# maser-idl Documentation

*Release 0.1.0*

**X.Bonnin, B.Cecconi, Q.N.Nguyen**

December 18, 2015

Contents:

# INTRODUCTION

The MASER IDL library (MASER-IDL) contains IDL routines to deal with services and data provided in the framework of the MASER portal.

For more information about MASER, please visit: http://maser.lesia.obspm.fr/

# INSTALLATION

## 2.1 System Requirements

In order to install MASER-IDL, make sure to have IDL 8.3 or higher available on your system.

The following IDL routine libraries shall be installed and callable from IDL:

- *CDAWlib*

MASER-IDL has been successfuly tested on the following Operating Systems:

- Mac OS X 10.10, 10.11
- Debian Jessie 8.2

In order to use the "cdf" routines, the NASA CDF software distribution shall be installed and configured on your system. Especially, make sure that the directory containing the CDF binary executables is on your $PATH, and the $CDF_LIB env. var. is set. Visit http://cdf.gsfc.nasa.gov/ to learn more about the CDF format and software.

## 2.2 How to get MASER-IDL

To download MASER-IDL, enter the following command from a terminal:

```
git clone git://git.renater.fr/maser/maser-idl.git
```

Make sure to have Git (https://git-scm.com/) installed on your system.

If everything goes right, you should have a new local "maser-idl" directory created on your disk.

## 2.3 How to set up MASER-IDL

To set up the library on your system, enter the following command from the "maser-idl" directory:

```
bash scripts/setup_maser-idl.sh
```

or

```
tcsh scripts/setup_maser-idl.csh
```

This will add the MASER-IDL routine directories into your $IDL_PATH env. variable.

If you have an issue during installation, please read the "Troubleshooting" section for help.

## 2.4 How to run MASER-IDL

From IDL, you can compile all of the MASER-IDL routines calling the *compile_maser-idl.pro* IDL batch file in the scripts/ sub-directory:

```
@compile_maser-idl
```

Be sure that all of the required external routine libraries are already compiled.

# OVERVIEW

The MASER-IDL library is organized as follows:

> **maser/**
>
> > **data/**
> >
> > > **stereo/**  Module to handle the STEREO NASA mission data.
> > >
> > > **wind/**  Module to handle the Wind NASA mission data.
> >
> > **services/**
> >
> > > **helio/**  Module to get and plot the HELIO Virtual Observatory data.
> >
> > **utils/**
> >
> > > **cdf/**  Module to handle the NASA Common Data Format (CDF).

Each module is described in details in the next sections.

In order to work, MASER-IDL relies on additional files and directories:

> **bin/**  Used to store the MASER-IDL binary files
>
> **data/**  Directory containing support data
>
> **doc/**  Contains the MASER-IDL documentation
>
> **scripts/**  Script files to set up and run the MASER-IDL library

# THE *UTILS/CDF* MODULE

The *utils/cdf* module contains the following IDL programs:

- *make_cdf.pro*
- *rcdf.pro*

For more information about the CDF format, please visit http://cdf.gsfc.nasa.gov/.

## 4.1 The *make_cdf* program

The *make_cdf* procedure allows users to produce an CDF format file, from a list of given zVariables and attributes, and providing the corresponding "master" CDF file.

### 4.1.1 The calling sequence

The full calling sequence of *make_cdf* is:

```
make_cdf,master_cdf,output_cdf,variables,vattributes=vattributes,gattributes=gattributes,/VERBOSE
```

, where:

**master_cdf**  is the path to the master CDF file to use

**output_cdf**  is the path of the output CDF file to create

**variables**  is a IDL structure containing the zVariables to be updated in the output CDF file

**vattributes**  is an optional IDL structure containing the attribute entries for the zVariables.

**gattributes**  is an optional IDL structure containing the global attribute entries

**/VERBOSE**  is a boolean input keyword to activate the verbose mode

## 4.2 The *rcdf* program

The *rcdf* function is a CDF format light reader. It returns a IDL structure containing the variable data - records and attributes - of an input CDF format file. The Global attributes can also be retrieved using a dedicated output argument.

Only zVariables data can be read.

## 4.2.1 The calling sequence

The full calling sequence of *rcdf* is:

> zVariables = rcdf(cdf_file, gatts=gattrs,/VERBOSE)

, where:

**cdf_file**  is the path of the input CDF format file to read

**gattrs**  can be used to return a structure containing the list of the CDF global attributes

> **/VERBOSE**          is a boolean input keyword to activate the verbose mode

## 4.2.2 The outputs

The *zVariables* output is a IDL structure containing one tag by zVariable. Each tag stores a sub-structure with the following items:

**id**  The index of the zVariable

**datatype**  The CDF data type of zVariable

**numelem**  The "Number Elements" as defined in the CDF format

**recvar**  The "Record Variance" as defined in the CDF format

**dimvar**  The "Dimension Variables" as defined in the CDF format

**dim**  The zVariable dimension sizes

**data**  The zVariable data (not returned if the zVariable has no record)

**vattributes**  Structure containing the zVariable attributes entries

# THE *SERVICES/HELIO* MODULE

The *services/helio* module provides routines to get and handle the HELIO Virtual Observatory.

## 5.1 The *hfc* directory

The *hfc* directory concerns the HELIO Heliophysics Feature Catalogue (HFC) service.

### 5.1.1 The hfc idl client

The HFC IDL client can be used to retrieve and plot the data from the HFC.

# THE *DATA*/*WIND* MODULE

The *data/wind* module directory provides routines to read the Wind spacecraft data.

## 6.1 The *waves* directory

The *waves* directory concerns Wind / Waves experiment data.

### 6.1.1 The *lesia* routines

Here are routines that can be used to read Waves data files produced by the LESIA:

- read_wind_waves_hres read the Wind/Waves L2 high resolution data (big endian binary format)
- read_wind_waves_60s read the Wind/Waves L2 60 sec. averaged data (big endian binary format)

# INDICES AND TABLES

- genindex
- modindex
- search