Installation of MasTer

(FOR Linux DISTROS)

Installation guide by:
N. d'Oreye, D. Smittarello, M. Jaspard, G. Celli – ECGS
B. Smets – RMCA

Version 5.0: Aug 31 2023

Preliminary information 2
0) Prepare the work 5
1) Installing some external components 6
2) Installing required tools 9
3) Installing MasTer 14
4) Directory structure for data, auxiliary data and results 17
5) Check installation 18
6) Operating with cron jobs 21
7) In case of insufficient RAM 22
8) Updates in this manual 23

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 (MasTer_Manual_x.x.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_v20231009** package, msbasv3 (20190121), msbasv4 (20201009) and shell scripts dated from Aug 2023.

Sources and codes are available from Github at repositories by ndoreye.
If you are not granted the access to the repositories, please send an email to ndo@ecgs.lu.

Download:
Either from the **distribution repository** (all in one package):
- https://github.com/ndoreye/MasTerToolbox_Distribution

Or from the **development repositories**:
- the documentation and manuals from ndoreye/DOCS
- the scripts from ndoreye/SCRIPTS_MT
- the MSBAS sources tuned for MasTer Toolbox from ndoreye/MSBAS (also available directly from S. Samsonov; see manual)
- the MasTerEngine sources from ndoreye/_Sources_ME (also available directly from D. Derauw; see manual)

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

This installation procedure as been tested on:
- **Ubuntu 22.0.4 LTS**

though former installations were operational.

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:

A script exists to assist you making the installation (at least for Debian OS). See
…/SAR/MasTerToolbox/SCRIPTS_MT/zz_Utilities_MT/***MasTer_install.sh***
(in distribution repository), or
…/SAR/MasTerToolbox/SCRIPTS_MT/zz_Utilities_MT/***MasTer_install.sh***
(in development repository).

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

For other type of Linux OS, 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 toolbox thanks 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.
- Smittarello, D., d'Oreye, N., Jaspard, M., Derauw, D., & Samsonov, S. (2022).
  Pair selection optimization for InSAR time series processing. *Journal of Geophysical Research: Solid Earth, 127, e2021JB022825.*

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_MT

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.

If you have old MasTer installation parameters in your **.bashrc**, it is advised to remove them and write new ones based on the instructions below.

1) Installing some external components

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

Note: if apt name or lib has changed, you can search by keyword the new name by typing at the Terminal:

- *sudo apt-cache search <keyword>*

Update installation:

- If not done yet:
  - *sudo apt install make*
  - *sudo apt install gzip*
- To ensure update version of OS, type at terminal:
  - *sudo apt update*
  - *sudo apt upgrade*
- For old version of Ubuntu (<18):
  - *sudo apt install snapd*

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.

- Command line download (gitkraken-amd64.deb), and install by typing
  - *wget -c https://release.gitkraken.com/linux/gitkraken-amd64.deb*
  - *sudo apt install ./gitkraken-amd64.deb*

- In case of manual download (gitkraken-amd64.deb from web site), install by typing
  - *sudo dpkg -i gitkraken-amd64.deb*

- may also be done using snap:
  - *sudo snap install gitkraken --classic*

- 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_MT
- Clone Repo > Github.com:
  - Where to Clone: $HOME/SAR/MasTerToolbox/MasTerEngine
  - Repo to clone: _Sources_ME
- New tab > Open a Repo > select in Finder $HOME/SAR/MasTerToolbox /RepoYouWant

To display raster figure: XnView, GIMP or Graphic Converter

For instance, see https://www.gimp.org/downloads/
Or at Terminal :

- *sudo add-apt-repository ppa:ubuntuhandbook1/gimp*
- *sudo apt update*
- *sudo apt install gimp*

Editor:

- Get a good text editor you like

FTP:

- Get a convenient ftp software that you like

gnu fortran (not mandatory):

- Get it from http://fortran-lang.org/learn/os_setup/install_gfortran
- install with:
  - *sudo apt-get install gfortran*

GMT and GDAL:

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

- At Terminal, type:
  - *sudo apt install gdal-bin*
  - *sudo apt install libgdal-dev*
  - *sudo apt install libgdal26*
  - check gdal by typing: *gdalinfo --version*
  - *sudo apt install libhdf5-dev*
  - *sudo apt install gmt*
  - *sudo apt install libnetcdf-dev*
  - *sudo apt install graphicsmagick ffmpeg*
  -
  - *sudo apt install openjpeg* (optional I think)

To display geocoded products: QGIS

- Visit https://qgis.org and get it eg as follow:
- At Terminal type:
  - *sudo apt install gnupg software-properties-common*
  - *wget -qO - https://qgis.org/downloads/qgis-2021.gpg.key | sudo gpg --no-default-keyring --keyring gnupg-ring:/etc/apt/trusted.gpg.d/qgis-archive.gpg –import*
  - *sudo chmod a+r /etc/apt/trusted.gpg.d/qgis-archive.gpg*
  - *sudo add-apt-repository "deb https://qgis.org/ubuntu $(lsb_release -c -s) main"*
  - *sudo apt update*
  - *sudo apt install qgis qgis-plugin-grass*

- note some interesting plugins are:
  - point sampling tool
  - PointConnetor

- Profile tool
- Qdraw
- QuickMapServices
- RasterDataPlotting (may require to install python first)
- Serval
- Temporal/Spectal Profile Tool
- Value Tool

For special characters if you keyboard sometimes plays it funny:

- Install Ubuntu package *Character Map*

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.

GNU functions and utilities:

Several commands might exist by default on Linux, but getting the gnu version again is wise.

To be sure to have the latest version of gnu tools, type at the Terminal:

- *sudo apt install sed*
- *sudo apt install gawk*
- *sudo apt install coreutils* (i.e. for gdate, gstat)
- *sudo apt install findutils* (i.e. for find)
- *sudo apt install grep* (i.e. for ggrep)
- *sudo apt install wget* (needed i.e. to download the S1 orbits)
- *sudo snap install curl* and add the path to curl (e.g. /snap/bin) in your $PATH

- Check that all these gnu functions are stored in the /usr/bin directory. If not, move them there. That directory MUST be defined as ${PATHGNU} state variable in your .bashrc. At Terminal, type
  - *which sed*
  - *which gawk*
  - *which date*
  - *which stat*
  - *which find*
  - *which grep*

  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=/usr/bin
  Don’t forget that a reboot (or source .bashrc) is required for changes in .bashrc to be taken into account.

- 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}
  - *sudo ln -s sed gsed*
  - *sudo ln -s gawk awk* (beware if *awk* already exist and would be a link pointing toward another version of *awk*. Replace it with our *gawk*)
  - *sudo ln -s date gdate*
  - *sudo ln -s stat gstat*
  - *sudo ln -s find gfind*

  And to be sure…(I know it is overkill, but one never know…):
  - *sudo ln -s grep ggrep*
  - *sudo ln -s seq gseq*
  - *sudo ln -s uniq guniq*
  - *sudo ln -s readlink greadlink*
  - *sudo ln -s xargs gxargs*
  - *sudo ln -s du gdu*

  Check also that *wget* is ${PATHGNU}

Libraries & modules:

- Install libraries and modules :
  - *sudo apt install clang*
  - *sudo apt install libfftw3-dev*
  - *sudo apt install libfftw3-long3*
  - *sudo apt install libfftw3-single3*
  - *sudo apt install libgeotiff-dev*
  - *sudo apt install libtiff-dev*
  - *sudo apt install libxml2*
  - *sudo apt install libxml2-dev*
  - *sudo apt install liblapack-dev*
  - *sudo apt install libomp-dev*
  - *sudo apt install libopenblas-dev*
  - *sudo apt install graphicsmagick-imagemagick-compat*
  - *sudo apt install imagemagick-6-common*
  - *sudo apt install imagemagick*
  - *sudo apt install g++*
  - *sudo apt-get espeak -y* (to make your computer talking during mass processing)
  - *sudo apt install parallel*

Attention: for *convert* function to operate properly **ps** and **eps** files, change permissions in /etc/ImageMagick/policy.xml (or /etc/ImageMagick-6/policy.xml) and set (e.g. using nano as sudo):

</policy domain="coder" rights="none" pattern="PS" />

as

</policy domain="coder" rights="read|write" pattern="PS" />

(i.e. with pipe between read and write), and

</policy domain="coder" rights="none" pattern="EPS" />

as

</policy domain="coder" rights="read|write" pattern="EPS" />

Also, change as follow:

</policy domain="resource" name="height" value="16KP" />

as

</policy domain="resource" name=" height" value="32KP" />

and

</policy domain="resource" name="width" value="16KP" />

as

</policy domain="resource" name=" width" value="32KP" />

and

</policy domain="resource" name="disk" value="1GiB" />

as

</policy domain="resource" name="disk" value="8GiB" />

For some plotting applications: Java (preferably installed before Fiji)

Install it with :

- *sudo apt install default-jdk*

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'*

  and update the **.bashrc** by adding a line with the answer of the line above:
  - *export JAVA_HOME=PATH_FROM_LINE_ABOVE*

Fiji/ImageJ:

Do not install it with the terminal !!! You must use the stand-alone version.

- Download latest version from *https://imagej.net/software/fiji/downloads*
- Unzip the compressed file in $HOME/SAR/EXEC.
- Locate ImageJ-linux64 in your $HOME/SAR/EXEC/Fiji.app directory.
- Update $PATHFIJI in your .bashrc. It must be something like
  export PATHFIJI=$HOME/SAR/EXEC/Fiji.app/
- Store the installed source e.g. in $HOME/SAR/EXEC/Sources_Installed.

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.

Note: this installation can also be made using install, but it might not be the latest version and path must be checked…

- *sudo apt install snaphu*

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*
- Edit $HOME/*SAR/EXEC/cpxfiddle.cc* and replace line 1857:
  if (argv[optind]=='\0')
  by
  if (argv[optind]==0 || argv[optind][0]=='\0')
- In Terminal:
  - *cd* $HOME/*SAR/EXEC*
  - *make -n cpxfiddle*

  - *g++ -O -c -ocpxfiddle.o cpxfiddle.cc*
  - *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: gnuplot is used e.g. to create time series figures or baseline plots.

- At Terminal:
  - *sudo apt install gnuplot-x11*
  - *which gnuplot*
- Ensure that the path provided by the former command displays the same path as $PATHGNU in your .bashrc, that is /usr/bin

python:

- At Terminal:
  - *sudo apt install python3*
  - *sudo apt install python3-opencv*
  - *sudo apt install python3-numpy*
  - *sudo apt install python3-scipy*
  - *sudo apt install python3-matplotlib*
  - *sudo apt install python3-gdal*
- Check which version is used:
  - at Terminal, type: *python -c 'import sys ; print(sys.path)'*
  - It must answer something like:
    "['', '/usr/bin/python38.zip', '/usr/lib/python3.8', '/usr/lib/python3.8/lib-dynload', '/usr/local/lib/python3.8/dist-packages', '/usr/lib/python3/dist-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.

    If versions differ, change the default python version with same command as above or try to install an appropriate version of the package.
- Link python to appropriate directory: It MUST be in /opt/local/bin/ as it is from there that it will be called by all the scripts:
  - *sudo mkdir -p /opt/local/bin*
  - *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 eg. in ***_LatLong2UTM.py*** script:
  - *pip install --upgrade pip*
  - *sudo apt install python3-pip*
  - *pip install utm*

- Install pyqt6 package, which is used eg. in ***MasterOrganizer.sh*** script:
  - */opt/local/bin/python -m pip install pyqt6*

- 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.

x-terminal-emulator: A terminal emulator is required to open terminal from command line (e.g. when splitting mass processing).

- Install it by typing at the terminal:
  - *sudo apt install deepin-terminal*

3) Installing MasTer

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

- **MasTerEngine (core parts: InSAR and MSBASTools)**:
  - See **Preliminary information/Sources and codes** above for Github instructions, or
  - To request the executables, contact: dderauw@ecgs.lu
- **MasTer Tool (SCRIPTS_MT: mass processing scripts)**:
  - See **Preliminary information/Sources and codes** above for Github instructions, or
  - To request the code, contact ndo@ecgs.lu
- **MSBAS (2D or 3D inversion time series)**:
  - See **Preliminary information/Sources and codes** above for Github instructions, or
  - Download original version from: https://doi.org/10.4095/313749, or
  - Request code tuned for MasTer with a make file adapted to Mac and Linux, contact 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 (source code), you can simply update it by copying the zipped file it in $HOME/SAR/EXEC and run the script /$HOME/SAR/MasTerToolbox/SCRIPTS_MT/zz_Utilities_MT_Ndo/***UpdateMasterEngine.sh*** with 2 parameters: the path to the zipped sources file and the date of the version.

3.2 Compile MSBAS:

- In Terminal, in $HOME/SAR/MasTerToolbox/MSBAS/3Dsbas_Archives/msbasv4
  decompress msbas_20201009_wExtract_Unified_20220919_Optimized_v1.1_Gilles.zip
- In
  $HOME/SAR/MasTerToolbox/MSBAS/3Dsbas_Archives/msbasv4/msbas_20201009_wExtract_Unified_20220919_Optimized_v1.1_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_v1.1_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_MT

- Update $PATH in your .bashrc. It must be something like
  PATH=$PATH:$HOME/SAR/MasTerToolbox/SCRIPTS_MT
  PATH=$PATH:$HOME/SAR/MasTerToolbox/SCRIPTS_MT/zz_Utilities_MT
  PATH=$PATH:$HOME/SAR/MasTerToolbox/SCRIPTS_MT/zz_Utilities_MT_NdO
  PATH=$PATH:$HOME/SAR/MasTerToolbox/SCRIPTS_MT/_cron_scripts
  PATH=$PATH:$HOME/SAR/MasTerToolbox/SCRIPTS_MT/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_MT
(i.e. the directory containing all the scripts)

PATHGNU=/usr/bin : path to where the gnu utilities are stored
(Note: in case you have several versions of the gnu utilities, it is highly recommended to have the path to /usr/bin defined first in your $PATH)

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

PATHCONV=/usr/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=/mnt/hp-1650-Data_Share1 (disk or dir where data or results will be stored)
PATH_3600=/mnt/hp-D3600-Data_Share1 (disk or dir where data or results will be stored)
PATH_3601=/mnt/hp-D3601-Data_RAID6 (disk or dir where data or results will be stored)
PATH_3602=/mnt/hp-D3602-Data_RAID5 (disk or dir where data or results will be stored)
PATH_DataSAR=/mnt/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 LaunchMTparameters.txt file.

For those which must be created by the user (in full or based on name provided in the LaunchMTparameters.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 LaunchMTparameters.txt file 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 !!

Note: In Linux, to mount external drive, you must

1) Define the mounting point of your disks, e.g for a disk named “syno_sar”:

```
sudo mkdir /mnt/syno_sar
sudo chown yourlogin /mnt/syno_sar
```

2) Add them in the /etc/fstab. Example below shows the mounting of a Windows server and a Synology server:

```
# EG FOR A WINDOWS STORAGE SERVER
# IMPORTANT: Leave "cache=none" to avoid issues with 10Gbit Ethernet-Network on eno1
//hp-storeeasy.ecgs.welter/hp-1650-Data_Share1 /mnt/1650  cifs comment=systemd.au-
tomount,username=YOURUSRNAME,password=YOURPWD,uid=YOURID,gid=users,noatime,iocharset=utf8,rw,mfsym-
links,cache=strict,vers=3.1.1,soft,noserverino,nobrl,nounix 0 0

# EG FOR SYNOLOGY STORAGE
```

```
//syno-sar.ecgs.welter/DataSAR /mnt/syno_sar cifs comment=systemd.au-
tomount,username=YOURUSRNAME,password=YOURPWD,uid=YOURID,gid=1001,iochar-
set=utf8,file_mode=0777,dir_mode=0777,noatime,rw,mfsymlinks,soft,nounix 0 0
```

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]
            |     |                  |--V20230828_MasTerEngine
            |     |--Executables…
            |
            |--MSBAS
            |     |----3Dsbas_Archives
            |     |              |--msbasv4
            |     |--Executables…
            |
            |-- SCRIPTS_MT
            |          |-- _cron_scripts
            |          |-- MasTerOrganizer
            |          |-- optimtoolbox
            |          |-- TemplatesForPlots
            |          |-- TSCombiFiles
            |          |-- zz_Utilities_MT
            |          |-- zz_Utilities_MT_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/Older is 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_Archives is 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.

3) 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**.

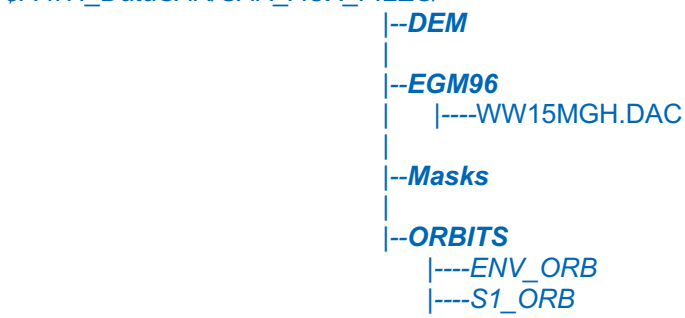

Figure 2 – Directory structure required for the storage of auxiliary files required by MasTer Toolbox.

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 LaunchMTparameters.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: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
#case $- in
#    *i*) ;;
#      *) return;;
#esac

[…]

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


# trick to avoid error at usage of say function
alias say='echo  | espeak -s 120 2>/dev/null'

# MasTer PATHS
###################

PATH=$PATH:/home/YourAccount/SAR/MasTerToolbox/MasTerEngine/_Sources_ME
PATH=$PATH:/home/YourAccount/SAR/MasTerToolbox/SCRIPTS_MT/MasTerOrganizer
PATH=$PATH:/usr/local/tigervnc/bin/
PATH=$PATH:/home/YourAccount/SAR/EXEC
PATH=$PATH:/home/YourAccount/SAR/MasTerToolbox/MSBAS
PATH=$PATH:/home/YourAccount/SAR/MasTerToolbox/MasTerEngine
PATH=$PATH:/home/YourAccount/SAR/MasTerToolbox/SCRIPTS_MT/zz_Utilities_MT_Ndo
PATH=$PATH:/home/YourAccount/SAR/MasTerToolbox/SCRIPTS_MT/zz_Utilities_MT
PATH=$PATH:/home/YourAccount/SAR/MasTerToolbox/SCRIPTS_MT/_cron_scripts
PATH=$PATH:/home/YourAccount/SAR/MasTerToolbox/SCRIPTS_MT
PATH=$PATH:/home/YourAccount/SAR/MasTerToolbox/SCRIPTS_MT/optimtoolbox

# MasTer VARIABLES
###################

export OPENBLAS_NUM_THREADS=1
export JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64"
export PATH_HOMEDATA=/mnt/dell3raid5/
export PATH_SynoData=/mnt/syno_data/
export PATH_3602=/mnt/3602/
export PATH_3601=/mnt/3601/
export PATH_3600=/mnt/3600/
export PATH_1650=/mnt/1650/
export PATH_DataSAR=/mnt/syno_sar/
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=${PATH_DataSAR}/SAR_AUX_FILES/ORBITS/S1_ORB
export PATH_SCRIPTS=/home/YourAccount/SAR/MasTerToolbox
export PATHCONV=/usr/bin
export PATHFIJI=/home/YourAccount/SAR/EXEC/Fiji.app/
export PATHGNU=/usr/bin
export PATHTOCPXFIDDLE=/home/YourAccount/SAR/EXEC/
export PATH=$PATH

#export OMP_NUM_THREADS=10,8,4
#export OMP_NUM_THREADS=4,3,2

# Trick to avoid error at usage of say function
alias say='echo  | espeak -s 120 2>/dev/null'
```

6) Operating with cron jobs

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

- All variable environments are defined here in the .***bashrc*** which is designed for interactive shell. Whatever you do, it won't be sources from scripts if it starts with the following lines (as by default in at least Ubuntu 18.04 and later):

```
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
   *i*) ;;
     *) return;;
esac
```

- It is advised to source the .***bashrc*** from the ***.bash_profile***

- It is advised to define all your state variables at the beginning of your bashrc

- 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
```

- You can launch your commands in the crontab as follow (supposing that the path to your bash in /bin):

```
mm hh * * * exec /bin/bash -l  /path/to/script  arg1 arg2 ... > /dev/null 2>&1
```

Example of crontab:

```
# Edit this file to introduce tasks to be run by cron.
# m h  dom mon dow   command

SHELL=/bin/bash
BASH_ENV=~/.bashrc

PATH_1650=/mnt/1650
PATH_3600=/mnt/3600
PATH_3601=/mnt/3601
PATH_3602=/mnt/3602
PATH_SynoData=/mnt/syno_data
PATH_DataSAR="/mnt/syno_sar"
PATH_HOMEDATA="/mnt/dell3raid5"
PATH_SCRIPTS=/home/nicolas/SAR/MasTerToolbox

S1_ORBITS_DIR=${PATH_DataSAR}/SAR_AUX_FILES/ORBITS/S1_ORB
ENVISAT_PRECISES_ORBITS_DIR=/mnt/syno_sar/SAR_AUX_FILES/ORBITS/ENV_ORB
EARTH_GRAVITATIONAL_MODELS_DIR=$PATH_DataSAR/SAR_AUX_FILES/EGM

PATHGNU=/usr/bin
PATHCONV=/usr/bin
PATHFIJI=$HOME/SAR/EXEC/Fiji.app/
PATHTOCPXFIDDLE=/${HOME}/SAR/EXEC/

# # # # # Domuyo 03h05; 5h35; 17h30
05 03 * * * cd /home/nicolas/SAR/MasTerToolbox/SCRIPTS_MT/_cron_scripts/; bash -l -c './Domuyo_S1_Step1_Read_SMCoreg_Pairs.sh'
35 05 * * * cd /home/nicolas/SAR/MasTerToolbox/SCRIPTS_MT/_cron_scripts/; bash -l -c './Domuyo_S1_Step2_MassProc.sh'
30 17 * * * cd /home/nicolas/SAR/MasTerToolbox/SCRIPTS_MT/_cron_scripts/; bash -l -c './Domuyo_S1_Step3_MSBAS.sh'
```

7) In case of insufficient RAM

It may happen that your Linux computer runs out of RAM and MSBAS inversion can't be completed. See the document "Create_Swap_Linux.pdf" which explains how to create a swap file on a Linux computer.

8) Updates in this manual

V 4.1 (October 26 2022):
- add installation of gnu parallel
- add chapter “7) Updates in this manual”

V 4.3 (April 20 2023):
- Refer to last versions of software and scripts
- Check against installer script

V 4.4 (June 27 2023):
- Add reference to technical note on how create swap file

V 5.0 (Aug 30 2023):
- Major changes in dirs and files naming
- Take into account new MasTer Engine V20230828 which is mandatory