Managing modules in OMFIT

DISCLAMER:

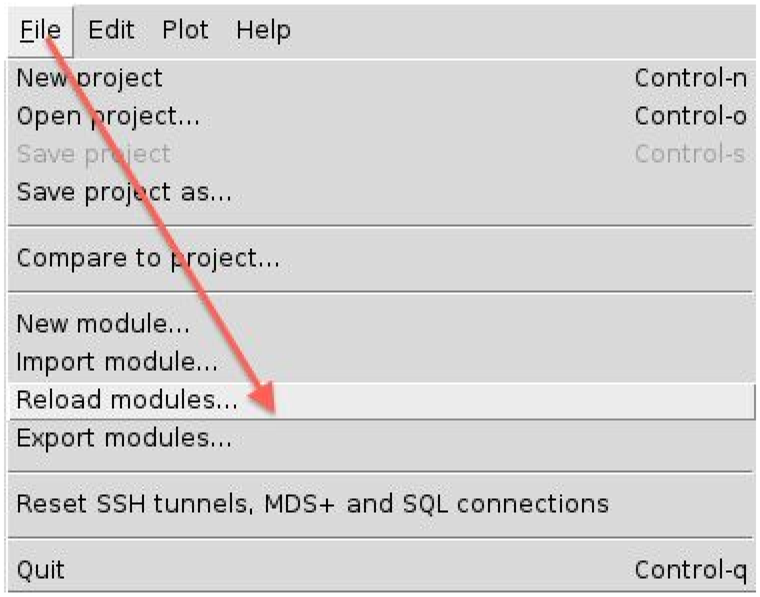
This is an editable public document! When you make a change to it, it will be visible immediately from everyone! Feel free to edit it and help other OMFIT users!

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.

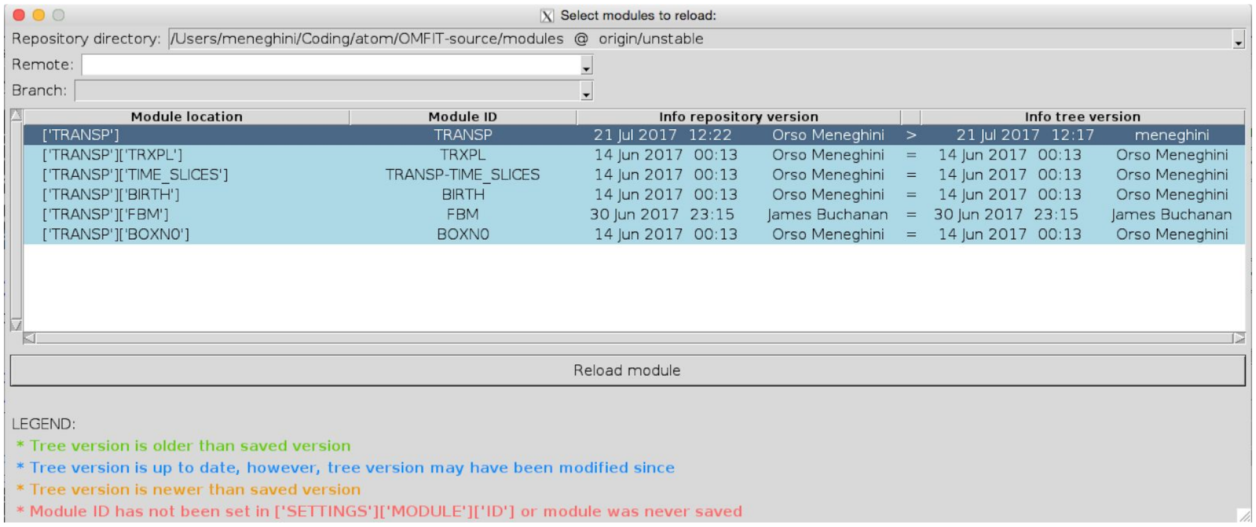
When saving a project, OMFIT will save all the data and python scripts in your tree. This is great, because you can always go back and reproduce your old results, and changes that people may make to the public modules will not affect how your project works.
However, it may happen that you may want to update one or more modules in your project if these become obsolete. This is when you want to “reload a module”.

Reloading a module

To start, go to `File > Reload modules...`



This will pop-up a new window, from which you will be able to select what modules you want to reload [multiple modules can be selected using shift-click (continuous selection) or control-click (selective selection)]. For our example, we want to reload the ONETWO module, which the GUI tells us has become obsolete (it's marked in green -> it's safe to reload).



Select modules to reload:

Repository directory: /Users/meneghini/Coding/atom/OMFIT-source/modules @ origin/unstable

Remote:

Branch:

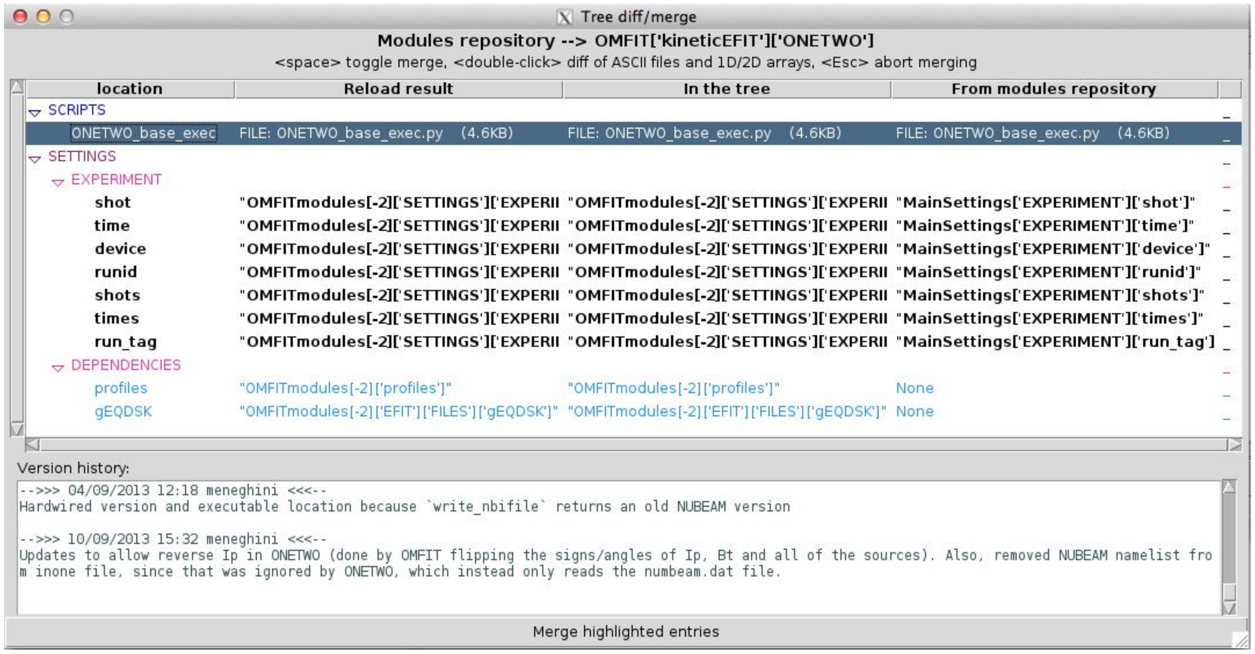
Module location	Module ID	Info repository version			Info tree version	
['TRANSP']	TRANSP	21 Jul 2017 12:22	Orso Meneghini	>	21 Jul 2017 12:17	meneghini
['TRANSP']['TRXPL']	TRXPL	14 Jun 2017 00:13	Orso Meneghini	=	14 Jun 2017 00:13	Orso Meneghini
['TRANSP']['TIME_SLICES']	TRANSP-TIME_SLICES	14 Jun 2017 00:13	Orso Meneghini	=	14 Jun 2017 00:13	Orso Meneghini
['TRANSP']['BIRTH']	BIRTH	14 Jun 2017 00:13	Orso Meneghini	=	14 Jun 2017 00:13	Orso Meneghini
['TRANSP']['FBM']	FBM	30 Jun 2017 23:15	James Buchanan	=	30 Jun 2017 23:15	James Buchanan
['TRANSP']['BOXN0']	BOXN0	14 Jun 2017 00:13	Orso Meneghini	=	14 Jun 2017 00:13	Orso Meneghini

Reload module

LEGEND:
* Tree version is older than saved version
* Tree version is up to date, however, tree version may have been modified since
* Tree version is newer than saved version
* Module ID has not been set in ['SETTINGS']['MODULE']['ID'] or module was never saved

The modules will be reloaded from the OMFIT installation that it is running, however modules can be reloaded from branches on remote GitHub OMFIT-source repositories as well. See the following tutorial on how to access github directly from within OMFIT.

Once you have selected the modules we want to reload, you can click on the `Reload, compare & merge` button. This will pop-up a new window, showing the differences between the tree version of the module and the module repository version. Notice that only the differences between the two are shown.



Tree diff/merge

Modules repository --> OMFIT['kineticEFIT']['ONETWO']

<space> toggle merge, <double-click> diff of ASCII files and 1D/2D arrays, <Esc> abort merging

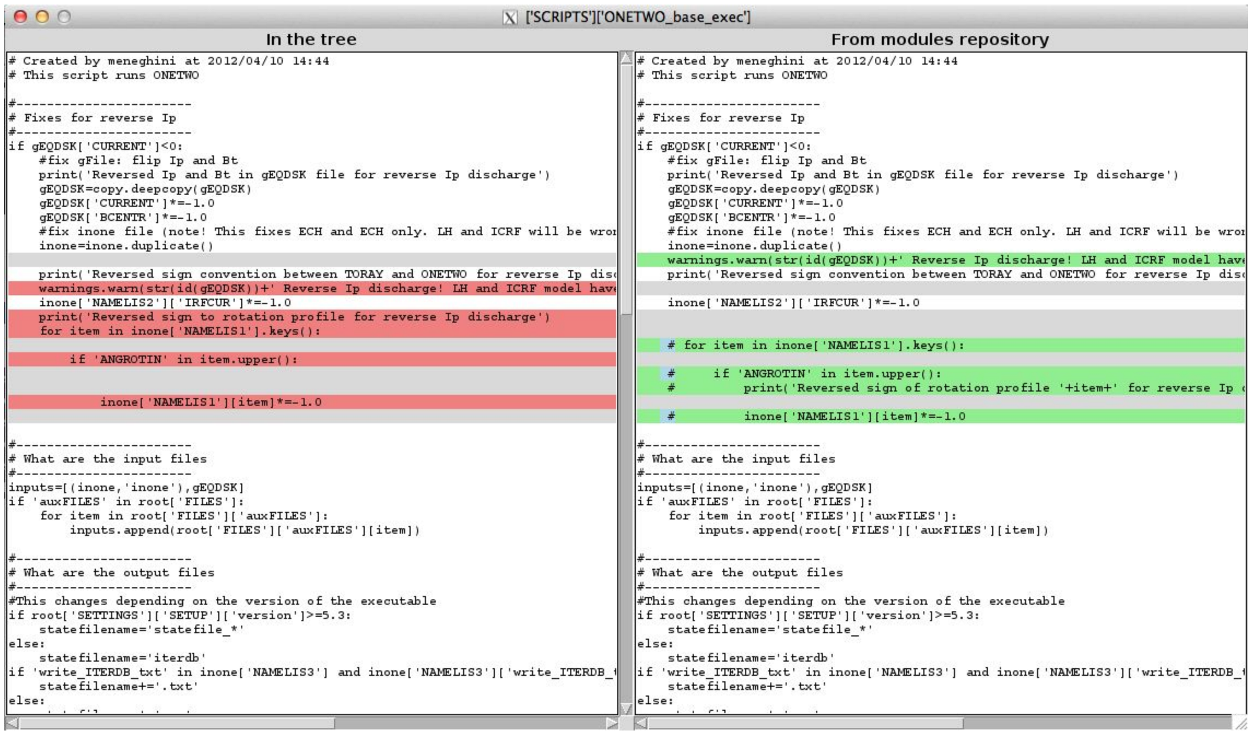
location	Reload result	In the tree	From modules repository
SCRIPTS			
ONETWO_base_exec	FILE: ONETWO_base_exec.py (4.6KB)	FILE: ONETWO_base_exec.py (4.6KB)	FILE: ONETWO_base_exec.py (4.6KB)
SETTINGS			
EXPERIMENT			
shot	"OMFITmodules[-2]['SETTINGS']['EXPERII	"OMFITmodules[-2]['SETTINGS']['EXPERII	"MainSettings['EXPERIMENT']['shot']"
time	"OMFITmodules[-2]['SETTINGS']['EXPERII	"OMFITmodules[-2]['SETTINGS']['EXPERII	"MainSettings['EXPERIMENT']['time']"
device	"OMFITmodules[-2]['SETTINGS']['EXPERII	"OMFITmodules[-2]['SETTINGS']['EXPERII	"MainSettings['EXPERIMENT']['device']"
runid	"OMFITmodules[-2]['SETTINGS']['EXPERII	"OMFITmodules[-2]['SETTINGS']['EXPERII	"MainSettings['EXPERIMENT']['runid']"
shots	"OMFITmodules[-2]['SETTINGS']['EXPERII	"OMFITmodules[-2]['SETTINGS']['EXPERII	"MainSettings['EXPERIMENT']['shots']"
times	"OMFITmodules[-2]['SETTINGS']['EXPERII	"OMFITmodules[-2]['SETTINGS']['EXPERII	"MainSettings['EXPERIMENT']['times']"
run_tag	"OMFITmodules[-2]['SETTINGS']['EXPERII	"OMFITmodules[-2]['SETTINGS']['EXPERII	"MainSettings['EXPERIMENT']['run_tag']
DEPENDENCIES			
profiles	"OMFITmodules[-2]['profiles']"	"OMFITmodules[-2]['profiles']"	None
gEQDSK	"OMFITmodules[-2]['EFIT']['FILES']['gEQDSK']"	"OMFITmodules[-2]['EFIT']['FILES']['gEQDSK']"	None

Version history:

```
-->>> 04/09/2013 12:18 meneghini <<<--
Hardwired version and executable location because `write_nbifile` returns an old NUBEAM version

-->>> 10/09/2013 15:32 meneghini <<<--
Updates to allow reverse Ip in ONETWO (done by OMFIT flipping the signs/angles of Ip, Bt and all of the sources). Also, removed NUBEAM namelist fro
m inone file, since that was ignored by ONETWO, which instead only reads the numbeam.dat file.
```

Merge highlighted entries

One can check what changes were made to a specific python script by `<Double-click>` on it. This will open up a different GUI showing the differences. Similarly, differences can be visualized for 1D and 2D arrays…



In the tree

```
# Created by meneghini at 2012/04/10 14:44
# This script runs ONETWO

#-----------------------
# Fixes for reverse Ip
#-----------------------
if gEQDSK['CURRENT']<0:
    #fix gFile: flip Ip and Bt
    print('Reversed Ip and Bt in gEQDSK file for reverse Ip discharge')
    gEQDSK=copy.deepcopy(gEQDSK)
    gEQDSK['CURRENT']*=-1.0
    gEQDSK['BCENTR']*=-1.0
    #fix inone file (note! This fixes ECH and ECH only. LH and ICRF will be wro
    inone=inone.duplicate()

    print('Reversed sign convention between TORAY and ONETWO for reverse Ip dis
    warnings.warn(str(id(gEQDSK))+' Reverse Ip discharge! LH and ICRF model hav
    inone['NAMELIS2']['IRFCUR']*=-1.0
    print('Reversed sign to rotation profile for reverse Ip discharge')
    for item in inone['NAMELIS1'].keys():

        if 'ANGROTIN' in item.upper():


            inone['NAMELIS1'][item]*=-1.0


#-----------------------
# What are the input files
#-----------------------
inputs=[(inone,'inone'),gEQDSK]
if 'auxFILES' in root['FILES']:
    for item in root['FILES']['auxFILES']:
        inputs.append(root['FILES']['auxFILES'][item])

#-----------------------
# What are the output files
#-----------------------
#This changes depending on the version of the executable
if root['SETTINGS']['SETUP']['version']>=5.3:
    statefilename='statefile_*'
else:
    statefilename='iterdb'
if 'write_ITERDB_txt' in inone['NAMELIS3'] and inone['NAMELIS3']['write_ITERDB_
    statefilename+='.txt'
else:
```

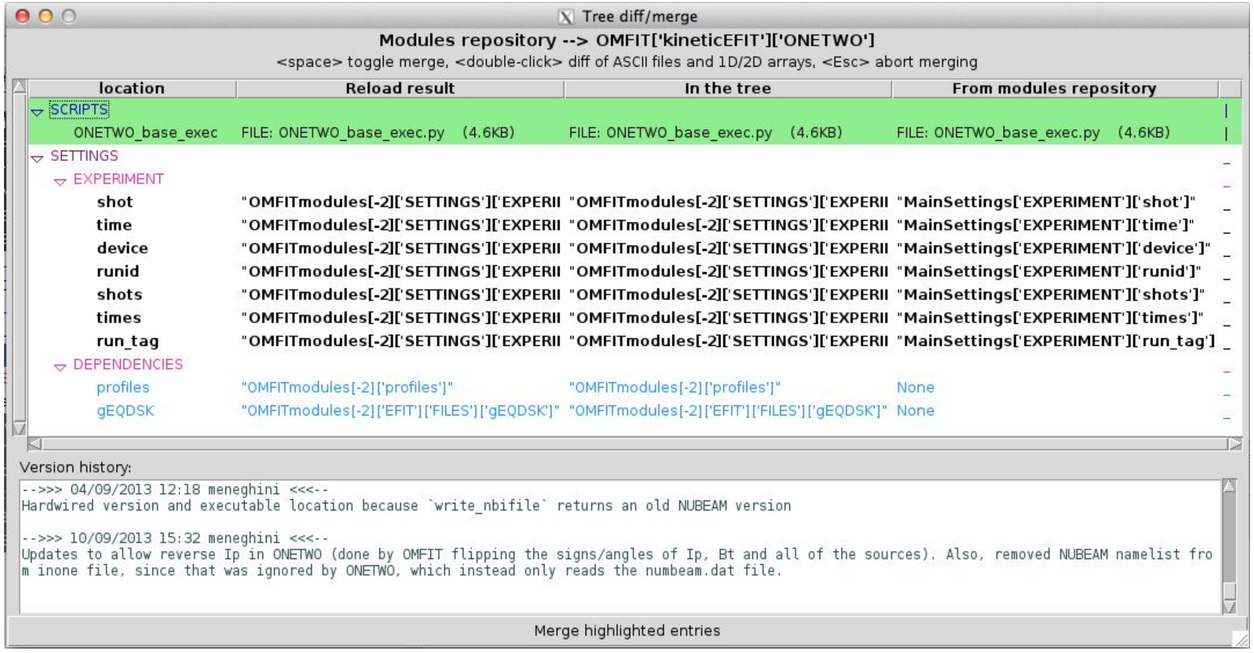
From modules repository

```
# Created by meneghini at 2012/04/10 14:44
# This script runs ONETWO

#-----------------------
# Fixes for reverse Ip
#-----------------------
if gEQDSK['CURRENT']<0:
    #fix gFile: flip Ip and Bt
    print('Reversed Ip and Bt in gEQDSK file for reverse Ip discharge')
    gEQDSK=copy.deepcopy(gEQDSK)
    gEQDSK['CURRENT']*=-1.0
    gEQDSK['BCENTR']*=-1.0
    #fix inone file (note! This fixes ECH and ECH only. LH and ICRF will be wro
    inone=inone.duplicate()
    warnings.warn(str(id(gEQDSK))+' Reverse Ip discharge! LH and ICRF model hav
    print('Reversed sign convention between TORAY and ONETWO for reverse Ip dis

    inone['NAMELIS2']['IRFCUR']*=-1.0


    # for item in inone['NAMELIS1'].keys():

    #     if 'ANGROTIN' in item.upper():
    #         print('Reversed sign of rotation profile '+item+' for reverse Ip 

    #         inone['NAMELIS1'][item]*=-1.0

#-----------------------
# What are the input files
#-----------------------
inputs=[(inone,'inone'),gEQDSK]
if 'auxFILES' in root['FILES']:
    for item in root['FILES']['auxFILES']:
        inputs.append(root['FILES']['auxFILES'][item])

#-----------------------
# What are the output files
#-----------------------
#This changes depending on the version of the executable
if root['SETTINGS']['SETUP']['version']>=5.3:
    statefilename='statefile_*'
else:
    statefilename='iterdb'
if 'write_ITERDB_txt' in inone['NAMELIS3'] and inone['NAMELIS3']['write_ITERDB_
    statefilename+='.txt'
else:
```

To choose what differences to reload from the modules repository, you should click on that entry and press `<Spacebar>`. This will highlight that entry in green. Only what is highlighted in green will be reloaded from the modules repository into the OMFIT tree.

Not all of the differences need to be imported from the repository into the OMFIT tree. Here, some basic understanding of how a specific module works may be necessary!

In general, one may want to reload the `scripts`, the `GUIs` and `plots`. Some of these may need some new variables defined in the `['SETTINGS']['SETUP']` or `['SETTINGS']['PHYSICS']` subtrees. Pay attention, that a likely difference between your tree and the module is that your tree may have some data (for example under `['FILES']`, `['INPUTS']` or `['OUTPUTS']` subtrees), while the modules will have no data. Taking these subtrees from the modules repository will delete the data, which is not necessarily what you want!


Tree diff/merge

Modules repository --> OMFIT['kineticEFIT']['ONETWO']

<space> toggle merge, <double-click> diff of ASCII files and 1D/2D arrays, <Esc> abort merging

location	Reload result	In the tree	From modules repository
SCRIPTS			
ONETWO_base_exec	FILE: ONETWO_base_exec.py (4.6KB)	FILE: ONETWO_base_exec.py (4.6KB)	FILE: ONETWO_base_exec.py (4.6KB)
SETTINGS			
EXPERIMENT			
shot	"OMFITmodules[-2]['SETTINGS']['EXPERII	"OMFITmodules[-2]['SETTINGS']['EXPERII	"MainSettings['EXPERIMENT']['shot']"
time	"OMFITmodules[-2]['SETTINGS']['EXPERII	"OMFITmodules[-2]['SETTINGS']['EXPERII	"MainSettings['EXPERIMENT']['time']"
device	"OMFITmodules[-2]['SETTINGS']['EXPERII	"OMFITmodules[-2]['SETTINGS']['EXPERII	"MainSettings['EXPERIMENT']['device']"
runid	"OMFITmodules[-2]['SETTINGS']['EXPERII	"OMFITmodules[-2]['SETTINGS']['EXPERII	"MainSettings['EXPERIMENT']['runid']"
shots	"OMFITmodules[-2]['SETTINGS']['EXPERII	"OMFITmodules[-2]['SETTINGS']['EXPERII	"MainSettings['EXPERIMENT']['shots']"
times	"OMFITmodules[-2]['SETTINGS']['EXPERII	"OMFITmodules[-2]['SETTINGS']['EXPERII	"MainSettings['EXPERIMENT']['times']"
run_tag	"OMFITmodules[-2]['SETTINGS']['EXPERII	"OMFITmodules[-2]['SETTINGS']['EXPERII	"MainSettings['EXPERIMENT']['run_tag']
DEPENDENCIES			
profiles	"OMFITmodules[-2]['profiles']"	"OMFITmodules[-2]['profiles']"	None
gEQDSK	"OMFITmodules[-2]['EFIT']['FILES']['gEQDSK']"	"OMFITmodules[-2]['EFIT']['FILES']['gEQDSK']"	None

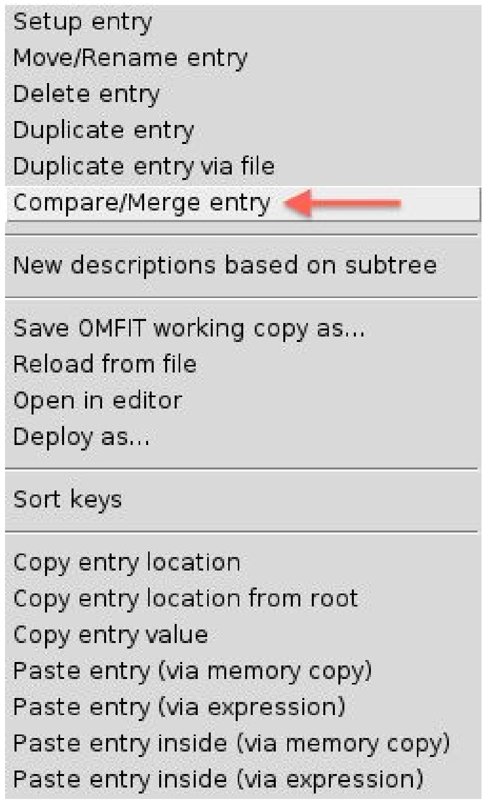
Version history:

```
-->>> 04/09/2013 12:18 meneghini <<<--
Hardwired version and executable location because `write_nbifile` returns an old NUBEAM version

-->>> 10/09/2013 15:32 meneghini <<<--
Updates to allow reverse Ip in ONETWO (done by OMFIT flipping the signs/angles of Ip, Bt and all of the sources). Also, removed NUBEAM namelist fro
m inone file, since that was ignored by ONETWO, which instead only reads the numbeam.dat file.
```

Press `<Return>` to proceed with the merge, meaning merge the entries that you have selected from the modules repository into the OMFIT tree of your project.

Compare/merge subtrees

A similar procedure can be used to compare and import differences between two subtrees within the same project (e.g. compare two namelists, see the differences and selectively pick these differences...). To do this, `<Right-click>` on any subtree element in OMFIT and click on the `Compare/Merge entry` from the pop-up menu.


Setup entry
Move/Rename entry
Delete entry
Duplicate entry
Duplicate entry via file
Compare/Merge entry
New descriptions based on subtree
Save OMFIT working copy as...
Reload from file
Open in editor
Deploy as...
Sort keys
Copy entry location
Copy entry location from root
Copy entry value
Paste entry (via memory copy)
Paste entry (via expression)
Paste entry inside (via memory copy)
Paste entry inside (via expression)