below (Graphical interface for Linux and Mac, Downloading the OSA docker image) and then just to Section 6.2.

Graphical interface, special instructions for Mac users

Note that these instructions could not be tested on the latest versions after 2020
The interaction with the graphical environment is platform dependent and needs to be properly set-up. For Linux you need to make sure that X11 is installed and that you are in a graphical session.

For OSX, the following instructions have been developed by CERN for the "root" program and are also valid for OSA, which uses "root" for its graphical interface (we refer for details an updates to the page https://hub.docker.com/r/rootproject/root-ubuntu16/

1. Make sure that XQuarz is installed. If not, install it from here: https://www.xquartz.org/
2. Open XQuartz, go to XQuartz, Preferences, select the Security tab, and tick the box "Allow connections from network clients". Then, quit XQuartz.
3. Open Terminal and issue the following command:
   `grep inet | awk '$1=="inet" {print $2}')`
   This will grab your IP address on the local network.
4. Run `echo $ip` to make sure it was successful. If nothing is displayed, replace "en0" by "en1" or a higher number in the above command.
5. Try again until it works.
6. When you have successfully defined the variable `ip` to store your IP address, issue the command: `xhost + $ip`
   This will start XQuartz and whitelist your local IP address. A message should be displayed saying that the IP address has been "added to the access control list".

6.2 Recommended method

We provide a script to wrap commands around docker to run OSA, this is available at https://gitlab.astro.unige.ch/savchenk/osa-docker/blob/master/osa-docker.sh Note that his is part of a public git repository that you can download as

```
git clone https://gitlab.astro.unige.ch/savchenk/osa-docker.git
```

In the `osa_docker.sh` script, you can define the local data location and the local archive location (they can be the same) by setting the environmental variables

```
export CURRENT_IC=/isdc/arc/rev_3
export REP_BASE_PROD=/isdc/arc/rev_3/
```

Note that **symlinks will not work** as the filesystem in docker is different from the parent one.

Then, you can create the science window list in your local directory and OSA as, for example

```
./osa-docker.sh "og_create scw.list IBIS test ./"
./osa-docker.sh "cd obs/test;ibis_science_analysis
```

The graphical interface in the script needs to be adapted to your platform (Linux or Mac OS-X), by setting the display variable for Mac OS-X while executing the script as (in bash):

```
DISPLAY=$ip:0 ./osa-docker.sh "cd obs/test; ibis_science_analysis"
```

(you might need `$ip:1` or higher on some Mac OS-X)

This script will take care of the docker setup steps described below.

6.3 Step-by-step usage

Downloading the OSA docker image

Download the Docker image from the public Docker Hub to which ISDC has uploaded the software. (This requires 2.6 Gb of disk space and in Internet connection.)

From the Terminal command line type:

```
docker pull integralsw/osa
```

(Note that you can also specify a version with `docker pull integralsw/osa:11.1`, where the number after the colon, here `11.1`, is a tag to the version. There are also `11.0` and `10.2` docker images if you want to download those versions. Omitting the tag will result in downloading the `latest` version, which is the recommended way.

Docker images will be stored in a system directory. Images can be listed from the command line with the instruction:

```
docker image ls
```

(see documentation for details, e.g., https://docs.docker.com/engine/reference/commandline/images/)

The command:

```
docker rmi ‘‘Image-name’’
```

will remove an image.

Defining the arguments and options

Any OSA command can be executed in the docker container after some settings.

The container acts as a separate OS in which the OSA initialization is performed internally with the script “/init.sh”. However, we have to provide through our local OS input data and a place to write the results. This can be done in the way explained below.

Any OSA command inside the docker image is run from the command line as:

```
docker run [arguments] integralsw/osa command
```

Let’s first create temporary directories to mount the `home` and `pfiles` to be visible within the docker image. This will avoid “polluting” our working directory with unnecessary outputs ($$ is a variable that stores the shell’s process ID number[2]):

```
mkdir -pv /tmp/osa-home-$$
mkdir -pv /tmp/osa-home-$$-pfiles
```

These directories will be identified and mounted using the *docker options*

```
-v /tmp/osa-home-$$-pfiles:/pfiles
-v /tmp/osa-home-$$:/home/integral
```

[2]At the terminal, enter `echo $$`, and then `ps`. You will see that the active shell in which you are writing these commands has the process number given by `$$`.

It is necessary to specify the user name in the docker image to be the same as in the native OS, which is done with the options:

```
--rm -it --user $(id -u) --entrypoint=''
```

The the graphical interface options are defined as:

`-e DISPLAY=$DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix` on Linux;
`-e DISPLAY=$ip:0 -v /tmp/.X11-unix:/tmp/.X11-unix` on OS-X.

Uning Docker,it is easy relax the requirement to use an OSA compliant directory, as described in https://www.isdc.unige.ch/integral/download/osa/doc/11.1/osa_um_intro/index.html. We can run the software somewhere and keep in separated places the ic_tree and the archive. Following the instructions in Section 3, we can download the ic_tree in a folder which we can refer to with the shell variable `CURRENT_IC`. Following Section 7, we can store the archive in a place referred to with the shell variable `REP_BASE_PROD` These directories will contain the folders `ic`,`idx` (`CURRENT_IC`) and `scw`,`aux` (`REP_BASE_PROD`). We should can mount this directory with the options:

```
-v $REP_BASE_PROD/scw:/data/scw:ro \
-v $REP_BASE_PROD/aux:/data/aux:ro \
-v $CURRENT_IC/ic:/data/ic:ro \
-v $CURRENT_IC/idx:/data/idx:ro \
```

The output data will be contained in the working dir.
In conclusion, these are the arguments and options we need to run OSA within the docker.

```
-e DISPLAY=$DISPLAY \
-v /tmp/.X11-unix:/tmp/.X11-unix \
-v $PWD:/home/integral \
-v $REP_BASE_PROD/scw:/data/scw:ro \
-v $REP_BASE_PROD/aux:/data/aux:ro \
-v $CURRENT_IC/ic:/data/ic:ro \
-v $CURRENT_IC/idx:/data/idx:ro \
--rm -it --user $(id -u) -entrypoint=''\
```

In OS-X (for a graphical interface, remember to issue `xhost + $ip` in your shell before running OSA) and change the first line with `-e DISPLAY=$ip:0`.

Running OSA Commands

We are now ready to work on the <command> to run OSA tasks. Since we need to define some environment variables, it is better to run a shell that will 1) define variables, 2) initialize OSA, and 3) cd into the working directory.

Our command will then be:

```
'bash -c
export HOME_OVERRRIDE=/home/integral;
.  /init.sh;
cd \$HOME;
<our OSA commands>'
```

Catalogs are shipped with the docker image. However, it is possible to use custom catalogs. What is needed is to place them in a folder mounted in the docker image.

We suggest to place the catalogs in the standard locations in your local file

```
$PWD/cat/my_gnrl_refr_cat.fits
```

and

```
$PWD/cat/my_omc_refr_cat.fits
```

And use them in the call within docker as:

```
'bash -c \
export HOME_OVERRRIDE=/home/integral;\
.  /init.sh;\
cd \$HOME;\
ISDC_REF_CAT=/home/integral/cat/my_gnrl_refr_cat.fits;\
ISDC_OMC_CAT=/home/integral/cat/my_omc_refr_cat.fits;\
<our OSA commands>'
```

We are ready to run `og_create` program to build our OSA observation group. We use IBIS as a example and assume that we have stored our list of science windows in the file `list_scw.txt`.

The OSA command is

```
og_create idxSwg=list_scw.txt ogid=my_og \
baseDir=$REP_BASE_PROD instrument=IBIS
```

However, we have to prepend all the necessary arguments for docker to run properly.

```
docker run \
-e DISPLAY=$DISPLAY \
-v /tmp/.X11-unix:/tmp/.X11-unix \
-v $PWD:/home/integral \
-v $REP_BASE_PROD/scw:/data/scw:ro \
-v $REP_BASE_PROD/aux:/data/aux:ro \
-v $CURRENT_IC/ic:/data/ic:ro \
-v $CURRENT_IC/idx:/data/idx:ro \
--rm -it --user $(id -u) -entrypoint=''\
'bash -c
export HOME_OVERRRIDE=/home/integral;
.  /init.sh;
cd \$HOME;
og_create idxSwg=list_scw.txt ogid=my_og \
baseDir=$REP_BASE_PROD instrument=IBIS'
```

This will create a directory `obs/my_og`, with the observation group for OSA. It is now possible to run `ibis_science_analysis` via the command:

```
docker run \
-e DISPLAY=$DISPLAY \
-v /tmp/.X11-unix:/tmp/.X11-unix \
-v $PWD:/home/integral \
-v $REP_BASE_PROD/scw:/data/scw:ro \
-v $REP_BASE_PROD/aux:/data/aux:ro \
-v $CURRENT_IC/ic:/data/ic:ro \
-v $CURRENT_IC/idx:/data/idx:ro \
--rm -it --user $(id -u) -entrypoint=''\
'bash -c
export HOME_OVERRRIDE=/home/integral;
.  /init.sh;
cd \$HOME/obs/my_og;
ibis_science_analysis'
```

The graphical interface opens and you can run the `ibis_science_analysis` as explained in the relative the user manual (in OS-X, remember to issue `xhost+$ip` in the shell before running the commands and set

the `DISPLAY` variable accordingly). With these commands, we moved into our observation group within the docker container to run `ibis_science_analysis`. Results will be locally stored in `$PWD/obs/my_og`.

To reduce data from other instruments, you should change the call to `og_create` appropriately and then call `jemx_science_analysis`, `spi_science_analysis`, or `omc_science_analysis`.

7 Getting and Installing Integral Data

7.1 Public Data

Requests for public INTEGRAL data can be made via the ISDC archive browser at the following URL: http://www.isdc.unige.ch/integral/archive Once you have selected the data you want to download, you will receive an e-mail with detailed instructions on how to download the public INTEGRAL data.

7.2 Private Data

Once the data for your observation has been fully processed, the PI will receive an e-mail with detailed instructions on how to download and install the private INTEGRAL data. The PI will receive first the message for the NRT data and then for the consolidated version. Since AO12, all the INTEGRAL data are available straightforwardly for sources that were not assigned to a scientist, with the exception of the observations assigned to PIs from the Russian federation. To download these data, it is necessary to visit the page: http://www.isdc.unige.ch/integral/-----------#NRT_CONS_data and agree to the "terms and conditions". Starting from AO13, all the consolidated data are made available, regardless of the PI provenance. For example, scientists interested in a new transient appearing on a private observation can download and use all the available INTEGRAL data for their publication.

Appendix A Troubleshooting

A.1 Problems with the `setenv` command

If the `setenv` command fails with a message like: '`setenv: command not found`' or '`setenv: not found`', you are very likely not using a command interpreter (shell) of a C-Shell flavour (e.g. csh or tcsh). Please replace the command `setenv my_variable my_value` by the following command sequence `my_variable=my_value ; export my_variable`

 No blank characters ' ' must be present in the `my_variable=my_value` command. Otherwise you will get an error message like '`my variable not found`'.

A.2 Problems with the `source` command

If the source command fails with a message like: `'source: not found'` or you get many error messages indication that the command actually failed, you are very likely not using a command interpreter (shell) of a C-Shell flavour (e.g. csh or tcsh). Please replace the command '`source my_script`' by the following command '`. my_script`'

 You really must type the '.' followed by a blank character before the name of the script. Otherwise the setting of the environment variables inside the script will not be kept after the script terminated.

A.3 Problems linking your own software with OSA libraries

If you are using a binary installation and you are having problems to link your own software with OSA libraries the following command sequence might help:

```
cd $ISDC_ENV ; mv lib lib_orig ; ln -s libalt lib
```

If you still have problems, you probably need to compile the OSA software from the source code and link your software using the newly built libraries. However, we suggest to use docker containers.

A.4 Reporting problems to the INTEGRAL help desk

For an efficient trouble-shooting, please include the following information in your E-mail to the help-desk inthelp@sciops.esa.int.

1. the version of the OSA software package
2. the name and version of the operating system. Type `uname -a` to retrieve this information. Linux users should in addition provide the contents of file /proc/version.
3. the version of the OSA_IC package used
4. the version of the OSA_CAT package used
5. if you are having a problem running the test scripts:
   (a) in the terminal window where the make test was run, execute the `env` or `printenv` command and provide the output to the ISDC.
   (b) the contents of the following files make_test_failed.log or make_dircmp_failed.log depending on whether the '`make test`' or '`make dircmp`' failed.
6. if you are building the OSA software from the source code:
   (a) the name and version of the compiler(s). For the GNU compilers, type '`gcc -v`' and '`g++ -V`'
   (b) the name and version of the make program. Type '`make -v`' or '`gmake -v`' depending on the name of the GNU make on your system.

Appendix B Installing the Software from Source Code

If you are running an operating system for which no binary package is available, or you cannot use the binary packages for another reason, do not hesitate to contact us via the INTEGRAL help-desk at inthelp@sciops.esa.int for assistance on using docker or singularity containers.

We strongly recommend that you try to use the docker image, which is routinely maintained or the binary packages.

Should you really need to build and install software for another operating system, we provide here some basic guidelines.

B.1 Supported Platforms

The OSA software was checked by ISDC to correctly compile, install and run on the following Linux distributions:

- CentOS 5.11 64 bit gcc 4.4
- CentOS 6.9 64 bit gcc 4.4
- CentOS 7.5 64 bit gcc 4.4
- Ubuntu 16.04 64 bit gcc 4.4
- Ubuntu 20.04 64 bit gcc 4.4

The binary OSA software packages that we provide at the ISDC are generated for these platforms.

In general, the OSA software will also compile and run on a variety of other Linux platforms. As we at the ISDC do not necessarily have access to those platforms, we unfortunately can only offer limited support for those platforms.

B.2 Third Party Software

Before compiling and installing the OSA SW from the source code you have to make sure that the following packages are installed:

1. GNU make version 3.79.1 (or higher)

    You really need this version of GNU make for all platforms. Most Linux distributions will already include the correct make. Earlier versions of GNU make are known not to work.
2. ROOT version 5.34.34
   available via the ISDC website at URL `http://www.isdc.unige.ch/integral/analysis#Software` in the section on 'Download OSA for Developers'.
3. gcc and gfortran version 4.4. This version works, later and earlier versions might not.

 Since OSA software version 3.0 you can choose between an installation with and without ROOT. If you choose to install without ROOT, you will NOT benefit from all OSA functionalities, i.e. GUI support is not available. To learn more about how to install OSA without ROOT please see section 'Setting up the Environment'.

For other operating systems it is recommended to install ROOT from the source code. If you want to download ROOT as a binary package, you must make sure that the compiler used to compile ROOT is identical to the one you are using for the installation of the OSA SW. Otherwise, please download the ROOT source code package and compile and install it yourself. Using different compilers may result in serious problems with your X-system.

B.3 Hardware Requirements

Disk Space : to build and install the OSA software from source code a total of some 1.7 GByte of disk space is needed – depending on the operating system and compiler used.

After the successful installation from the source code you may reduce the disk space needed by OSA software by executing '`make distclean`' in the same directory where you executed '`make install`'. This will reduce the amount of disk space needed to some 890 MB. You may additionally remove the source code directories (support-sw, analysis-sw, contrib-sw). This will reduce the disk space needed to some 790 MB.

You will need approximately 10 GB of disk space to hold the data from one day of INTEGRAL observations.

Memory/CPU : to actually run the software on observation data you will need at least some 500 MB of memory per simultaneously active user on your system. Although the software will run on slower systems, it is recommended to use systems with at least 500 MHz processors.

B.4 Getting the Software

The INTEGRAL OSA software source code package (osa10.2-source.tar.gz) is available via the following URL

http://www.isdc.unige.ch/integral/osa/current/developers

Once you have downloaded the source code package

1. `cd` to a directory to which you have write access and
2. unpack the downloaded tar-file using e.g. the following command:
   `gzip -dc osa11.0-source.tar.gz | tar xvf -`

Resulting from the above commands you will find the directory `osa11` containing all the OSA source code. For a typical installation, this directory is referred to as `ISDC_ENV` in the INTEGRAL OSA user documentation.

B.5 Setting up the Environment

This section describes the setting of the installation relevant environment variables. It is not intended to provide a description of the environment that is needed to run the software. The installation environment will not be sufficient to run the software. Please see the instrument specific cookbooks for a detailed description on how to set-up the environment to actually run the OSA software.

Compiler

Make sure that your compiler's bin and lib directories are contained in environment variables PATH and LD_LIBRARY_PATH respectively. A way to check this, is to use the '`which`' command followed by the name of your compiler. If the compiler is not found, type the following commands:

```
setenv PATH ‘‘__path_to_your_compiler__/bin:$PATH’’
setenv LD_LIBRARY_PATH ‘‘__path_to_your_compiler__/lib:$LD_LIBRARY_PATH’’
```

If one or both of the above commands fail with a message like: 'LD_LIBRARY_PATH: Undefined variable', omit the '`:$PATH`' or '`:$LD_LIBRARY_PATH`' part in the commands respectively.

If the `setenv` commands fail with a message like: '`setenv: command not found`' or '`setenv: not found`' please read the appendix on 'Troubleshooting'.

In addition, you have to make sure that all compiler relevant environment variables are correctly set up. Depending on the compiler and OS used, those variables are:

Linux (including Ubuntu) using GNU C/C++ and Fortran Compilers

```
setenv CC "gcc -Df2cFortran"
setenv CXX "g++ -Df2cFortran"
setenv F90 gfortran
```

The ISDC default setting on Linux is to switch the code optimiser on (option `-O2`).

You must have a FORTRAN90 compiler installed on your system and compatible with gcc (e.g., `gfortran`). Otherwise the build will fail.

If the `setenv` commands fail with a message like: '`setenv: command not found`' or '`setenv: not found`' please read the appendix on 'Troubleshooting'.

ROOT

You can choose between an installation with and without ROOT. If you choose to install without ROOT, you will NOT benefit from all OSA functionalities, i.e. GUI support is not available. Once you have installed OSA software without ROOT and would like to use the GUIs, you will have to re-install OSA software from the source code.

If you choose NOT to install with ROOT, you have to make sure that environment variable ROOTSYS is NOT set. Otherwise you will run into problems during the OSA software compilation.

If you choose to install with ROOT, make sure that environment variable ROOTSYS is set to where your ROOT installation is located. The ROOTSYS environment variable should typically point to a directory with at least the following sub-directories: `bin, etc, icons, include` and `lib`. Make sure that your ROOT's bin and lib directories are contained in environment variables PATH and LD_LIBRARY_PATH respectively. A way to check that is to actually execute the '`root`' command and verify the version of root. To exit root, please type '.q'. If root is not found, type the following commands:

```
setenv PATH ‘‘$ROOTSYS/bin:$PATH’’
setenv LD_LIBRARY_PATH ‘‘$ROOTSYS/lib:$LD_LIBRARY_PATH’’
```

If one or both of the above commands fail with a message like: 'LD_LIBRARY_PATH: Undefined variable', omit the '`:$PATH`' or '`:$LD_LIBRARY_PATH`' part in the commands respectively.

If the `setenv` commands fail with a message like: '`setenv: command not found`' or '`setenv: not found`' please read the appendix on 'Troubleshooting'.

ISDC_ENV

Decide where the ISDC software package should be installed. This is controlled via the environment variable ISDC_ENV. You may set the value of ISDC_ENV to any location where you have write permission. It is however recommended to set ISDC_ENV to the directory where the source code of the OSA software package is located.

When installing on a multi-platform system, you might find it useful to have a common source code tree for the various binary installations you maintain. This can be achieved by setting the ISDC_ENV environment variable to e.g. `__osa_src_rep__/LocalLinux` for an installation under Linux.

Make sure that `$ISDC ENV/bin` is contained in the environment variable PATH – even if this directory does not yet exist at this stage of the installation. If `$ISDC ENV/bin` is not included, execute the following command: `setenv PATH "$ISDC_ENV/bin:$PATH"`. If the `setenv` command fails, please read the appendix on 'Troubleshooting'.

B.6 Configure, Build and Install

When the installation environment is successfully set up you may proceed to configure, build and install the software package. This is done by executing the following command sequence in the directory where you unpacked the OSA software source package:

configure

1. cd to the directory where you have unpacked the downloaded tar-file.
2. If you want to install OSA software without ROOT execute the command
   `support-sw/makefiles/ac_stuff/configure --without-cern-root`
3. If you want to install OSA software with ROOT execute the command
   `support-sw/makefiles/ac_stuff/configure`

These commands will create all system dependent Makefiles that are needed to build the software. Please type '`support-sw/makefiles/ac_stuff/configure --help`' to learn about the configure options.

 The setting of all environment variables mentioned in section 'Setting up the Environment' has to be done prior to executing the configure command. If, after executing the configure command, you change your mind about the value of an environment variable, you must first execute '`make distclean`' and then redo the configuration step.

 When installing on a multi-platform system and you decided to share the source code tree, you must execute '`make distclean`' and then redo the configuration step for each platform.

 Make sure that the environment variable `MAKEFLAGS` is not set.

If the configure step terminated with an error and you cannot figure out the problem yourself, please consult the 'Frequently Asked Questions' at http://www.isdc.unige.ch/integral/analysis#FAQ to check for a solution to your problem or send an E-mail to the INTEGRAL help-desk at inthelp@sciops.esa.int. For an efficient trouble-shooting, please see appendix 'Troubleshooting' for information you should provide to help us solving the problem you have.

Build and install

As for construction of OSA. you need to directly type

```
make install
```

 You really have to type '`make install`'. A simple `make` will fail.

This will compile all relevant source files, create the library files, the F90 modules and the C/C++ and F90 executables. The installation terminates by moving the library, the C/C++ header files, the F90 modules, the parameter files and the user manuals to the correct location.

If everything went ok, you should see the following message:

```
************ everything compiled and installed successfully ***********
executables are installed in :  $ISDC_ENV/bin
parameter files are installed in :  $ISDC_ENV/pfiles
F90 modules are installed in :  $ISDC_ENV/f90mod
documentation, help, .txt files are installed in :  $ISDC_ENV/help
C/C++ include .h files are installed in :  $ISDC_ENV/include
.a, .so library files are installed in :  $ISDC_ENV/lib
script files are installed in :  $ISDC_ENV/share/default_scripts
other shared files are installed in :  $ISDC_ENV/share
FITS file template files are installed in :  $ISDC_ENV/templates
```

If '`make install`' terminated with an error and you cannot figure out the problem yourself, please consult the 'Frequently Asked Questions' at http://www.isdc.unige.ch/integral/analysis#FAQto check for a solution to your problem or send an E-mail to the INTEGRAL help-desk at inthelp@sciops.esa.int. For an efficient trouble-shooting, please see appendix 'Troubleshooting' for information you should provide to help us solving the problem you have.

Once the installation is successfully terminated you may recover some disk space by removing temporary compilation products. This is done by executing '`make distclean`'. You may even recover more disk space and remove the entire source code tree by executing '`rm -rf support-sw analysis-sw other-sw`' in OSA source code directory.

Now OSA software is fully installed and is ready to be used. Please refer to the OSA documentation available at http://www.isdc.unige.ch/integral/analysis for details on how to run the OSA software. The 'Introduction to INTEGRAL Data Analysis' and the cookbook section of the instrument specific user manual will be of particular use for you.

Appendix C Document Change History

11.2	Updated for OSA 11.2 (minor).
11.1	Updated for OSA 11.1 with Ubuntu 20.04. Moved to TEX.
11.0	Updated for OSA 11 with a section on docker usage. Removed references to Mac OS.
10.2	Updated for OSA 10.2 (use of same release number). Addition of an Ubuntu binary and update of Mac OS X versions: for Mavericks (10.9), Yosemite (10.10), and El Capitan (10.11), instead of Lion (10.7) and Mountain Lion (10.8).
2.3	Updated for OSA 10.1. Mac OS X Mountain Lion (10.8) instead of Snow Leopard (10.6) and Mac OS X distribution as `tar.gz' file instead of `dmg.zip'.
2.2	Updated for OSA 10.0. The testdata are one package for all instruments
2.1.2	Updated for OSA 9.0. Removed references to Solaris as this is not supported anymore
2.1.1	Changed version of GNRL_REFR_CAT from 28 to 30
2.1	Updated for OSA 8.0, especially the IC download description
2.0.6	Updated description for CAT and IC installation, especially the symbolic linking part for the IC data.
2.0.5	Highlighted more the OMC catalogue, catalogue version number fixed and removed a couple of typos.
2.0.4	Adapted to OSA 7.0
2.0.3	changed e-mail address of INTEGRAL help desk
2.0.2	Adapted to OSA 6.0
2.0.1	Adapted to OSA 5.1
2.0	Complete re-write