

GRASS GIS 7: A theoretical and practical course

Session 1 – GRASS data management
Projections and EPSG codes
Creating an own GRASS GIS location

Markus Neteler

NINA 2016, Oslo, Norway

mundialis GmbH & Co. KG
<http://www.mundialis.de>





Session Objectives

- Hints about network based installation & usage of GRASS GIS
- User access rights management
- Intro: EPSG codes for projection management
- Creating a new location with the location manager
 - A new location from a GeoTIFF file
 - A new location using an EPSG code



GRASS GIS data management

Hints about **network based installation & usage** of GRASS GIS:

- GRASS GIS supports teams through **shared network locations**
- Data: In a shared location, store the base cartography in the **PERMANENT mapset**. The shared PERMANENT location **minimizes data duplication**
- Access rights management is performed (see next slide)
- “/grassdata/” is ideally created and managed on a shared network directory

GRASS GIS software **network installation**:

- The software can simply be installed centrally in a network directory and used by all – advantage: just a single binary to maintain



User access rights management (shared network GRASS GIS locations):

- Only the directory owner of PERMANENT can modify maps stored inside but all group members can read these data
- Every user works in his/her own mapset (for others optionally readable but not writable)
- A user can “activate” (add to map search path) other mapsets with `g.mapsets`, i.e. the selected mapset will be added to the search path
 - Hence “grassdata/” is ideally managed on a network directory



Intro: EPSG codes for projection management

*EPSG: European Petroleum Survey Group
(International Association of Oil & Gas Producers)*

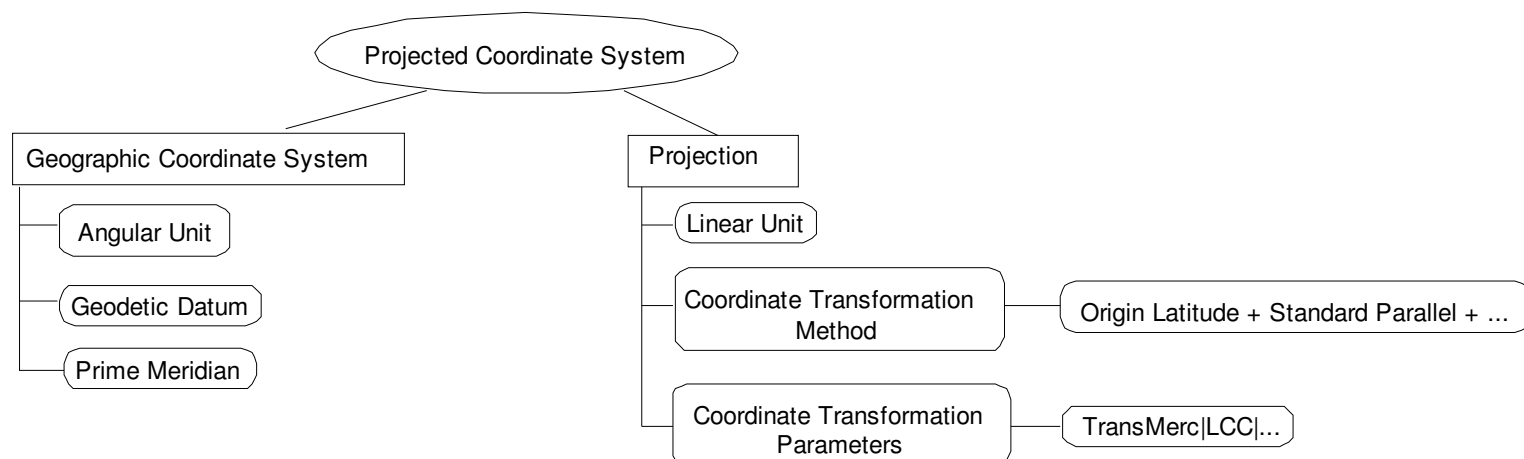
EPSG codes are **standardized numbers** for national and international coordinate reference systems and coordinate transformations – used by many GIS software packages.

E.g., GRASS GIS locations can be generated with **EPSG codes**.
Also used in PROJ4: <http://trac.osgeo.org/proj/> and GDAL: <http://www.gdal.org/>

Codes available as SQL database from <http://www.epsg.org> or
found in the PROJ4 installation at /usr/share/proj/epsg

Useful Web sites:

<http://www.epsg-registry.org> and <http://epsg.io/>





GRASS GIS Location Manager

TASK: Creating a new location with the location manager

Terminology reminder:



- A **LOCATION** is simply a set of directories which contains the GRASS data of a project.
- Within each LOCATION, there are **MAPSET(S)**
 - The mandatory "**PERMANENT**" MAPSET contains projection information and some more definitions. May be used to store the base cartography in it as "PERMANENT" is visible to all users accessing a LOCATION.
 - Other optional MAPSETs contain user based GRASS data

Hint:

External data (GeoTIFF, SHAPE, ...): best stored in a separate directory

GRASS GIS Location Manager



GRASS GIS 7.0.2svn startup (r65855)

 **grass gis**
Bringing advanced geospatial technologies to the world

1. Select GRASS GIS database directory

GRASS GIS database directory contains Location

2. Select GRASS Location

demolocation
ecad5_grassdata_ll
nc_spm_08_grass7
nc_spm_08_reduced
spearfish70

3. Select GRASS Mapset

PERMANENT
user1

All data in one Location is in the same coordinate reference system (projection). One Location can be one project. Location contains Mapsets.

Mapset contains GIS data related to one project, task within one project, subregion or user.

Location Wizard

(subtle hint)

GRASS GIS Location Manager



Define new GRASS Location

Define GRASS Database and Location Name

GIS Data Directory:

Project Location:

Location Title:

Define new GRASS Location

Choose method for creating a new location

☐ Select coordinate system parameters from a list

☐ Select EPSG code of spatial reference system

☒ Read projection and datum terms from a georeferenced data file

☐ Read projection and datum terms from a Well Known Text (WKT) .prj file

☐ Specify projection and datum terms using custom PROJ.4 parameters

☐ Create a generic Cartesian coordinate system (XY)

Create from SHP
Or GeoTIFF file

GRASS GIS Location Manager



Define new GRASS Location

Select georeferenced file

Georeferenced file:

Define new GRASS Location

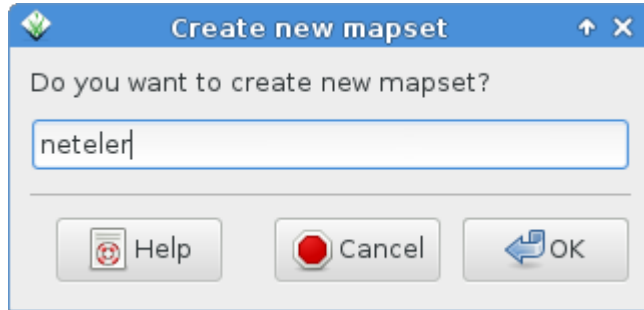
Summary

GRASS Database: /home/neteler/grassdata
Location Name: my_nc_metric
Location Title: North Carolina metric units
Projection: matches file /home/neteler/gis_data/elev_state_500m.tif
PROJ.4 definition (non-definitive):
+proj=lcc
+lat_1=36.16666666666666
+lat_2=34.33333333333334
+lat_0=33.75
+lon_0=-79
+x_0=609601.22
+y_0=0
+no_defs
+a=6378137
+rf=298.257222101
+towgs84=

Import data?

Do you want to import </home/neteler/gis_data/elev_state_500m.tif> to the newly created location? The location's default region will be set from this imported map

GRASS GIS Location Manager

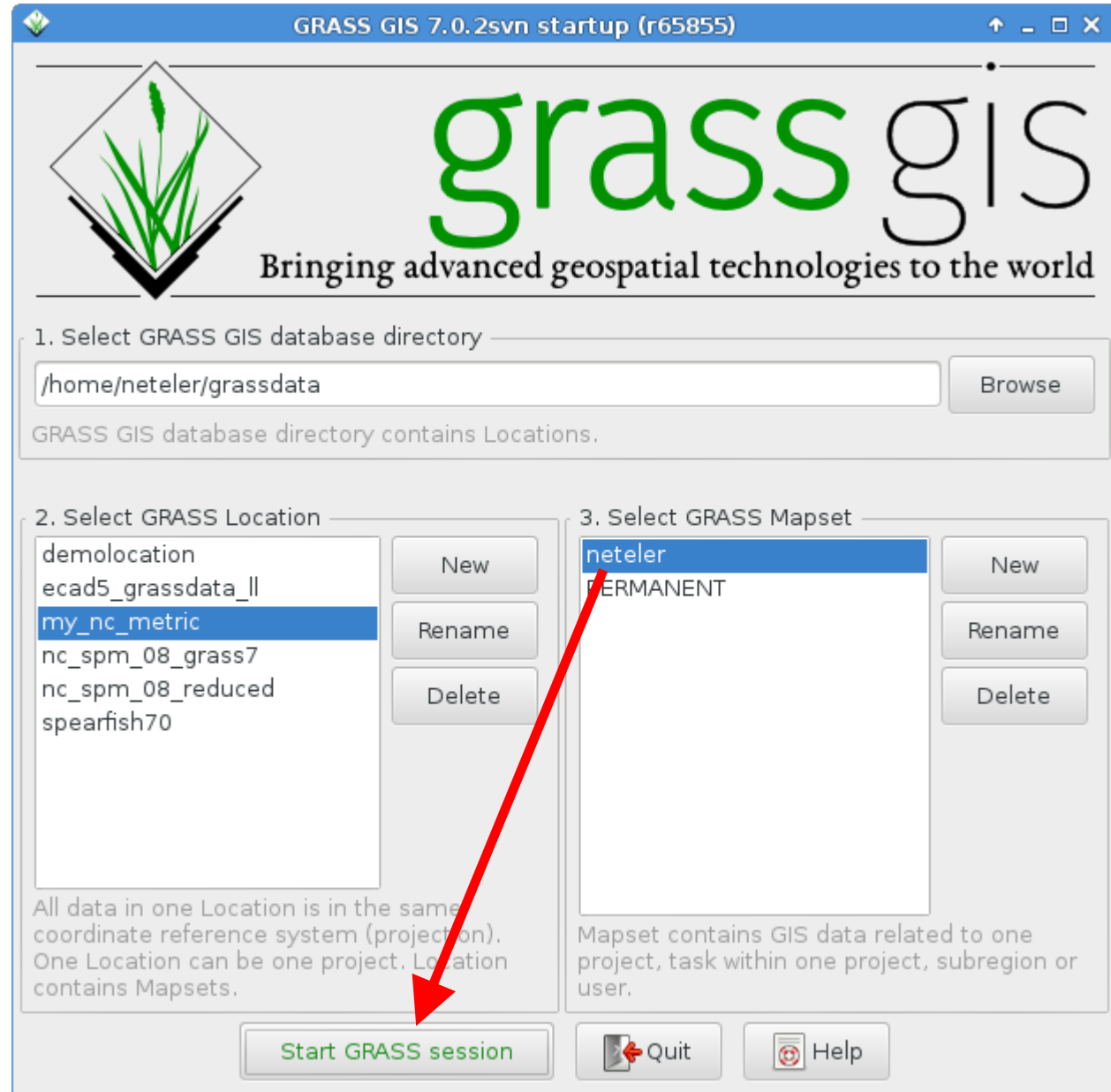


Hint - reasonable mapset names:

- User name
- Place name
- Project name
- ...

Now we are back to the welcome screen and have the new location there.

Ready to use GRASS GIS!





GRASS GIS Location Manager

TASK: Creating a new location with the location manager

Variant 2: using an EPSG code



GRASS GIS Location Manager: Using EPSG code



Define new GRASS Location

Choose method for creating a new location

☐ Select coordinate system parameters from a list

☒ Select EPSG code of spatial reference system

☐ Read projection and datum terms from a georeferenced data file

☐ Read projection and datum terms from a file

☐ Specify projection and datum terms manually

☐ Create a new location without a coordinate system

Define new GRASS Location

Choose EPSG Code

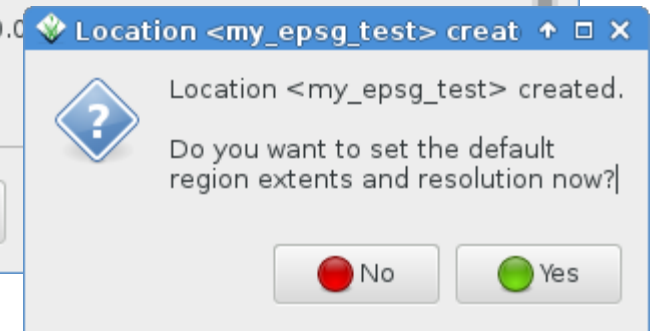
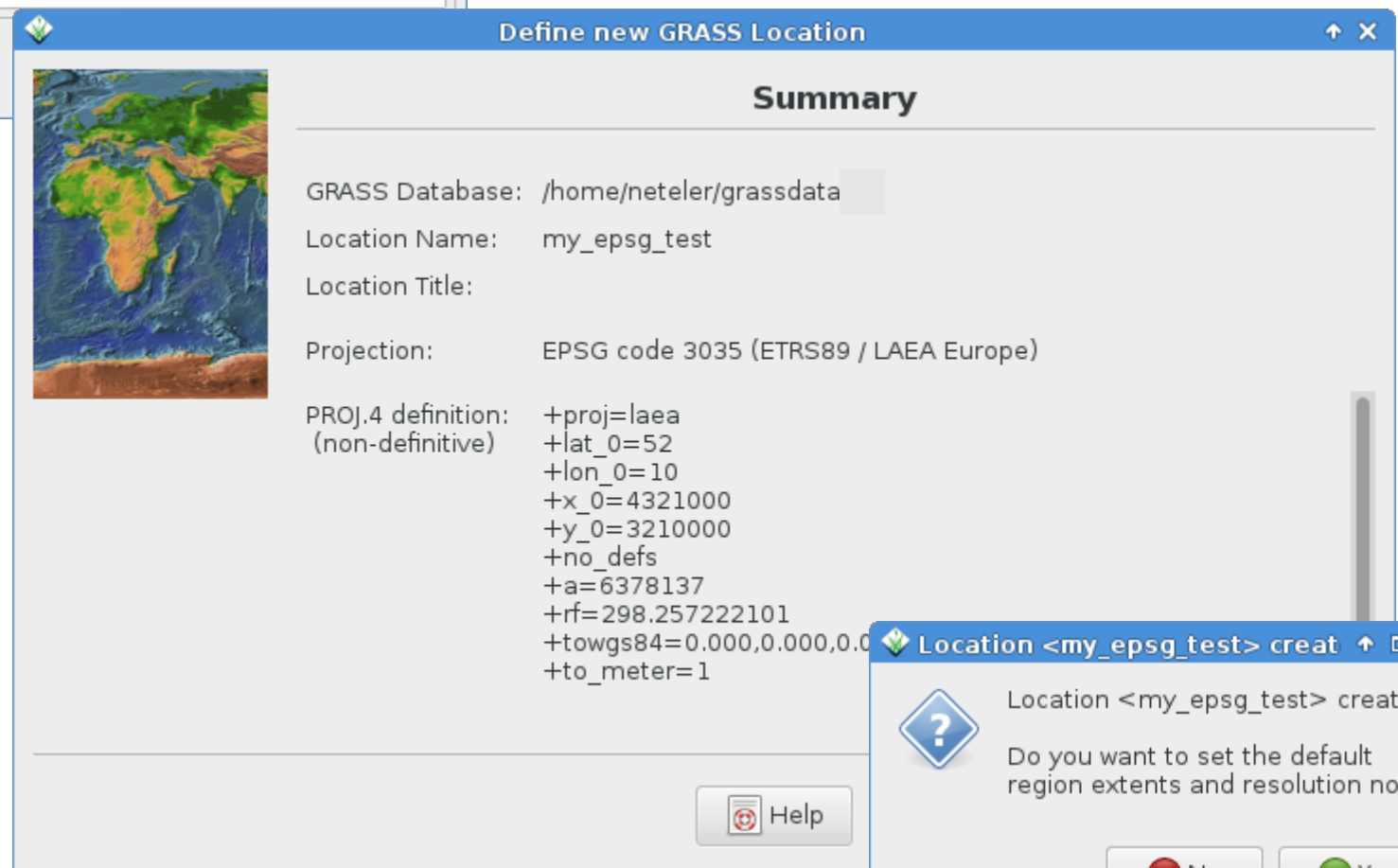
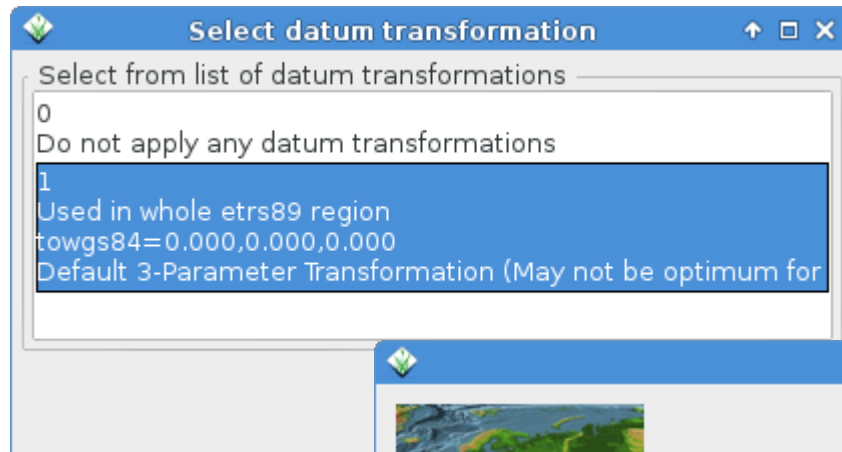
Path to the EPSG-codes file:

EPSG code:

Code	Description	Parameters
2163	US National Atlas Equal Area	+proj=laea +lat_0=4...
3035	ETRS89 / LAEA Europe	+proj=laea +lat_0=5...
3408	NSIDC EASE-Grid North	+proj=laea +lat_0=9...
3409	NSIDC EASE-Grid South	+proj=laea +lat_0=-9...
3571	WGS 84 / North Pole LAEA Bering Sea	+proj=laea +lat_0=9...
3572	WGS 84 / North Pole LAEA Alaska	+proj=laea +lat_0=9...
3573	WGS 84 / North Pole LAEA Canada	+proj=laea +lat_0=9...
3574	WGS 84 / North Pole LAEA Atlantic	+proj=laea +lat_0=9...

LAEA as
search string

GRASS GIS Location Manager: Using EPSG code



GRASS GIS Location Manager: Using EPSG code

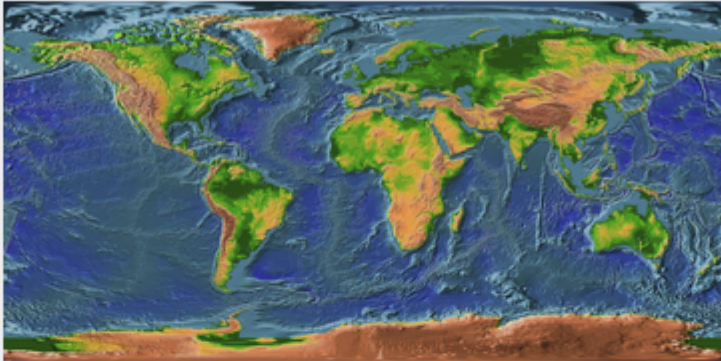


Optionally: Setting manually the default region

Set default region extent and resolution

North
4256100.0

West 408000.0 East 643230.0



South
4043670.0

N-S resolution
50
Rows: 4248

E-W resolution
50
Cols: 4704

Cells: 19982592

☐ Click here to hide 3D settings

Top 1.0 Bottom 0.0 T-B resolution 1.0

Depth: 1 3D Cells: 19982592

GRASS GIS Location Manager



GRASS GIS 7.0.2svn startup (r65855)

 **grass gis**
Bringing advanced geospatial technologies to the world

1. Select GRASS GIS database directory

GRASS GIS database directory contains Locations.

2. Select GRASS Location

3. Select GRASS Mapset

All data in one Location is in the same coordinate reference system (projection). One Location can be one project. Location contains Mapsets.

Mapset contains GIS data related to one project, task within one project, subregion or user.

The new location is ready!



Creating an own GRASS Location

Creating a new location from command line 1/2

Examples:

```
# creating a new location from an EPSG code  
grass70 -c epsg:4326 ~/grassdata/latlong_wgs84
```

```
# creating a new location from a SHAPE file  
grass70 -c NUTS_BN_03M_2013.shp ~/grassdata/eu_etr89
```

```
# creating a new location from a GeoTIFF file  
grass70 -c EUD_CP-DEMS_3500035000-AA.tif ~/grassdata/eu_laea
```




Creating an own GRASS Location

Creating a new location from command line 2/2

Example for using the --exec interface:

<https://grass.osgeo.org/grass72/manuals/grass7.html#batch-jobs-with-the-exec-interface>

```
# creating a new location from a GeoTIFF file + import:  
grass72 -c elevation.tif ~/grassdata/test1/ --exec r.import \  
    input=elevation.tif output=elevation
```

Instead of a GRASS GIS command you can also specify a script!



Creating an own GRASS Location

Later, in the new location, to set region extent...

Options:

- g.region (or wxGUI → Settings) – knowing the boundaries
- r.in.gdal (during data import) – from raster map
- v.in.ogr (during data import) – from vector map