

# How to create a new module

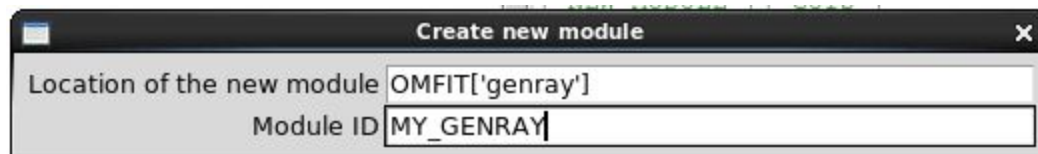
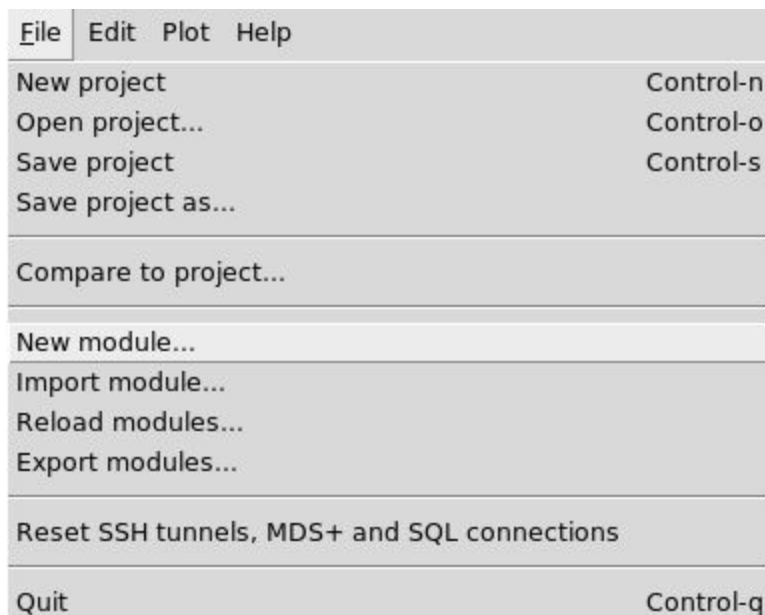
---

## DISCLAIMER:

This is an **editable public document!** When you make a change to it, it will be visible immediately by everyone! Feel free to edit it and help other OMFIT users!

Since this is an evolving document, there may be some small inconsistencies as different figures have been taken by different people with different versions of OMFIT for different analyses.

---





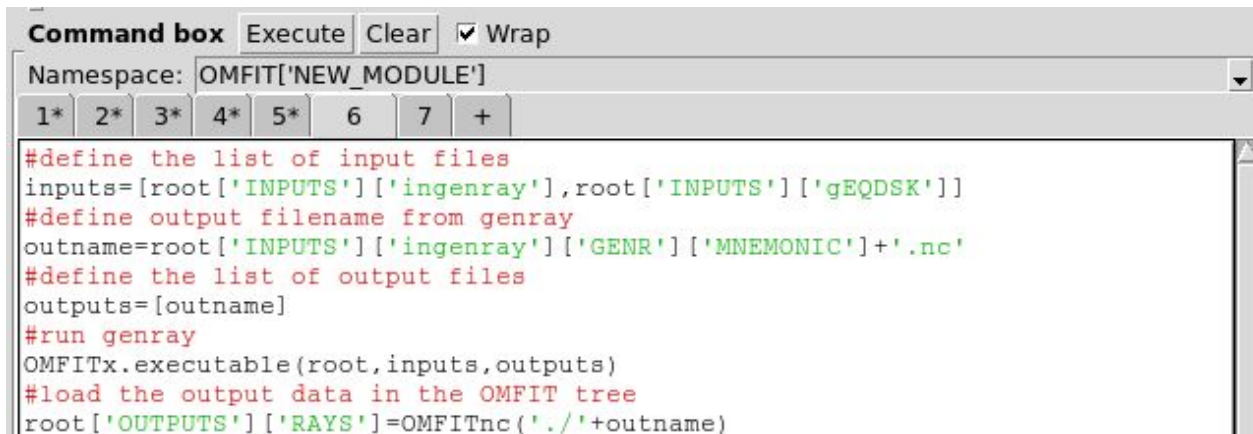
Set the executable by right-clicking on the 'executable' entry, selecting 'modify tree entry', and set the entry value:

```
OMFIT['genray']['SETTINGS']['SETUP']['executable']= 'module purge;  
module load genray; xgenray'
```

Load some existing gEQDSK and GENRAY input namelist in the module:



Now let's write a small script to execute the code (note the namespace)



The screenshot shows a 'Command box' window with a title bar containing 'Execute', 'Clear', and a checked 'Wrap' checkbox. Below the title bar, the 'Namespace' is set to 'OMFIT[NEW\_MODULE]'. The main area contains a script with the following code:

```
#define the list of input files
inputs=[root['INPUTS']['ingenray'],root['INPUTS']['gEQDSK']]
#define output filename from genray
outname=root['INPUTS']['ingenray']['GENR']['MNEMONIC']+'.nc'
#define the list of output files
outputs=[outname]
#run genray
OMFITx.executable(root,inputs,outputs)
#load the output data in the OMFIT tree
root['OUTPUTS']['RAYS']=OMFITnc('./'+outname)
```

```
#define the list of input files
inputs=[root['INPUTS']['ingenray'],root['INPUTS']['gEQDSK']]
#define output filename from genray
outname=root['INPUTS']['ingenray']['GENR']['MNEMONIC']+'.nc'
#define the list of output files
outputs=[outname]
#run genray
OMFITx.executable(root,inputs,outputs)
#load the output data in the OMFIT tree
root['OUTPUTS']['RAYS']=OMFITnc('./'+outname)
```

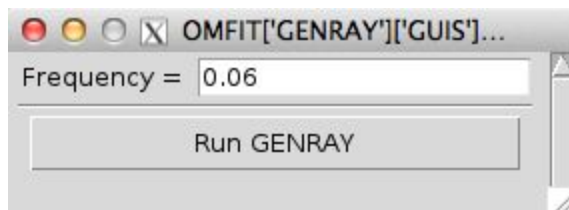
Let's execute the script.

If successful we can save the script under `root['SCRIPTS']['runGENRAY']` by right clicking in the command box, then choosing "Save to tree"

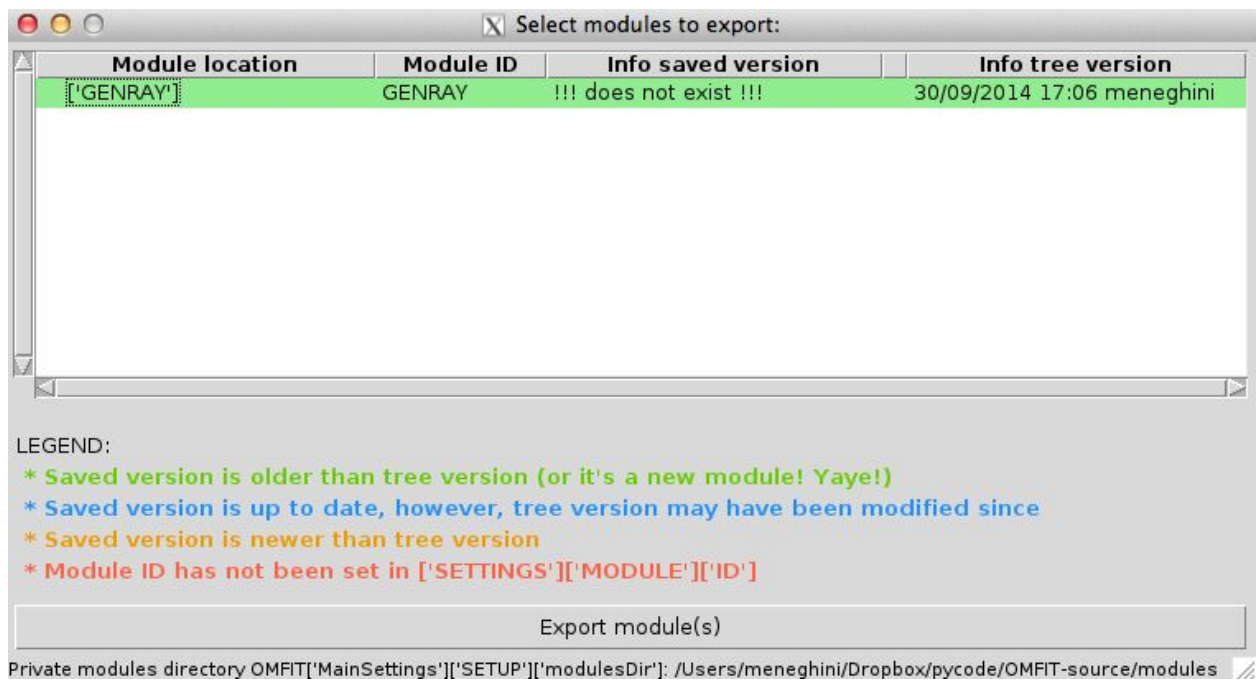
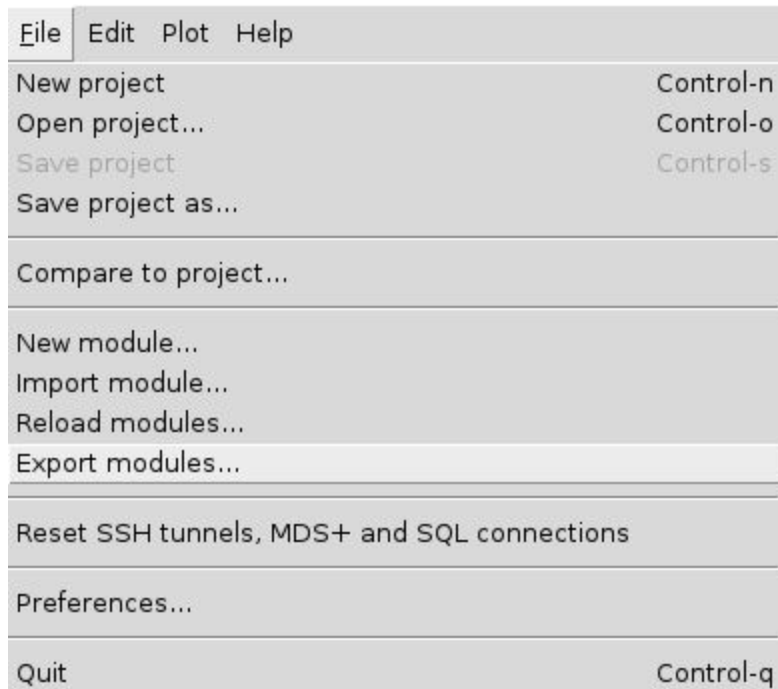
To create a GUI:

The screenshot shows a configuration dialog box for OMFIT. At the top, the 'Entry location' is set to 'OMFIT[GENRAY][GUIS][GENRAYgui]' and the 'Entry value' is 'OMFITpythonGUI("GENRAYgui.py")'. Below these fields are several options: 'Existing file...', 'Remote file...', 'is string', 'Modify original file', and 'Dynamic expression'. A large list of radio buttons follows, organized in a grid. The 'Python GUI' option is selected. Other options include 'Int', 'None', 'Namelist', 'CSV', 'File', 'Python pickle', 'EQDSK (g,a,m)', 'Osborne pFile', 'MDS+ tree', 'Python task', 'OMFIT tree', 'User defined class', 'Float', 'Linspace', 'NetCDF', 'ASCII table', 'ASCII file', 'GA code input', 'TGYRO dir', 'MDS+ value', 'OMFIT module', 'Complex', 'Array', 'IDL save', 'INI config', 'Directory', 'LLNL pdb', 'ONETWO outone', 'SQL database', 'Python plot', 'OMFIT project', 'Uncertainty', 'Matrix', 'IDL language', 'MATLAB', 'Web link', 'Kepler actor', 'NIMROD', 'ITM ual', and 'Python class'.

```
OMFITx.Entry("root['FILES']['ingenray']['WAVE']['FRQNCY']", 'Frequency')
OMFITx.Separator()
OMFITx.Button('Run GENRAY', root['SCRIPTS']['GENRAYexec'].run)
```



To export the new module:



Then choose the module and click "Export module(s)".