Installation of MasTer

**(FOR Mac OS DISTROS)**

*Installation guide by:*

*N. d’Oreye, D. Smittarello, M. Jaspard, G. Celli – ECGS*

*B. Smets – RMCA*

*Version 4.0: October 19 2022*

[Preliminary information 2](#_Toc117083939)

[0) Prepare the work 5](#_Toc117083940)

[1) Installing some external components 6](#_Toc117083941)

[2) Installing required tools 8](#_Toc117083942)

[3) Installing MasTer 13](#_Toc117083943)

[4) Directory structure for data, auxiliary data and results 16](#_Toc117083944)

[5) Check installation 17](#_Toc117083945)

[6) Operating with cron jobs 20](#_Toc117083946)

## Preliminary information

**MasTer tool** is a set of mostly shell scripts aiming at processing automatically a large number of interferometric pairs and feeding and running the **MSBAS** processor [Samsonov and d’Oreye, 2012, 2017; Samsonov et al., 2017, 2020] in order to obtain the desired deformation maps and time series in vertical and horizontal components. Geocoded amplitude, coherence, interferometric phase and deformation maps are computed using the **MasTer Engine**, a command line InSAR processor derived from the Centre Spatial de Liege (CSL) InSAR Suite (CIS) [Derauw, 1999; Derauw et al, 2019].

This guide explains the installation procedure for **MasTer (MasTer tool, MasTer Engine, MSBAS)**, on a Mac computer. It does not replace the manual file provided with the software distribution (MassProcess\_notes\_Distro\_4.0.DOCX), but rather provide much more details to allow any user with less-advanced knowledge of the Mac environment to properly install the software suite.

The installation procedure has been last tested with the **MasTerEngine\_v20220929** package, msbasv3 (20190121), msbasv4 (20201009) and shell scripts dated from October 2022.

**Sources and codes** are available from Github at repositories by ndoreye. If accesses are private, ask [ndo@ecgs.lu](mailto:ndo@ecgs.lu) for being granted the access.

Gitkraken is a convenient interface for Github but is not mandatory.

**Download :**

* the documentation and manuals from ndoreye/DOCS
* the scripts from ndoreye/SCRIPTS\_OK repository
* the MSBAS sources from ndoreye/MSBAS (also available directely from S. Samsonov; see manual)
* the MasTerEngine sources from ndoreye/\_Sources\_ME (also available directly from D. Derauw; see manual)

This installation procedure as be tested on :

* **MacOS Monterey version 12.6**

though former installation was operational already on Mojave (V10.14.6) and former MacOS versions.

**It is assumed that the user has the administrator rights** on the computer and can provide the root privilege when the “sudo” command is executed in the terminal.

In this installation guide, each command line that has to be written and executed in a terminal is written in *italic blue*. Files names are in blue. Paths are given in green.

Note that a script exists to assist you making the installation (at least for recent Mac OS X such as Monterey). See …/SAR/MasTerToolbox/SCRIPTS\_OK/zz\_Utilities\_CIS/***MasTer\_install.sh***.

Read carefully all the text and questions displayed by the script.

For older OS X, you may need to adapt the script or proceed manually by adapting the procedure below.

Note that the ***MasTer\_install.sh*** installation script may be run several times (either to correct a problem during the installation or to perform an update of any element at any time). However, if some elements were installed manually (eg Gitkraken, Fiji or QGIS…), the script may not properly detect them as they may have been installed in an expected. This must not be a problem though.

**Kind reminder:** The software is provided for free. To help us carrying on with the development of the tool, we kindly ask you to cite the **MasTer Toolbox** in publications that contains results produced by the toolboxthanks to at least the most recent references from below:

*For MasTer Engine*

* Derauw, D., (1999). Phasimétrie par Radar à Synthèse d’Ouverture; théorie et applications. PhD Dissertation, University of Liège, 141 pages.
* Libert, L., Derauw, D., d’Oreye, N., Barbier, C., Orban, A., (2017). Split-band interferometry-assisted phase unwrapping for the phase ambiguities correction. Remote Sensing, 9(9), 879.

*For MSBAS*

* Samsonov, S., d’Oreye, N., (2012). Multidimensional time-series analysis of ground deformation from multiple InSAR data sets applied to Virunga Volcanic Province. Geophysical Journal International 191(3), 1095–1108. doi: 10.1111/j.1365-246X.2012.05669.x.
* Samsonov, S., d’Oreye, N., (2017). Multidimensional Small Baseline Subset (MSBAS) for two-dimensional deformation analysis: Case study Mexico City. Canadian Journal of Remote Sensing 82(6), 1–12. doi: 10.1080/07038992.2017.1344926
* Samsonov S., W. Feng, A. Peltier, H. Geirsson, N. d’Oreye, KK. F.Tiampo (2017). Multidimensional Small Baseline Subset (MSBAS) for volcano monitoring in two dimensions: opportunities and challenges. Case study Piton de la Fournaise volcano. Journal of Volcanology and Geothermal Research, Vol 344, 121-138, https://doi.org/10.1016/j.jvolgeores.2017.04.017
* Samsonov, S., A. Dille, O. Dewitte, F. Kervyn, and N. d'Oreye (2020). Satellite interferometry for mapping surface deformation time series in one, two and three dimensions: A new method illustrated on a slow-moving landslide. Engineering Geology, (266), 105471, doi:10.1016/j.enggeo.2019.105471.items

*For MasTer toolbox*

* Derauw, D., N. d’Oreye, M. Jaspard, A. Caselli, and S. Samsonov (2020), Ongoing automated ground deformation monitoring of Domuyo - Laguna del Maule area (Argentina) using Sentinel-1 MSBAS time series: Methodology description and first observations for the period 2015-2020.,   
  Journal of South American Earth Sciences, 104, 102850, doi:10.1016/j.jsames.2020.102850.
* d’Oreye N., Derauw, D., S. Samsonov, M. Jaspard, and D. Smittarello (2021), MasTer: a full automatic multi-satellite InSAR mass processing tool for rapid incremental 2D ground deformation time series. Proc. IEEE IGARSS21, Brussels, July 2021.
* d’Oreye N., Derauw, D., S. Samsonov, M. Jaspard, D. Smittarello, G. Celli (2022), MasTer: InSAR automated Mass processing Toolbox for Multidimensional time series. Instruction manual V4.0, Oct. 2022. 191 pp.

*For DORIS*

* Kampes, B.M., Hanssen, R.F., Perski, Z., (2003). Radar interferometry with public domain tools. In: Third International Workshop on ERS SAR Interferometry, FRINGE03, Frascati, Italy, 1-5 Dec. 2003, 6 pages.

## 0) Prepare the work

The tools will need the following directories:

* /opt/local/bin
* $HOME/SAR/EXEC
* $HOME/SAR/MasTerToolbox/DOC
* $HOME/SAR/MasTerToolbox/MasTerEngine/\_Sources\_ME/Older
* $HOME/SAR/MasTerToolbox/MSBAS
* $HOME/SAR/MasTerToolbox/SCRIPTS\_OK

Where $HOME*/* is obviously your home directory.

It is also advised to source your **.barshrc** in your **.bash\_profile** as all the config will be set in **.bashrc.**

If you do not have these files, create them and ensure you can write and execute them.

## Installing some external components

The following tools and apps are not mandatory but experience shows that it may greatly help the user.

**Macports:**

* Download latest version from [www.macports.org](http://www.macports.org) > Available Downloads:
  + E.g., for Monterey, take MacPorts-2.7.2-12-Monterey.pkg
  + Open and install
  + And/or:
    - *sudo port selfupdate*
    - *sudo port upgrade outdated*
  + If your OS is 10.15 or more recent, it is required to change default shell Zsh with bash for scripts compatibility issues:   
    *chsh -s /bin/bash*

**GitKraken :**

Although not mandatory, Gitkraken is a useful tool to sync with the last versions of MasTer Toolbox. It requires credentials to access MasTer private repositories on Github (contact ndo@ecgs.lu). GitKraken used to access private repositories requires a license though.

* Download GitKraken (from https://www.gitkraken.com) and connect e.g. via Github
* Create profile in GitKraken (may need pay version for accessing private repositories)
* Launch GitKraken
* Clone Repo > Github.com :
  + Where to Clone: $HOME/SAR/MasTerToolbox
  + Repo to clone: DOC, MSBAS, SCRIPTS\_OK
* Clone Repo > Github.com :
  + Where to Clone: $HOME/SAR/MasTerToolbox/MasTerEngine
  + Repo to clone: \_Sources\_ME
* New tab > Open a Repo > select in Finder SAR/MasTerToolbox/RepoYouWant

**To display raster figure: XnView, GIMP or Graphic Converter**

* my favourite is Graphic Converter though it is not free…
* To install GIMP:
  + *sudo port install "gimp2"*

**A text editor: BBEdit** (not free though)

* Get it from <https://www.barebones.com/products/bbedit/download.html>

or any other convenient editor that you like

**Cyberduck**

* Get it from <https://cyberduck.io>

or any other convenient ftp software that you like

**gnu fortan** (not mandatory):

* Get it from <https://github.com/fxcoudert/gfortran-for-macOS/releases>
* install

**GMT and GDAL:**

GMT, GDAL and some associated utilities are required for plots, images and GIS manipulations.

* Using MacPorts, at Terminal, type:
  + *sudo port selfupdate* (to be sure)
  + *sudo port install gdal +hdf5 +netcdf +openjpeg*
  + *sudo port install gmt6*
  + *sudo port install graphicsmagick ffmpeg*
  + search for installation location: *which gmt6*
  + link that *gmt6* to /opt/local/bin/*gmt* for portability

**To display geocoded products: QGIS**

* Download latest version from <https://qgis.org/en/site/forusers/download.html>
* note some interesting plugins are:
  + point sampling tool
  + PointConnetor
  + Profile tool
  + Qdraw
  + QuickMapServices
  + RasterDataPlotting
  + Serval
  + Temporal/Spectal Profile Tool
  + Value Tool

**To quick look scripts with colours when hitting the space bar: QLColorCode**

* + *sudo port install QLColorCode*
  + *reboot*

## 2) Installing required tools

**MasTer** needs some pre-existing software and libraries to work properly. This is what we call the “dependencies”. So, we first need to install them. These installations are explained here, sorted by type of installation procedure.

**bashrc:**

* Open Terminal
* Type: *touch .bashrc*
* Type: *nano .bashrc*
* Copy/paste infos from example\_bashrc.txt eg using nano editor
* Adapt to needs (see below)
* Exit nano by Crtl-X > Y
* Create a .bash\_profile that will use your .bashrc. Type: *touch .bash\_profile*
* Type: *nano .bash\_profile*
* Copy/paste the following lines :

*if [ -r ~/.bashrc ] ; then   
 source ~/.bashrc*

*fi*

* Exit nano by Crtl-X > Y
* Type:

*chmod 700 .bashrc*

*chmod 700 .bash\_profile*

* Reboot

Note: for recent Mac, default shell is not bash. See here how to get it back to default bash:

<https://www.howtogeek.com/444596/how-to-change-the-default-shell-to-bash-in-macos-catalina/>

**Xcode:**

* Install Xcode (required for compilations below). In Terminal type: *xcode-select –install*

**Libraries & modules:**

* Install libraries and modules using macports:
  + *Sudo port install clang-14*
  + *sudo port install fftw-3-long*
  + *sudo port install fftw-3-single*
  + *sudo port install hdf5*Note: At compilation, a note is displayed reading:

Mac users may need to set the environment variable "HDF5\_USE\_FILE\_LOCKING" to the five-character string "FALSE" when accessing network mounted files. This is an application run-time setting, not a configure or build setting. Otherwise errors such as "unable to open file" or "HDF5 error" may be encountered.

Not sure what to do out of it yet though…

* + *sudo port install tiff*
  + *sudo port install libgeotiff*
  + *sudo port install libxml2*
  + *sudo port install lapack*
  + *sudo port install libomp-devel*
  + *sudo port install ImageMagick*
  + *sudo port install gdal +libkml*

**GNU functions and utilities:**

Mac version of some tools such as *sed* or *awk* etc do not have exactly the same syntax as their gnu version. For the sake of portability (and because some options are not available for Mac versions), we use here gnu tools:

* At Terminal:
  + *sudo port install gsed*
  + *sudo port install gawk*

Note: At compilation, a note is displayed reading:

readline support has been removed from gawk. If you need to run gawk interactively, install rlwrap:

*sudo port install rlwrap*

and run gawk using rlwrap:

*rlwrap gawk ...*

* + *sudo port install* *coreutils* (i.e. for gdate, gstat)
  + *sudo port install findutils* (i.e. for find)
  + *sudo port install grep*(i.e. for ggrep)
  + *sudo port install wget* (needed i.e. to download the S1 orbits)
  + *sudo port install curl*
* Check that all these gnu functions are stored in the /opt/local/bin directory. If not, move them there. That directory MUST be defined as ${PATHGNU} state variable in your .bashrc. At Terminal, type
  + *which gsed*
  + *which gawk*
  + *which gdate*
  + *which gstat*
  + *which gfind*
  + *which ggrep*

If answer to all these lines is not the same as echo ${PATHGNU}, edit your .bashrc and change ${PATHGNU} state variable. It must be something like   
export PATHGNU=/opt/local/bin

Don’t forget that a reboot (or source .bashrc) is required for changes in .bashrc to be taken into account.   
Note: ensure that /opt/local/bin is in your path before the /usr/bin to ensure that it would in priority take into account your GNU commands in case of doubt.

* Just in case one of these functions would be called without its g-name in a script, it is recommended to have them in ${PATHGNU} with both names, that is with and without heading g (that is **sed** and g**sed**, **awk** and **gawk**, **grep** and **ggrep**, **date** and **gdate**, **stat** and **gstat**. This can be simply done with a link. In Terminal, type:
  + *cd* ${PATHGNU}
  + *ln -s gsed sed*
  + *ln -s gawk awk*
  + *ln -s gdate date*
  + *ln -s gstat stat*
  + *ln -s gfind find*

And to be sure…(I know it is overkill, but one never know…):

* + *ln -s ggrep grep*
  + *ln -s gseq seq*
  + *ln -s guniq uniq*
  + *ln -s greadlink p readlink*
  + *ln -s gdu du*
  + *ln -s gxargs xargs*

Check also that *wget* is ${PATHGNU}

**For some plotting applications: Java**

Install it with :

* + *sudo port install jdk19*

or manually :

* + see http://www.java.com
  + follow instructions to install Java

Notes:

* if some scripts crash with the following message, re-install java and then re-install Fiji:   
  *The operation couldn’t be completed. Unable to locate a Java Runtime that supports (null).*
* One may need to define JAVA\_HOME state variable in the **.bashrc** for some application using it. Check where it is installed by typing:
  + *java -XshowSettings:properties -version 2>&1 > /dev/null | grep 'java.home' | cut -d = -f2- | cut -d " " -f2-*

and update the **.bashrc** by adding a line with the answer of the line above:

* + *export JAVA\_HOME=PATH\_FROM\_LINE\_ABOVE*

**Fiji/ImageJ:**

* Download latest version from [*https://imagej.net/software/fiji/downloads*](https://imagej.net/software/fiji/downloads)
* Install
* In Finder, locate Fiji.app in your /Applications directory. Right click on it and click on “Show Package Contents”
* Navigate in Contents>MacOS and copy the path to executable (must be ImageJ-macosx)
* Update $PATHFIJI in your .bashrc. It must be something like   
  export PATHFIJI=/Applications/Fiji.app/Contents/MacOS/

**snaphu:**

* Download latest version (e.g. snaphu-v2.0.5.tar) from <https://web.stanford.edu/group/radar/softwareandlinks/sw/snaphu/>   
  and store it in in $HOME/SAR/EXEC.
* unzip it e.g. using *tar -zxvf*
* edit the Makefile in the /src directory to specify your compiler and the optimization flags. Ensure that makefile has no static option in following line  
   CFLAGS = -O3 # -D NO\_CS2
* compile: at Terminal in /src dir, type *make*   
  (it seems that you can ignore warnings; most recent compilers are more sensitive)
* Move compiled snaphu from /bin into $HOME/SAR/EXEC
* Download also the snaphu\_man1.txt if you need more info about snaphu and its configurations.
* Store the installed source e.g. in $HOME*/*SAR/EXEC/Sources\_Installed.

**cpxfiddle:**

Not mandatory, but pretty convenient for creating raster files for quick looking at results.

* Download source from <https://github.com/TUDelftGeodesy/Doris/tree/master/sar_tools>
* Cut/paste source in new document with your favorite editor
* Save the file in $HOME*/SAR/EXEC/cpxfiddle.cc*
* In Terminal:
  + *cd* $HOME*/SAR/EXEC*
  + *make -n cpxfiddle*
  + *g++ -O -c -ocpxfiddle.o cpxfiddle.cc*  (you can ignore warning)
  + *g++ -O cpxfiddle.o -o cpxfiddle*
* Clean sources in $HOME*/SAR/EXEC/*cpxfiddle.o
* Store the installed source cpxfiddle.cc e.g. in $HOME*/*SAR/EXEC/Sources\_Installed
* Update $PATHTOCPXFIDDLE in your .bashrc. It must be something like   
  export PATHTOCPXFIDDLE=/${HOME}/SAR/EXEC/

**gnuplot:**

* At Terminal:
  + *sudo port install gnuplot*
  + *which gnuplot*
* Ensure that the path provided by the former command displays the same path as $PATHGNU in your .bashrc. It must be something like

export PATHGNU=/opt/local/bin

**osascript:**

* should be by default on your mac. Test at Terminal with : *osascript -h*

**python:**

* At Terminal:
  + *sudo port install python310*
  + *sudo port select --set python python310* (To make this the default Python or Python 3)
  + *sudo port select --set python3 python310* (To make this the default Python or Python 3)
  + *sudo port install py310-opencv4*
  + *sudo port install py-numpy*
  + *sudo port install py310-scipy*
  + *sudo port install py310-matplotlib*
  + *sudo port install py310-gdal*
* Check which version is used:
  + at Terminal, type: *python -c 'import sys ; print(sys.path)'*
  + It must answer something like:   
    “['', '/opt/local/Library/Frameworks/Python.framework/Versions/3.10/lib/python310.zip', '/opt/local/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10', '/opt/local/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/lib-dynload', '/opt/local/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages']“
  + If a package such as numpy does not load, search where it is stored and check that it is in the same version as your python. For instance, numpy here must be in something like:   
    /opt/local/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages/numpy  
    If versions differ, change the default python version with same command as above or try to install an appropriate version of the package.
* Copy python to appropriate directory:

For security reasons recent Mac OS now install python in /opt/local/bin instead of /usr/bin, and this is a good practice. If for any reason your system did install python in /usr/bin, please link it to /opt/local/bin/python as that is from where it will be called by all the scripts:

* + *which python3* (To know where it is installed, e.g. */usr/bin*)
  + *sudo ln -s /usr/bin*/pyhton3 */opt/local/bin/python3*(To make this the default Python, providing that the path is indeed */usr/bin*)
  + *sudo ln -s /usr/bin*/pyhton3 */opt/local/bin/python*

(Just in case… i.e. to be sure that it will be called either as python or python3)

* Install utm package, which is used ***in \_LatLong2UTM.py*** script:
  + You must get python 3.10 and pip. To install pip, type at the Terminal :

*curl* [*https://bootstrap.pypa.io/get-pip.py -o get-pip.py*](https://bootstrap.pypa.io/get-pip.py%20-o%20get-pip.py)

*python get-pip.py*

* + Check pip. Since python is probably not in your PATH, type at the Terminal :

*$HOME/Library/Python/3.10/bin/pip --version*

It must answer something like  
pip 22.1.2 from /Users/doris/Library/Python/3.10/lib/python/site-packages/pip (python 3.10)

* + Now install it. Type at the Terminal :

*$HOME/Library/Python/3.10/bin/pip install utm*

* Install pyqt6 package, which is used eg. in ***MasterOrganizer.sh*** script:
  + */opt/local/bin/python -m pip install pyqt6*
  + */opt/local/bin/python -m pip install appscript*
* Add the path to python (*/opt/local/bin/*) in the $PATH in your bashrc. To avoid any risk of using another version of python if badly called, add it in the beginning of the $PATH.

## 3) Installing MasTer

**MasTer** consists of several packages that need to be obtained or downloaded from different sources:

* **MasTerEngine** **(core parts: InSAR and MSBASTools)**:
  + To request the code, contact: [dderauw@ecgs.lu](mailto:dderauw@ecgs.lu)
* **MasTer** **Tool** **(SCRIPTS\_OK: mass processing scripts)**:
  + To request the code, contact [ndo@ecgs.lu](mailto:ndo@ecgs.lu)
* **MSBAS (2D or 3D inversion time series)**:
  + Download from: <https://doi.org/10.4095/313749>
  + You can also request the code prepared with a make file adapted to Mac and Linux MasTer installation, contact [ndo@ecgs.lu](mailto:ndo@ecgs.lu)
  + In case of problem, contact sergey.samsonov@canada.ca

**3.1 Compile MasTerEngine:**

If executable files were provided (zip file),

* simply unzip it and copy the binaries in $HOME/SAR/MasTerToolbox/MasTerEngine
* then store the zipped file in   
  $HOME/SAR/MasTerToolbox/MasTerEngine/\_Sources\_ME/Older/V*mostrecent*\_MasterEngine  
  where*mostrecent* is the date of the last version, which is in the name of the zip file. This date must be as YYYYMMDD and will be used by some scripts to track (and log) the version used during processing.
* Update $PATH in your .bashrc. It must be something like PATH=$PATH:/$HOME/SAR/MasTerToolbox/MasTerEngine

If you were provided with the source code (zip file), simply unzip it in   
$HOME/SAR/MasTerToolbox/MasTerEngine/\_Sources\_ME/Older/V*mostrecent*\_MasterEngine

where*mostrecent* is the date of the last version, which is in the name of the zip file. This date must be as YYYYMMDD and will be used by some scripts to track (and log) the version used during processing. Then,

* In Terminal, in   
  $HOME/SAR/MasTerToolbox/MasTerEngine/\_Sources\_ME/Older/V*mostrecent*\_MasterEngine/MasterEngineYYYYMMDD/InSAR/sources   
  type: *make*
* Move all files from

$HOME/SAR/MasTerToolbox/MasTerEngine/\_Sources\_ME/Older/V*mostrecent*\_MasterEngine/MasterEngineYYYYMMDD/InSAR/bin

in $HOME/SAR/MasTerToolbox/MasTerEngine

* In Terminal, in   
  $HOME/SAR/MasTerToolbox/MasTerEngine/\_Sources\_ME/Older/V*mostrecent*\_MasterEngine/MasterEngineYYYYMMDD/MSBASTools/sources   
  type: *make*
* Move all files from

$HOME/SAR/MasTerToolbox/MasTerEngine/\_Sources\_ME/Older/V*mostrecent*\_MasterEngine/MasterEngineYYYYMMDD/MSBASTools /bin

in $HOME/SAR/MasTerToolbox/MasTerEngine

* Clean (delete) the decompressed directories; keep only the zip file in:  
  $HOME/SAR/MasTerToolbox/MasTerEngine/\_Sources\_ME/Older/V*mostrecent*\_MasterEngine
* Update $PATH in your .bashrc. It must be something like PATH=$PATH:/$HOME/SAR/MasTerToolbox/MasTerEngine

Note that if you are provided with an updated version of MasTer Engine (either as source code or executable binaries), you can simply update it by copying the zipped file it in $HOME/SAR/EXEC and run the script

/$HOME/SAR/MasTerToolbox/SCRIPTS\_OK/zz\_Utilities\_CIS\_Ndo/***UpdateMasterEngine.sh***

**3.2 Compile MSBAS:**

* In Terminal, in $HOME/SAR/MasTerToolbox/MSBAS/3Dsbas\_Archives/msbasv4

decompress msbas\_20201009\_wExtract\_Unified\_20220919\_Optimized\_v1\_Gilles.zip

* In   
  $HOME/SAR/MasTerToolbox/MSBAS/3Dsbas\_Archives/msbasv4/msbas\_20201009\_wExtract\_Unified\_20220919\_Optimized\_v1\_Gilles

type: *make all*

(it seems that you can ignore warnings; most recent compilers are more sensitive)

* Move *msbasv4* in: $HOME/SAR/MasTerToolbox/MSBAS
* In   
  $HOME/SAR/MasTerToolbox/MSBAS/3Dsbas\_Archives/msbasv4/msbas\_20201009\_wExtract\_Unified\_20220919\_Optimized\_v1\_Gilles/msbas\_extract

type: *make all*

* Move *msbas\_extract* in: $HOME/SAR/MasTerToolbox/MSBAS
* Clean (delete) the decompressed directories ; keep the sources in   
  $HOME/SAR/MasTerToolbox/EXEC/Sources\_Installed
* Update $PATH in your .bashrc. It must be something like   
  PATH=$PATH:$HOME/SAR/MasTerToolbox/MSBAS

**3.3 Get the MasTer** **Toolbox** **scripts at the right place:**

* Ensure that the scripts (obtained from Github or from unzipped file) are stored at the right place:   
  $HOME/SAR/MasTerToolbox/SCRIPTS\_OK
* Update $PATH in your .bashrc. It must be something like

PATH=$PATH:$HOME/SAR/MasTerToolbox/SCRIPTS\_OK

PATH=$PATH:$HOME/SAR/MasTerToolbox/SCRIPTS\_OK/zz\_Utilities\_CIS

PATH=$PATH:$HOME/SAR/MasTerToolbox/SCRIPTS\_OK/zz\_Utilities\_CIS\_NdO

PATH=$PATH:$HOME/SAR/MasTerToolbox/SCRIPTS\_OK/\_cron\_scripts

PATH=$PATH:$HOME/SAR/MasTerToolbox/SCRIPTS\_OK/MasterOrganizer

* If scripts were receiver from a zipped file, it can be stored in   
  $HOME/SAR/MasTerToolbox/EXEC/Sources\_Installed

**3.4 Define the environmental variables:**

MasTerEngine, MSBAS and MasTer Toolbox requires environmental variables that will be called from the scripts. These are :

PATH\_SCRIPTS=/${HOME}/SAR/MasTerToolbox : path to SCRIPTS\_OK

(i.e. the directory containing all the scripts)

PATHGNU=/opt/local/bin : path to where the gnu utilities are stored (better to have it in

the $PATH as well)

PATHFIJI=/Applications/Fiji.app/Contents/MacOS/: path to where Fiji application is

PATHCONV=/opt/local/bin : path to where the convert command is (from ImageMagick)

PATHTOCPXFIDDLE=/${HOME}/SAR/EXEC/ : path to where the cpxfiddle command is (from TU Delft)

The path to some disks where data, intermediate results and final results will be stored must also be defined by state variable. At least 5 variables must be defined, that is:

PATH\_1650, PATH\_3600, PATH\_3601, PATH\_3602 and PATH\_DataSAR.

Names are inherited from architecture at ECGS and will be called throughout the processing.

For security reason, it is a good practice to have them pointing to different physical drives, if possible, but this is not mandatory; they can also point toward directories instead of disks, e.g.:

PATH\_1650=/Volumes/hp-1650-Data\_Share1 (disk or dir where data or results will be stored)

PATH\_3600=/Volumes /hp-D3600-Data\_Share1 (disk or dir where data or results will be stored)

PATH\_3601=/Volumes /hp-D3601-Data\_RAID6 (disk or dir where data or results will be stored)

PATH\_3602=/Volumes /hp-D3602-Data\_RAID5 (disk or dir where data or results will be stored)

PATH\_DataSAR=/Volumes /DataSAR/ (disk or dir where auxiliary data will be

automatically stored such as orbits, geoid etc…

in a dir named SAR\_AUX\_FILES)

Note also that more disks can be defined and used e.g. for sharing the computation through several disks when intensive computation is required.

When processing ENVISAT or Sentinel 1 data, MasTerEngine will require orbital information. It will usually pick them for you and store them automatically in dedicated directories where it will get them when required. These directories must be defined by the following state variable names:

S1\_ORBITS\_DIR=${PATH\_DataSAR}/SAR\_AUX\_FILES/ORBITS/S1\_ORB

ENVISAT\_PRECISES\_ORBITS\_DIR=${PATH\_DataSAR}/SAR\_AUX\_FILES/ORBITS/ENV\_ORB/vor\_gdr\_d

To create DEM corrected from geoid height, the following state variable must be defined:

EARTH\_GRAVITATIONAL\_MODELS\_DIR=$PATH\_DataSAR/SAR\_AUX\_FILES/EGM  
A directory named EGM96 must exist in that directory and it must contain the geoid file WW15MGH.DAC. It can be downloaded from (link checked 05 10 2022):

<https://web.archive.org/web/20130314064801/http://earth-info.nga.mil/GandG/wgs84/gravitymod/egm96/binary/WW15MGH.DAC>

## 4) Directory structure for data, auxiliary data and results

In addition to the structure of the installation directories, it is important to store your data, auxiliary data (DEM, masks, kml, orbits, geoid, parameters files…), intermediate results and final results in a structure as described below.

MasTer toolbox expects a directory structure similar as what is shown below. Most of the directories where the intermediate and final results are stored will be created by the scripts based on the information provided in the LaunchParameters.txt file.

For those which must be created by the user (in full or based on name provided in the LaunchParameters.txt file), it is required to follow the following architecture and directories naming.

In *italic green* is the information that must be adapted to your satellite(s), orbital mode(s) and region(s). These directories can be located in the disk(s) you want.

Note that, for safety reasons, it is advised to store e.g. raw data and data in csl format (i.e. from MasTerEngine) on different hard drives, or the mass processed results and the msbas results on different hard drives etc….

Names in red are mandatory.

Disk 1 (i.e. PATH\_1650):

PATH\_1650/kml : where to store kml for cropping, or for coherence check

PATH\_1650/ Param\_files\_SuperMaster : where to store your LaunchParametersFiles.txt by */SAT/TRK*….

PATH\_1650/SAR\_CSL/*SAT/TRK*/NoCrop : where data in csl format are stored (incl. crops)

PATH\_1650/SAR\_SM/AMPLITUDES/*SAT/TRK/Region* : where amplitudes gif will be computed

PATH\_1650/SAR\_SM/MSBAS/*Region* : where compatible pairs for msbas are computed

PATH\_1650/SAR\_SM/RESAMPLED/*SAT/TRK* : where data resampled on Super Master are saved

Disk 2 (i.e. PATH\_3600):

PATH\_3600/SAR\_DATA/*SAT/…* : where raw data are stored (safer to keep them on a different hard drive than 1650)

Disk 3 (i.e. PATH\_3601):

PATH\_3601/SAR\_MASSPROCESS/*SAT/TRK/Crop* : where directories with each pair computation details and results are stored

PATH\_3601/SAR\_MASSPROCESS/*SAT/TRK/Crop*/Geocoded : where geocoded maps are stored in sub dir Amp, Coh, Defo etc…

PATH\_3601/SAR\_MASSPROCESS/*SAT/TRK/Crop*/GeocodedRasters : where rasters of geocoded maps (for quick look) are stored

Disk 4 (i.e. PATH\_3602):

PATH\_3602/MSBAS\_RESULTS/*Region* : where msbas series are computed

Maybe in one of these disks, or somewhere else where enough storage room is available:

$PATH\_DataSAR/SAR\_AUX\_FILES/DEM : where DEMs are stored

$PATH\_DataSAR/SAR\_AUX\_FILES/EGM96 : where geoid is stored

$PATH\_DataSAR/SAR\_AUX\_FILES/MASKS/*SAT/TRK* : where coherence masks are computed

$PATH\_DataSAR/SAR\_AUX\_FILES/ORBITS/*SAT* : where orbits are stored

Note that $PATH\_DataSAR is the state variable defined in .bashrc !!

## 5) Check installation

**5.1 The directory structure**

At the end, you should have MasTer installed in a folder named “SAR” as shown in **Fig. 1**.

*/$HOME/SAR/EXEC*

*|--Executables…*

*|--Sources\_Installed*

*/$HOME/SAR/MasTerToolbox*

*|--****DOC***

*| |----Biblio*

*| |----MasTerEngine*

*| |----MasTer\_Toolbox*

*| |----MSBAS*

*|*

*|--****MasTerEngine***

*| |--\_Sources\_ME/Older*

*| | |--[several sources of old versions]*

*| | |--V20220929\_MasTerEngine*

*| |--Executables…*

*|*

*|--****MSBAS***

*| |----3Dsbas\_Archives*

*| | |--msbasv4*

*| |--Executables…*

*|*

*|--* ***SCRIPTS\_OK***

*| |--* *\_cron\_scripts*

*| |--* *MasTerOrganizer*

*| |--* *TSCombiFiles*

*| |--* *zz\_Utilities\_CIS*

*| |--* *zz\_Utilities\_CIS\_Ndo*

*| A lot of scripts…*

**Figure 1 – Directory structure required for the proper usage of MasTer Toolbox.**

Notes:

1 ) The folders /$HOME/SAR/MasTerToolbox/MasTerEngine/\_Sources\_ME/Olderis used to store the archives and the most recent sources of both components of MasTerEngine, that is InSAR and MSBASTools.

Their latest executable version, when compiled, must be moved to   
/$HOME/SAR/MasTerToolbox/MasTerEngine/.

Hence, when a new version of these packages is available, no need to change the paths of the environmental variables.

2) Similarly, the folders /$HOME/SAR/MasTerToolbox/MSBAS/\_3Dsbas\_Archivesis used to store the archives and the most recent sources of both components of MSBAS, that is MSBAS\_EXTRACT and MSBASv*i* (where *i* stands for the version). Their latest executable version, when compiled, must be moved to /$HOME/SAR/MasTerToolbox/MSBAS/.

Hence, when a new version of these packages is available, no need to change the paths of the environmental variables.

1. You can use the script Check\_Installation.sh to check your installation. See script and manual.

You should also have auxiliary files stored in a folder named “SAR\_AUX\_FILES” in DataSAR as shown in **Fig. 2**.

*$PATH\_DataSAR/SAR\_AUX\_FILES/*

*|--****DEM***

*|*

*|--****EGM96***

*| |----*WW15MGH.DAC

*|*

*|--****Masks***

*|*

*|--****ORBITS***

*|----ENV\_ORB*

*|----S1\_ORB*

**Figure 2 – Directory structure required for the storage of**

**auxiliary files required by MasTer Toolbox.**

Notes:

1 ) The folders /$PATH\_DataSAR/SAR\_AUX\_FILES/DEM/is used to store the DEM. It may be associated to a state variable $EXTERNAL\_DEMS\_DIR used automatically by MasTer Engine. However, for the automation with MasTer Toolbox, to avoid possible confusion, the path to the DEM for each run is rather given in the LaunchParameters.txt file.

2) The folders /$PATH\_DataSAR/SAR\_AUX\_FILES/EGM must contain the geoid file in EGM96/WW15MGH.DAC.

3) The folders /$PATH\_DataSAR/SAR\_AUX\_FILES/MASK is not mandatory, but it is a convenient place to store the mask you would build and used during your processing.

3) The folders /$PATH\_DataSAR/SAR\_AUX\_FILES/ORBITS must contain the orbits for ENVISAT (if required) or Sentinel 1 (if required). These directories must be associated to the state variable $ENVISAT\_PRECISES\_ORBITS\_DIR and $S1\_ORBITS\_DIR respectively.

**5.2 The environmental variables in .bashrc**

At the end, your ./bashrc must look like:

#

# .bashrc

#

[…]

# User specific environment and startup programs

PATH=/opt/local/bin:$PATH

PATH="$PATH:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin"

BASH\_ENV=$HOME/.bashrc

USERNAME=""

LANG=C

# export EDITOR=/usr/bin/edit

export EDITOR=/usr/bin/nano

export PAGER=/usr/bin/less

export USERNAME LANG BASH\_ENV PATH

export JAVA\_HOME=/Library/Internet\ Plug-Ins/JavaAppletPlugin.plugin/Contents/Home

# MasTer PATHS

##################

PATH=$PATH:/Users/doris/SAR/MasTerToolbox/SCRIPTS\_OK/MasTerOrganizer

PATH=$PATH:/Users/doris/SAR/EXEC

PATH=$PATH:/Users/doris/SAR/MasTerToolbox/MSBAS

PATH=$PATH:/Users/doris/SAR/MasTerToolbox/MasTerEngine/\_Sources\_ME

PATH=$PATH:/Users/doris/SAR/MasTerToolbox/MasTerEngine

PATH=$PATH:/Users/doris/SAR/MasTerToolbox/SCRIPTS\_OK/zz\_Utilities\_CIS\_Ndo

PATH=$PATH:/Users/doris/SAR/MasTerToolbox/SCRIPTS\_OK/zz\_Utilities\_CIS

PATH=$PATH:/Users/doris/SAR/MasTerToolbox/SCRIPTS\_OK/\_cron\_scripts

PATH=$PATH:/Users/doris/SAR/MasTerToolbox/SCRIPTS\_OK

# MasTer VARIABLES

##################

export PATH\_HOMEDATA=/Users/doris/

export PATH\_3602=/Volumes/hp-D3602-Data\_RAID5/

export PATH\_3601=/Volumes/hp-D3601-Data\_RAID6/

export PATH\_3600=/Volumes/hp-D3600-Data\_Share1/

export PATH\_1650=/Volumes/hp-1650-Data\_Share1/

export PATH\_DataSAR=/Volumes/DataSAR/

export EARTH\_GRAVITATIONAL\_MODELS\_DIR=${PATH\_DataSAR}/SAR\_AUX\_FILES/EGM

export ENVISAT\_PRECISES\_ORBITS\_DIR=${PATH\_DataSAR}/SAR\_AUX\_FILES/ORBITS/ENV\_ORB

export S1\_ORBITS\_DIR=${PATH\_DataSAR}/SAR\_AUX\_FILES/ORBITS/S1\_ORB

export PATH\_SCRIPTS=/Users/doris/SAR/MasTerToolbox

export PATHCONV=/opt/local/bin

export PATHFIJI=/Applications/Fiji.app/Contents/MacOS/

export PATHGNU=/opt/local/bin

export PATHTOCPXFIDDLE=/Users/doris/SAR/EXEC/

export PATH=$PATH

#export OMP\_NUM\_THREADS=10,8,4

#export OMP\_NUM\_THREADS=4,3,2

## 6) Operating with cron jobs

If you want to operate **MasTer** toolbox from cron jobs, pay attention to the following:

* It is advised to launch scripts by redirecting the messages to /dev/null to avoid filling up the mailbox by ending the command line with

> /dev/null 2>&1

* If bash is not your default shell (as for the recent Mac OS X) and you want to run scripts from cronjobs, call the scripts in your crontab by calling bash as follow:

hh mm \* \* \* /bin/bash YourScript.sh > /dev/null 2>&1

Example of .bashrc:

# Always end command line launched by cron with > /dev/null 2>&1 to mute output and avoir filling up the mailbox. It will also prevent wrong count of running processes if any...

## TO GET THE SCRIPTS RUNNING USING CMD LINES AND SCRIPTS STORED IN PATH AVAILABLE IN BASHRC RATHER THAN IN COMMON

## /user/local/bin FOR INSTANCE, DO NOT FORGET TO SOURCE THE BASHRC AT THE BEGINNING OF THE SCRIPT !!

# MM HH DoMonth Month DoWeek(0=sunday)

## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONGO VVP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## Step 1: S1 & CSK

45 04 \* \* \* /bin/bash /Users/doris/SAR/SCRIPTS\_OK/\_cron\_scripts/VVP\_S1\_Step1\_Read\_SMCoreg\_Pairs.sh > /dev/null 2>&1

30 12 \* \* \* /bin/bash /Users/doris/SAR/SCRIPTS\_OK/\_cron\_scripts/VVP\_CSK\_Step1\_Read\_SMCoreg\_Pairs.sh > /dev/null 2>&1

## Step 2: S1 & CSK

31 06 \* \* \* /bin/bash /Users/doris/SAR/SCRIPTS\_OK/\_cron\_scripts/VVP\_S1\_Step2\_Read\_MassProc.sh > /dev/null 2>&1

31 07 \* \* \* /bin/bash /Users/doris/SAR/SCRIPTS\_OK/\_cron\_scripts/VVP\_CSK\_Step2\_MassProc.sh > /dev/null 2>&1

## Step 3: S1

02 21 \* \* \* /bin/bash /Users/doris/SAR/SCRIPTS\_OK/\_cron\_scripts/VVP\_S1\_Step3\_MSBAS.sh > /dev/null 2>&1

02 02 \* \* \* /bin/bash /Users/doris/SAR/SCRIPTS\_OK/\_cron\_scripts/VVP\_CSK\_Step3\_MSBAS.sh > /dev/null 2>&1

## Website Update (once a week)

00 06 \* \* 0 /bin/bash /Users/doris/SAR/SCRIPTS\_OK/\_cron\_scripts/rsync\_satellite\_images\_to\_virunga\_volcanoes.sh > /dev/null 2>&1