

Status of the ETS

- The version in the repository is the definitive version
 - Any changes/additions should be made to this version
 - Will be switching from “solps-mdsplus.aug.ipp.mpg.de” to ITM-gforge
 - Anybody needing access should use gforge to join the ETS project or contact me (David.Coster@ipp.mpg.de)
- Features
 - Uses CPOs
 - Not yet in Kepler
 - Uses “itm_types” and “itm_constants”
 - Runs and produces the same answer on
 - Gateway (PGI, g95, gfortran)
 - JET (PGI)
 - Garching (Lahey Fujitsu, Intel (32&64), NAG)
 - Useful for finding bugs and non-standard Fortran

Need to have

- A standard “Makefile” structure
- A standard repository layout
- A standard layout of the code and sources
- A standard for documentation
 - doxygen?
- A standard module providing Fortran 95 types
 - Equivalent for C & C++ also?
- A standard module providing Fortran 95 physics constants
 - Equivalent for C & C++ also?



Proposals

- The following are proposals based on
 - ETS
 - AMNS prototype implementation
- These are here to provoke discussion
 - Not to try and dictate a particular implementation

Makefile proposal

- Should provide
 - Support multiple systems with different configurations
 - Compile option
 - Run a test case option
 - “clean” (remove old binaries)
 - “depend” (update dependencies)
 - “DOC” (produce automatic documentation)
 - “tags” (produce emacs tags)
 - “version” (identifying version)
 - “status” (identifying system information)

ETS Makefile

```

SYS=linux.Fujitsu
-include obj/SYS
OBJECTCODE=obj/${SYS}
DOXYGEN=bin/linux/doxygen
include config/${SYS}
include obj/compile
VPATH=src/solvers src/ets src/types src/analytics src/test
VPATH+=src/schemas

EXCLUDELIST = solver_test.o
ANALYTICS_O = ${filter-out ${EXCLUDELIST},${patsubst
    %.f90,${OBJECTCODE}/%.o,${shell echo src/analytics/*.f90 | sed 's:src/[^\]*/::g'}}}
ETS_O = ${filter-out ${EXCLUDELIST},${patsubst %.f90,${OBJECTCODE}/%.o,${shell
    echo src/ets/*.f90 | sed 's:src/[^\]*/::g'}}}
SOLVER_O = ${filter-out ${EXCLUDELIST},${patsubst
    %.f90,${OBJECTCODE}/%.o,${shell echo src/solvers/*.f90 | sed 's:src/[^\]*/::g'}}}
TEST_O = ${filter-out ${EXCLUDELIST},${patsubst %.f90,${OBJECTCODE}/%.o,${shell
    echo src/test/*.f90 | sed 's:src/[^\]*/::g'}}}
TYPES_O = ${filter-out ${EXCLUDELIST},${patsubst %.f90,${OBJECTCODE}/%.o,${shell
    echo src/types/*.f90 | sed 's:src/[^\]*/::g'}}}
SCHEMAS_O = ${filter-out ${EXCLUDELIST},${patsubst
    %.f90,${OBJECTCODE}/%.o,${shell echo src/schemas/*.f90 | sed 's:src/[^\]*/::g'}}}

ADD_O = ${ETS_O} ${SOLVER_O} ${ANALYTICS_O} ${TYPES_O} ${TEST_O}
    ${SCHEMAS_O}

ifeq (${GOT_PERFLIB}, 'yes')
    perflib = -L${OBJECTCODE} -lperflib
    PERF_O =
else
    VPATH+=src/perf
    PERF_O = ${filter-out ${EXCLUDELIST},${patsubst %.f90,${OBJECTCODE}/%.o,${shell
        echo src/perf/*.f90 | sed 's:src/[^\]*/::g'}}}
    ADD_O += ${PERF_O}
endif

run: ${OBJECTCODE}/solver_test
    @mkdir -p data/OUTPUT
    @time ${OBJECTCODE}/solver_test
    @echo Comparing the current output with an earlier version in data.GW
    @diff -rwq -x .svn data.GW/ data/
    @echo No problems should have been reported
    
```

```

${OBJECTCODE}/solver_test: ${OBJECTCODE}/solver_test.o ${ADD_O}
    ${FC} ${FCOPTS} -o $@ $^ ${perflib}

${OBJECTCODE}/test: ${OBJECTCODE}/test.o ${NEUTRALS_O} ${SOLVER_O}
    ${FC} ${FCOPTS} -o $@ $^ ${perflib}

echo:
    @echo ${ETS_O}
    @echo ${NEUTRALS_O}
    @echo ${SOLVER_O}

clean:
    -rm ${OBJECTCODE}/*.o ${OBJECTCODE}/*.${MOD}
    ${OBJECTCODE}/solver_test

docclean:
    rm -rf doc/html doc/latex doc/man doc/rtf doc/xml

depend:
    obj/sfmake depend -p '$${OBJECTCODE}/' ${INCLUDE} -f
    ${OBJECTCODE}/dependencies src/**.f[F] src/**.f[F]90

tags:
    rm TAGS ; etags src/**.f[F]90

${OBJECTCODE}/dependencies:
    mkdir -p ${OBJECTCODE}
    touch ${OBJECTCODE}/dependencies
    ${MAKE} depend

DOC:
    ${DOXYGEN} doc/Doxyfile
    cd doc/latex/ && ${MAKE}

version:
    @echo
    @echo Version of the project = `svnversion . repos/ETS/trunk/ETS`
    @echo SVN version of the Makefile = `svnversion .`
    @echo dpc `date +%Y-%m-%d`
    @echo

status:
    @echo `date +%Y-%m-%d` ${SYS} `svnversion . repos/ETS/trunk/ETS`
    `hostname` `uname -r -p`

include ${OBJECTCODE}/dependencies
    
```


Proposed repository layout

	RW	RO	Explanation
trunk	Resp Off	Task Collaborators	Used for main line development
branches	Task Collaborators		Used for private development
-tags •internal -1.0_RC1 -1.0_RC2 -... -1.1_RC1	Project leader	Members of the project	Release candidate made available by project leader for testing
-tags •release -1.0 -1.1	Task Force leader	Members of the ITM or Restricted if binary only release	Release made available by the task force leader

Proposed code layout

	Comments
src	Contains source
config	Configurations for different compilers
•obj/ -GW -JET -g95	Object code for different systems
doc	Documentation including automatically produced documentation (doxygen)
test	Test cases including saved reference output

itm_types

!> Module implementing the ITM basic types

!>

!> Source:

!> based on SOLPS b2mod_types.F

!>

!> \author David Coster

!>

!> \version "\$Id: itm_types.f90 86 2009-01-28 14:47:59Z dpc \$"

```
module itm_types
```

```
    INTEGER, PARAMETER :: R8 = SELECTED_REAL_KIND (14)
```

```
    INTEGER, PARAMETER :: R4 = SELECTED_REAL_KIND (6)
```

```
end module itm_types
```




itm_constants

```
!> Module implementing the ITM physics constants
!>
!> Source:
!> based on SOLPS b2mod_constants.F
!> '09/12/07 xpb : source CODATA 2006 (http://www.nist.gov/)'
!>
!> \author David Coster
!>
!> \version "$Id: itm_constants.f90 86 2009-01-28 14:47:59Z dpc $"
```

```
module itm_constants
```

```
  use itm_types
```

```
  real (kind=R8), parameter :: pi=3.141592653589793238462643383280_R8
```

```
  real (kind=R8), parameter :: c=2.99792458e8_R8
```

```
  real (kind=R8), parameter :: me=9.10938215e-31_R8
```

```
  real (kind=R8), parameter :: mp=1.672621637e-27_R8
```

```
  real (kind=R8), parameter :: ev=1.602176487e-19_R8
```

```
  real (kind=R8), parameter :: qe=ev
```

```
  real (kind=R8), parameter :: mu0=4.0e-7_R8*pi
```

```
  real (kind=R8), parameter :: eps0=1.0_R8/(mu0*c*c)
```

```
  real (kind=R8), parameter :: avogr=6.02214179e23_R8
```

```
  real (kind=R8), parameter :: KBolt=1.3806504e-23_R8
```

```
  character*64, parameter :: itm_constants_version='$Id: itm_constants.f90 86 2009-01-28 14:47:59Z dpc $'
```

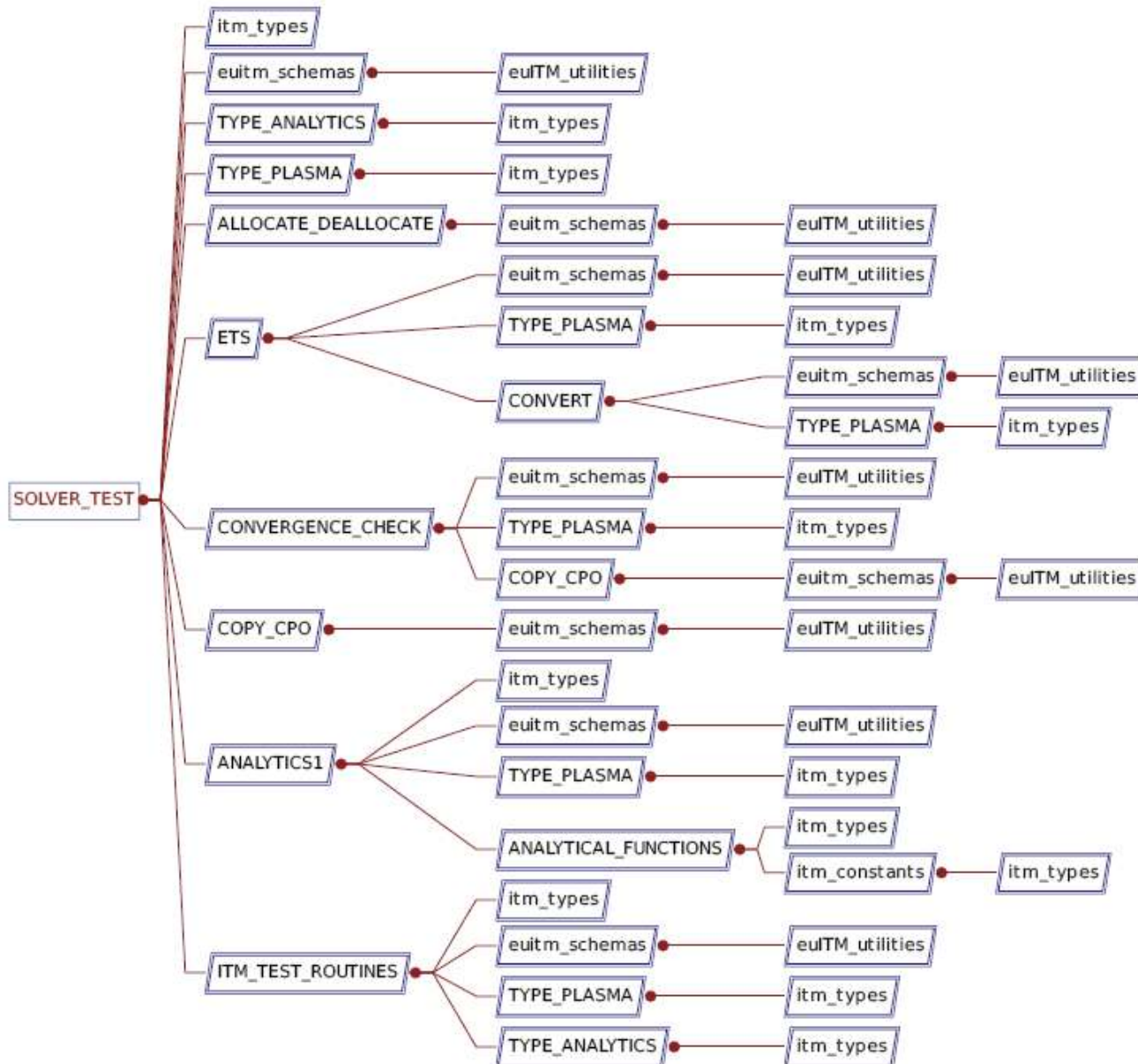
```
end module itm_constants
```



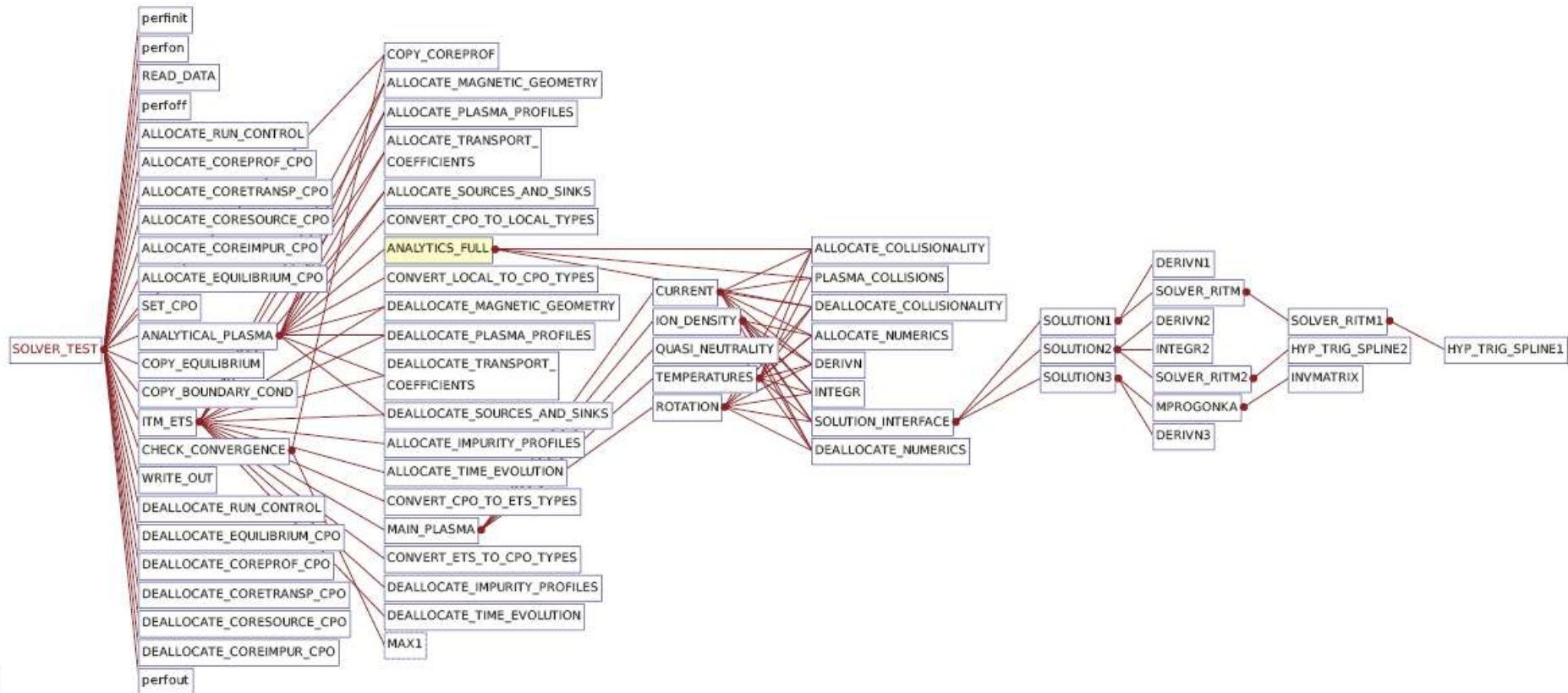
ETS documentation

- doxygen
 - <http://www.efda-itm.eu/~wwwimp3/ETS/>
- Understand for fortran (commercial tool)
 - Graphs on the next couple of pages
 - http://www.efda-itm.eu/~wwwimp3/ETS_Understand/Understand/ETS_html/

ETS Uses Tree

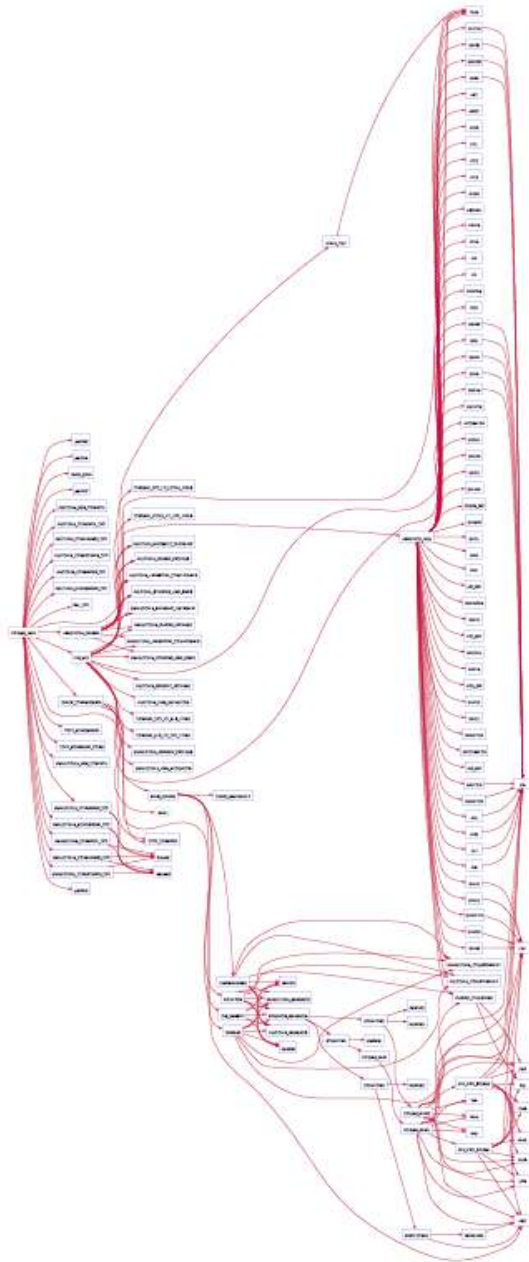


ETS Call Tree





ETS Call Tree



ETS Call Tree

