RCS-related FAQ

This page reports the Frequently Asked Questions encountered during the RCS implementation activities.

Table of content

RPW data general questions

Which standards applied to the RPW data?

The standards for Solar Orbiter science data are defined in https://issues.cosmos.esa.int/solarorbiterwiki/display/SOSP/Metadata+Definition+for+Solar+Orbiter+Science+Data.

Especially, this document provides the conventions in terms of file format, naming and data versioning.

All RPW science data must comply these standards.

N.B. Solar Orbiter science data standards partially rely on the ISTP/IACG CDF guidelines (see https://spdf.gsfc.nasa.gov/sp_use_of_cdf.html).

Where can I find Solar Orbiter SPICE kernels?

All SOC-provided Solar Orbiter SPICE kernels retrieved by the ROC are made available in https://rpw.lesia.obspm.fr/roc/data/private/solo/soc/spice/ (restricted access).

For more details about:

- SOC-provided ancillary data products: https://issues.cosmos.esa.int/solarorbiterwiki/display/SOSP/SOC-Provided+Ancillary+Data+For+Solar+Orbiter
- Solar Orbiter SPICE kernels: https://issues.cosmos.esa.int/solarorbiterwiki/display/SOSP/Solar+Orbiter+SPICE+Kernels
- SPICE software: https://naif.jpl.nasa.gov/naif/index.html

CDF-related questions

What is a "skeleton table"?

A "skeleton table" is an ASCII format file used to generate a "master" binary CDF file.

The "master" binary CDF file can be then used as a template to produce series of CDF files with the same structure.

See also "How to generate a "master" CDF binary file?".

How to generate a "master" CDF binary file?

The CDF generation principle for RPW is described here.

For more information about the CDF and skeleton/master mechanism, visit http://cdf.gsfc.nasa.gov/.

How to generate a skeleton or master CDF file from an Excel format file?

The MASER4PY Python package allows to generate skeleton table or Master CDF from an Excel 2007 format file (.xlsx).

More details can be found in https://pypi.org/project/maser4py/.

How to check RPW CDF compliance against Solar Orbiter data standards?

There is no ESA-provided software to check CDF compliance against ESAC Solar Orbiter data standards (SOL-SGS-TN-0009).

However the ROC has started to develop a light Python program to check RPW/Solar Orbiter data compliance.

A first version of this tool can be found in https://gitlab.obspm.fr/ROC/DataPool/-/blob/master/SOLO/RPW/CDF/Tools/check_rpw_l1l2.py (restricted access).

The program must be run under Python 3.6+ and requires spacepy module. Enter 'python check_rpw_l1l2.py –help' to display help message.

> This tool is not fully tested yet. Use at your own risk!

How to check RPW CDF compliance against NASA ISTP/SPDF data standards?

The NASA-provided software SKTeditor (https://spdf.gsfc.nasa.gov/skteditor/) can be used to check compliance of CDF file against NASA ISTP/SPDF data standards.

An instance of the SKTeditor is installed in the roc2-dev.obspm.fr server to run CDF check using the dedicated command line interface (see "How to run the NASA SKTeditor CLI in the roc2-dev.obspm.fr sever?")

How to run the NASA SKTeditor CLI in the roc2-dev.obspm.fr server?

From a terminal logged in roc2-dev.obspm.fr server:

First set up the environment by entering:

```
source /usr/local/lib/cdf/current/bin/definitions.B

export CLASSPATH=.:${CDF_JAVA}/classes/cdfjava.jar:${CDF_JAVA}/cdftools/CDFToolsDriver.jar:${CDF_JAVA}/cdfml
/cdfml.jar:"$CLASSPATH"

export LD_LIBRARY_PATH=.:${CDF_JAVA}/lib:${CDF_LIB}:"$LD_LIBRARY_PATH"

export CLASSPATH=/usr/local/cdf/skteditor:"$CLASSPATH"
```

N.B.

- Instructions above can also be inserted into a .bashrc file to be run at start
- Instructions above are given assuming a BASH shell user, but other shells can also be used (see CDF install manual for more details).

Then, to check the compliance of a given <cdf_filename>, enter:

```
java -cp /usr/local/lib/cdf/skteditor/spdfjavaClasses.jar gsfc.spdf.istp.tools.CDFCheck <cdf_filename>
```

RCS software and product delivery questions

How a RCS software must be delivered to ROC?

The mechanism to deliver RCS software to the ROC is presented in the "ROC Engineering Guidelines for External Users" (REGU) document (see ROC-GEN-SYS-NTT-00019-LES, available in ROC Documents).

In summary:

- RCS software must be delivered with at least the source code and the executable "ready-to-be-used" on the roc-dev server. The executable can be a script or a binary file.
- RCS software must be delivered with the up-to-date documentation (user manual, software specification and design)
- RCS software must be delivered with the up-to-date descriptor file (see RCS ICD for more details, available in ROC Documents)
- RCS software must be "pushed" on the "master" branch of the dedicated project in the https://gitlab.obspm.fr/ROC/RCS group. There is one project by RCS
- Every RCS software delivered must be tagged on Git with a version (use "git tag" command for this)
- The ROC team must be informed each time a new version of the software is "pushed"

How can I deliver my RCS test data package to ROC ?

The RCS test data package is required by the ROC Team to check that a given version of the RCS works as expected. Every version of the RCS software must hence be delivered with its test data package.

The RCS test data package must be delivered using the sftp-lesia.obspm.fr server. Table below gives for each team the user account and target folder to be used to deliver the test data package.

Team in charge	RCS name	SFTP user account	Target folder in the SFTP server
BIAS	BICAS	solbia	/obs/solbia/testdata
LFR	LFR_CALBUT	sollfr	/obs/sollfr/testdata
SCM	SCMCAL	solscm	/obs/solscm/testdata
TDS	TDS_CALBA	soltds	/obs/soltds/testdata
THR	THR_CALBAR	solthr	/obs/solthr/testdata

INFO

- Target folders must not be used for long term file storage. Every file dropped by teams in the folder will be moved to a long-term storage directory.
- Access to the SFTP server is only possible for authorized accounts and machines. See with the ROC Team for details.

How can I deliver my RPW calibration table (RCT) file to ROC ?

The RPW calibration table (RCT) files must be delivered as CDF files in the dedicated SFTP server at LESIA. Table below gives for each team the user account and target folder to be used to deliver the calibration table files.

Team in charge	RCS name	SFTP user account	Target folder in the SFTP server
BIAS	BICAS	solbia	/obs/solbia/cal
LFR	LFR_CALBUT	sollfr	/obs/sollfr/cal
SCM	SCMCAL	solscm	/obs/solscm/cal
TDS	TDS_CALBA	soltds	/obs/soltds/cal
THR	THR_CALBAR	solthr	/obs/solthr/cal

INFO

- Target folders must not be used for long term file storage. Every file dropped by teams in the folder will be moved to a long-term storage directory.
- Access to the SFTP server is only possible for authorized accounts and machines. See with the ROC Team for details.
- CDF skeletons of the RCT must be delivered as any other CDF skeleton (see "How to deliver a CDF skeletons file to ROC?")

How to deliver a CDF skeletons file to ROC?

The mechanism to deliver CDF skeletons to the ROC is presented in the "ROC Engineering Guidelines for External Users" (REGU) document (see ROC-GEN-SYS-NTT-00019-LES, available in ROC Documents).

In summary:

- CDF skeletons must be delivered as a skeleton table in the ASCII format
- CDF skeletons must be "pushed" by teams in the "SOLO/RPW/CDF/Skeleton" folder of the ROC "DataPool" Git respository (https://gitlab.obspm.fr/ROC/DataPool)
- CDF skeletons can be only "pushed" by teams in the "rcs" branch of the ROC "DataPool" Git respository

IMPORTANT

Every "commit" should be commented to have a traceability of the changes.

How to deliver RPW L3 CDF files to ROC?

RPW L3 CDF files are not generated by the ROC at LESIA, but they must be delivered to the ROC which will be in charge of validating the files, sharing them in the RPW private/public data servers and deliverying them to Solar Orbiter Archive (SOAR).

There are few rules to follow when deliverying files:

- Make sure that RPW L3 CDF files are compliant with RPW/Solar Orbiter data standards (add link)
- L3 CDF files must be dropped in the dedicated folder in the ROC SFTP server at LESIA (contact ROC team for details)
- The descriptor field of the CDF filename must be appended with the "-cdag" suffix (e.g., "solo_L3_rpw-tnr-fp-cdag_20230102_V01.cdf")

If one these rules are not fullfiled, then the ROC will reject the delivery of RPW L3 CDF files.

Warning

- The ROC will not manage the data version of the L3 CDF delivered. It is the responsibility of people in charge to ensure the consistency and the traceability of the CDF data version. Especially any file already delivred (i.e., with the same data version) will be automatically rejected by the ROC.

What is the software environment used by the ROC at LESIA to run RCS?

Information about the ROC software environment can be found in Software environment

Warning

Make sure the RCS delivered to LESIA are consistent with the ROC software environment.
If it is not the case, the ROC team might reject the delivery.

RCS-ROC interface-related questions

What is the descriptor file?

The descriptor file (or descriptor) is JSON format file used by the ROC pipelines to identify and call the RCS in an autonomous way.

A RCS delivered without or with a badly formatted descriptor cannot be run by the ROC pipelines.

See RCS ICD for more details about the descriptor, available in ROC Documents.

How can be sure that my descriptor file is well formatted?

The ROC team will always performed verifications of your descriptor just after each RCS delivery.

A JSON format file, as for XML, can be validated against a schema. A JSON schema is following the specifications from http://json-schema.org/draft-04/schema. Dedicated documentation can be found at https://spacetelescope.github.io/understanding-json-schema.

A copy of the RCS descriptor JSON file “cawa-schema.json” can be found in the RCS folder of the https://gitlab.obspm.fr/ROC/DataPool Git repository. Teams are free to download and use this file to verify that their descriptor is compliant.

This schema is also showed in the appendices of the RCS ICD (available in ROC Documents).

N.B. The ROC team plans to make available an instance of its RCS Interface Validation Pipeline (RIVP) on the roc-dev.obspm.fr server. The RIVP is the light version of the ROC pipeline, which can be used to test the execution of RCS in conditions closed to the production.

How can be sure that my RCS software is compliant with the ROC pipeline calling interface?

The ROC pipeline calling interface is described in the RCS ICD (available in ROC Documents).

The RIVP can also be used to test the compliance (see How can be sure that my descriptor file is well formatted? above).

ROC Gitlab-related questions

How to get a local copy of the ROC "DataPool" Git repository?

The RCS skeleton CDF files must deliver to the ROC team using Git.

The "DataPool" Git respository to use is hosted on the ROC Gitlab server. Before working with this repository, be sure that you have the right access permissions (if not, please contact the ROC team using the roc.support mailing list). To know how to access to this repository, see next question.

To get a local copy of the "DataPool" repository (i.e., clone), enter the following command from a terminal:

> git clone https://gitlab.obspm.fr/ROC/DataPool (HTTPS)

or

> git clone git@gitlab.obspm.fr:ROC/DataPool (SSH)

If everything goes right, you should have a local folder "DataPool" in your system.

> **WARNING**
>
> By default, Git will download the "master" branch only. This branch is protected and contain the skeleton/master CDF files actually used by the ROC.
>
> In order to submit their skeleton CDF files, the RCS teams must use the dedicated "rcs" branch.

To switch to the "rcs" branch, enter the following commands in the given order :

1. git checkout -b rcs
2. git branch --set-upstream-to=origin/rcs rcs
3. git pull

The first command creates the local branch "rcs" and switches to this new branch. The second one synchronizes the local branch with the remote branch "origin/rcs" on the ROC Gitlab server. The third command downloads (i.e., merges) data from the remote branch "origin/rcs" into the local branch "rcs.

NOTES:

- All of the "git ..." command must be run from the DataPool/ folder
- The "git pull" command should always be run at first, in order to ensure that the remote and local branches are always synchronized
- To switch from a branch to another, use the "git checkout *name_of_the_branch*" command.

How to access to the ROC "DataPool" Git repository?

1. Connect a first time to the gitlab server https://gitlab.obspm.fr, using your LDAP login/pass (i.e., the same used to access to the roc-dev server.)
2. Send an email to the roc.support list when it is done, and the ROC team will give you the access to the repository
3. You should then be able to see the https://gitlab.obspm.fr/ROC/DataPool repository

> **INFO**
>
> - The step 1 is only required at the first connexion to register your account into the gitlab server.
> - You will have only the read/write access to the "rcs" branch of the repository ; this branch must be used to deliver your CDF skeletons. The « master » branch is used by the ROC and cannot be changed by teams.
> - You might use ssh keys to access to the repository. In this case, just send your public key to the roc.support list and the ROC team will add it to the authorized key list.

RCS issue management questions

Where I can submit issues in case of problems with RPW files produced by ROC?

Any issue related to RPW data produced by ROC (i.e., L1/HK/L1R/L2/ANC, etc.) must be submitted using the dedicated issue management board in the RCS gitlab projects.

The table below lists the URLs of the issue board for each RCS:

RCS name	Issue tracker
BICAS	https://gitlab.obspm.fr/ROC/RCS/BICAS/issues
LFR_CALBUT	https://gitlab.obspm.fr/ROC/RCS/LFR_CALBUT/issues
SCMCAL	https://gitlab.obspm.fr/ROC/RCS/SCMCAL/issues
TDS_CALBA	https://gitlab.obspm.fr/ROC/RCS/TDS_CALBA/issues
THR_CALBAR	https://gitlab.obspm.fr/ROC/RCS/THR_CALBAR/issues

Issues related to L1/HK must be assigned to Xavier Bonnin.
Issues related to L1R/L2 must be assigned to Quynh Nhu Nguyen.

Full list of all issues can be read in https://gitlab.obspm.fr/groups/ROC/RCS/-/issues.

The ROC Team can also use the RCS issue boards to submit to the team in charge any problem encountered with RCS.

How can I access to the RCS issue management board?

The access is restricted and must be requested to roc.support sympa mailing list (i.e., roc dot support at sympa dot obspm dot fr).

RPW Calibration Tables-related questions

Miscellaneous questions

Where I can find the list of acronyms relative to the ROC project?

The list of acronyms are available here: Acronyms and Definitions

How can I use maser4py program on the roc2-dev.obspm.fr server?

The maser4py is installed on the roc2-dev.obspm.fr server.

To load it, run the command:

```
source /opt/roc/virtualenvs/maser4py/bin/activate
```

To check if the program works you can enter:

```
maser --version
```

Which will return the version of the program.

To get help:

```
maser --help
```

The maser4py module requires NASA CDF library to be installed to work. You can install and load a local copy on your home from https://cdf.gsfc.nasa.gov/. Or you can load the default version installed on the roc2-dev.obspm.fr server, by running the command:

source /opt/roc/lib/cdf/current/bin/definition.B

(Assuming BASH shell).