Leveraging FLASH Architecture for New Problems: Core-Collapse Supernovae

Sean M. Couch
Hubble Fellow, University of Chicago
smc@flash.uchicago.edu

Argonne Training Program on Extreme-Scale Computing
St. Charles, IL, 5 August 2013

FLASH Rules Help

- Exclusive Unit-by-Unit ownership of module-scope date is tremendous
- Use of inheritance, enforced by setup/build system is powerful
- Coding standards, such as naming conventions, “use, ONLY:” statements, etc., tedious but worth it!

Core-Collapse Requirements

- Higher-order shock-capture hydro/MHD
- Self-gravity
- Realistic EOS
- Electron capture physics during collapse
- Neutrino heating/cooling
- Neutrino leakage
- Multiple concurrent setups w/o code duplication

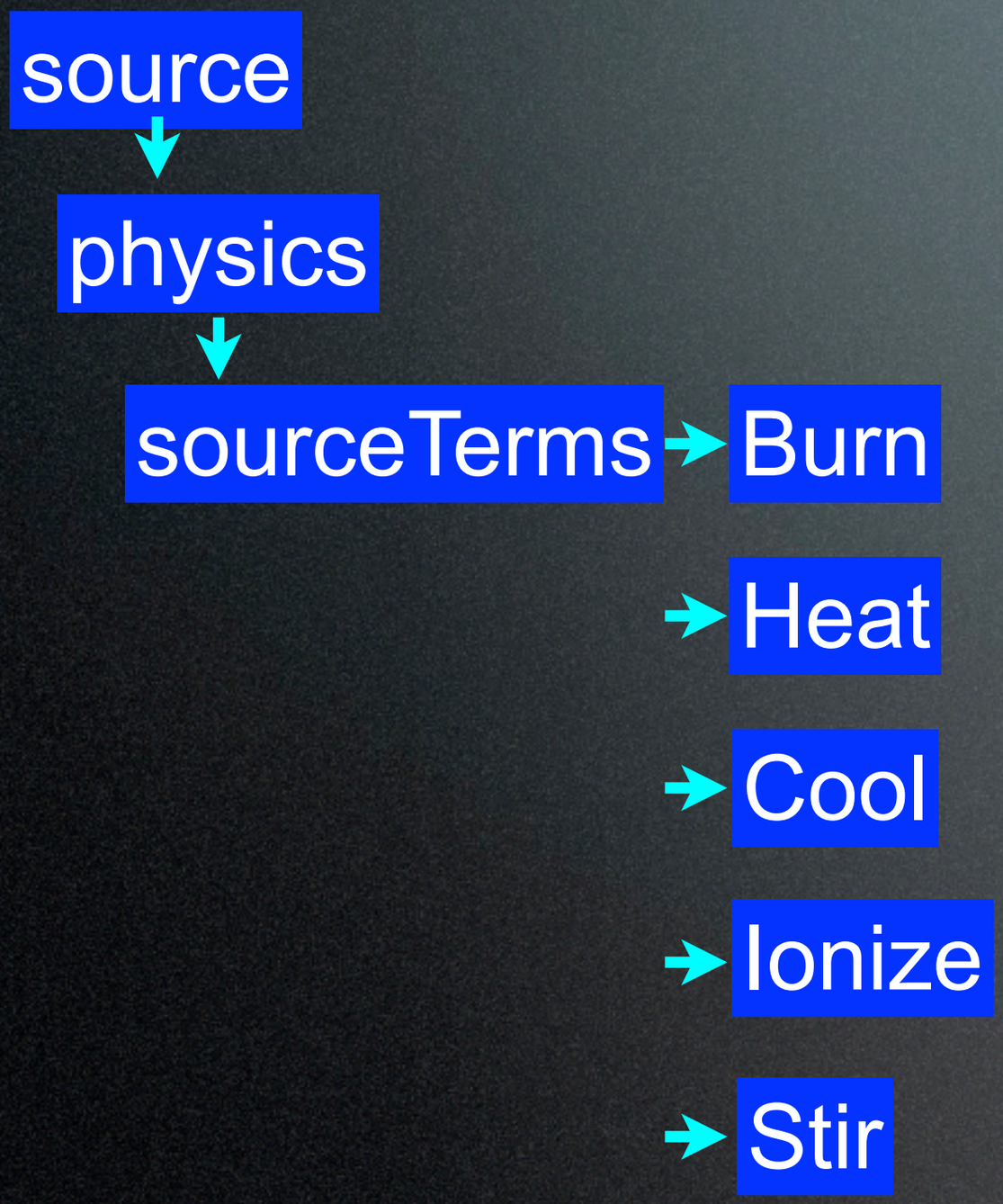
Core-Collapse Requirements

- yes • Higher-order shock-capture hydro/MHD
- yes • Self-gravity
- no • Realistic EOS
- no • Electron capture physics during collapse
- no • Neutrino heating/cooling
- no • Neutrino leakage
- yes • Multiple concurrent setups w/o code duplication

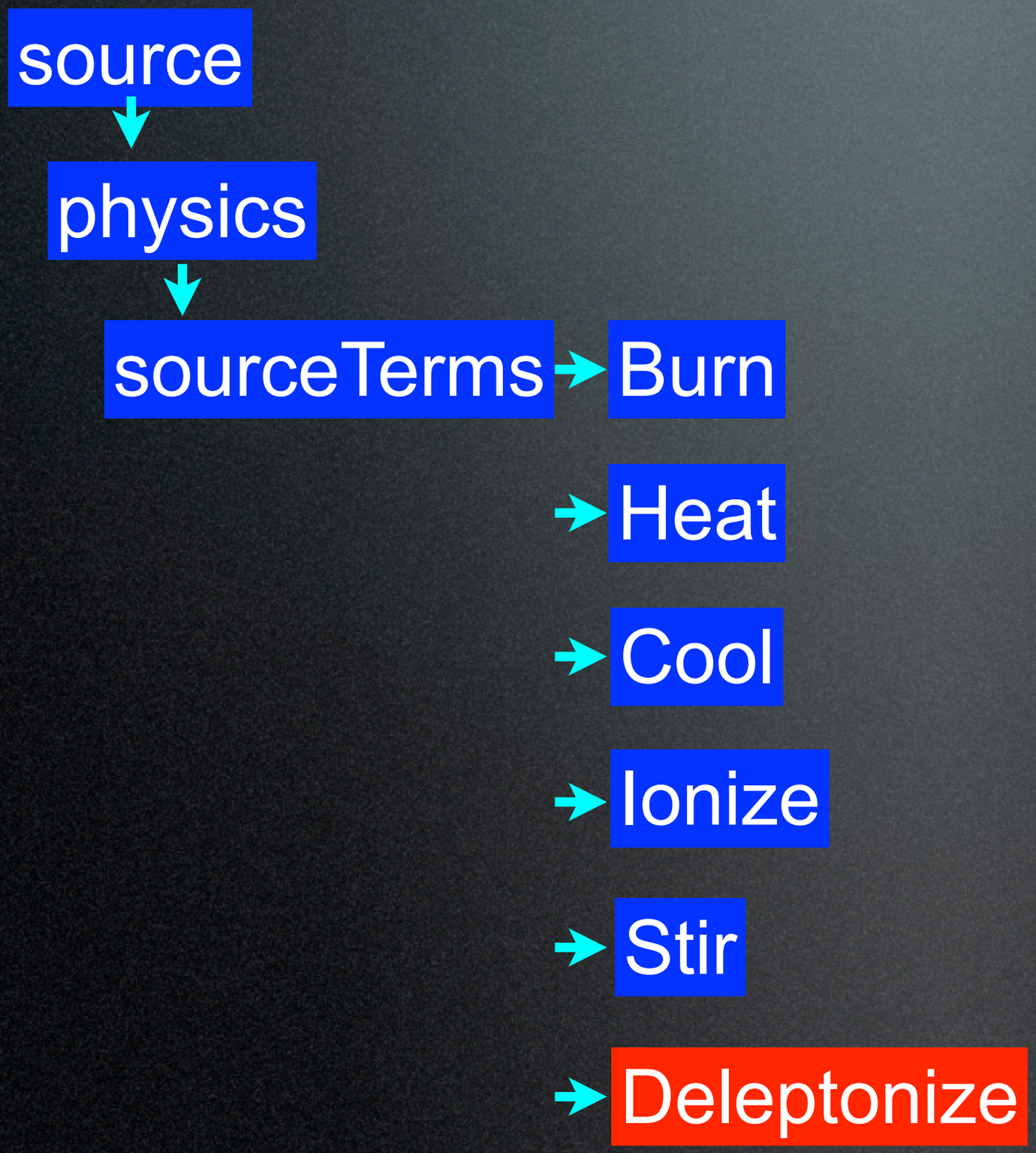
Core-Collapse Requirements

- yes • Higher-order shock-capture hydro/MHD
- yes • Self-gravity
- no • Realistic EOS
- no • Electron capture physics during collapse
- no • Neutrino heating/cooling
- no • Neutrino leakage
- yes • Multiple concurrent setups w/o code duplication

Neutrino Source Terms

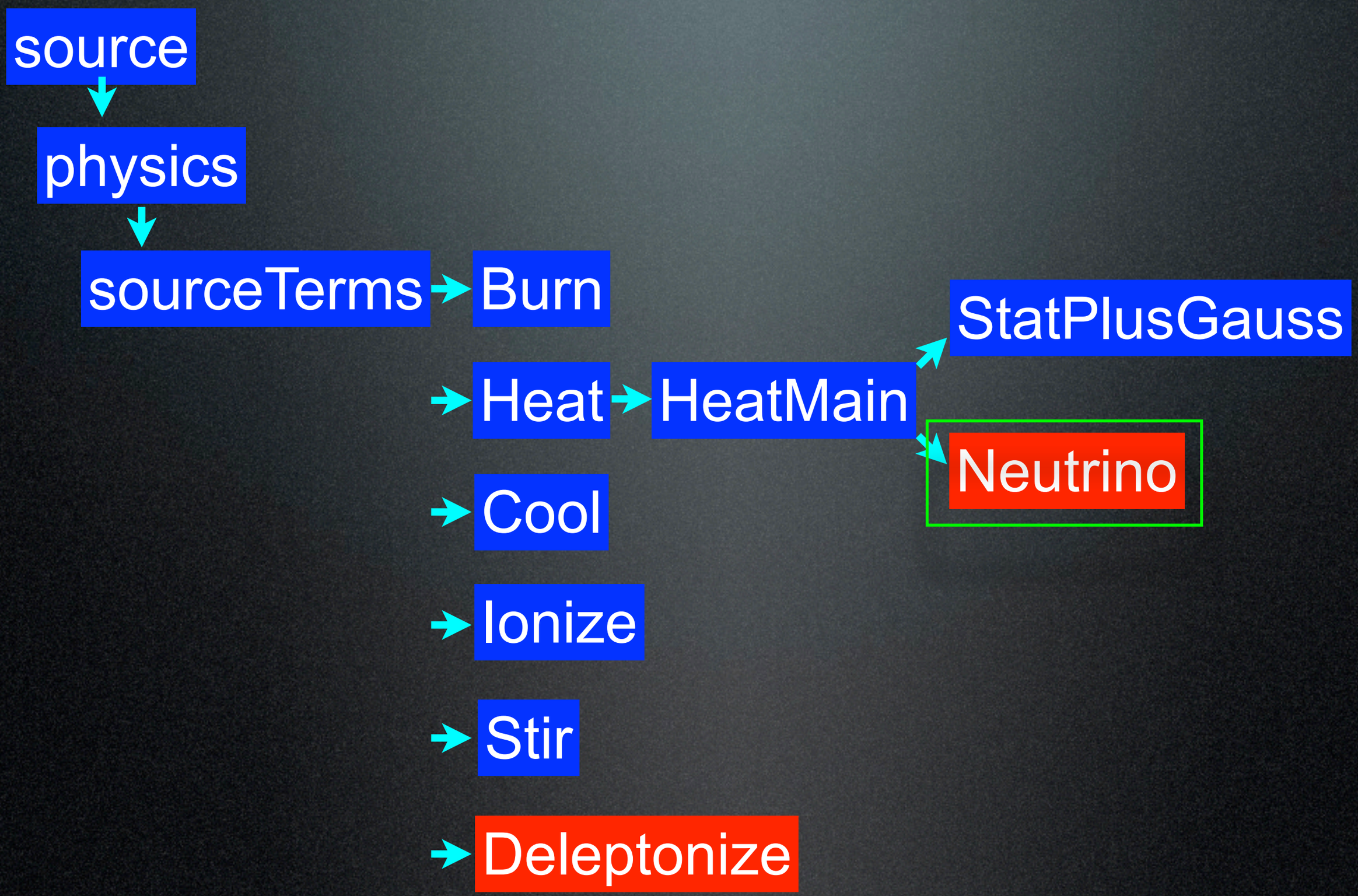


Neutrino Source Terms

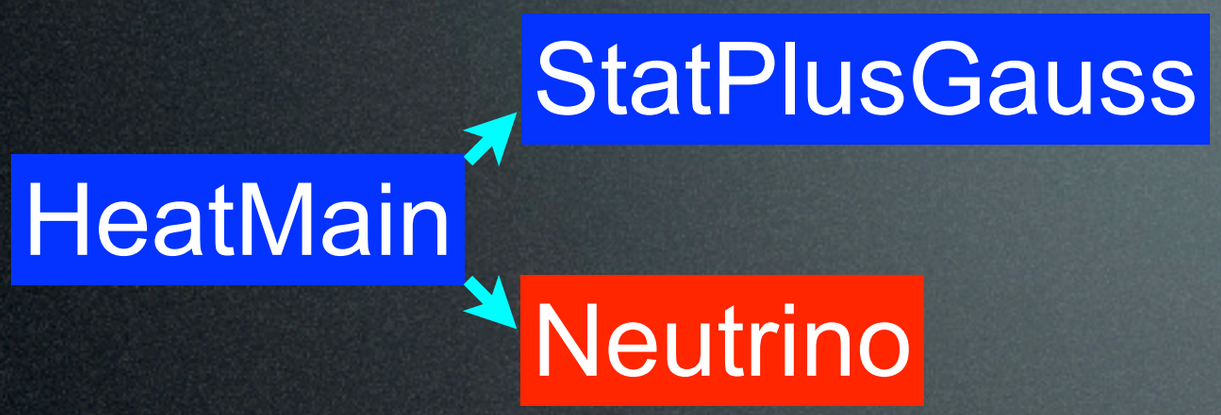


Neutrino Source Terms

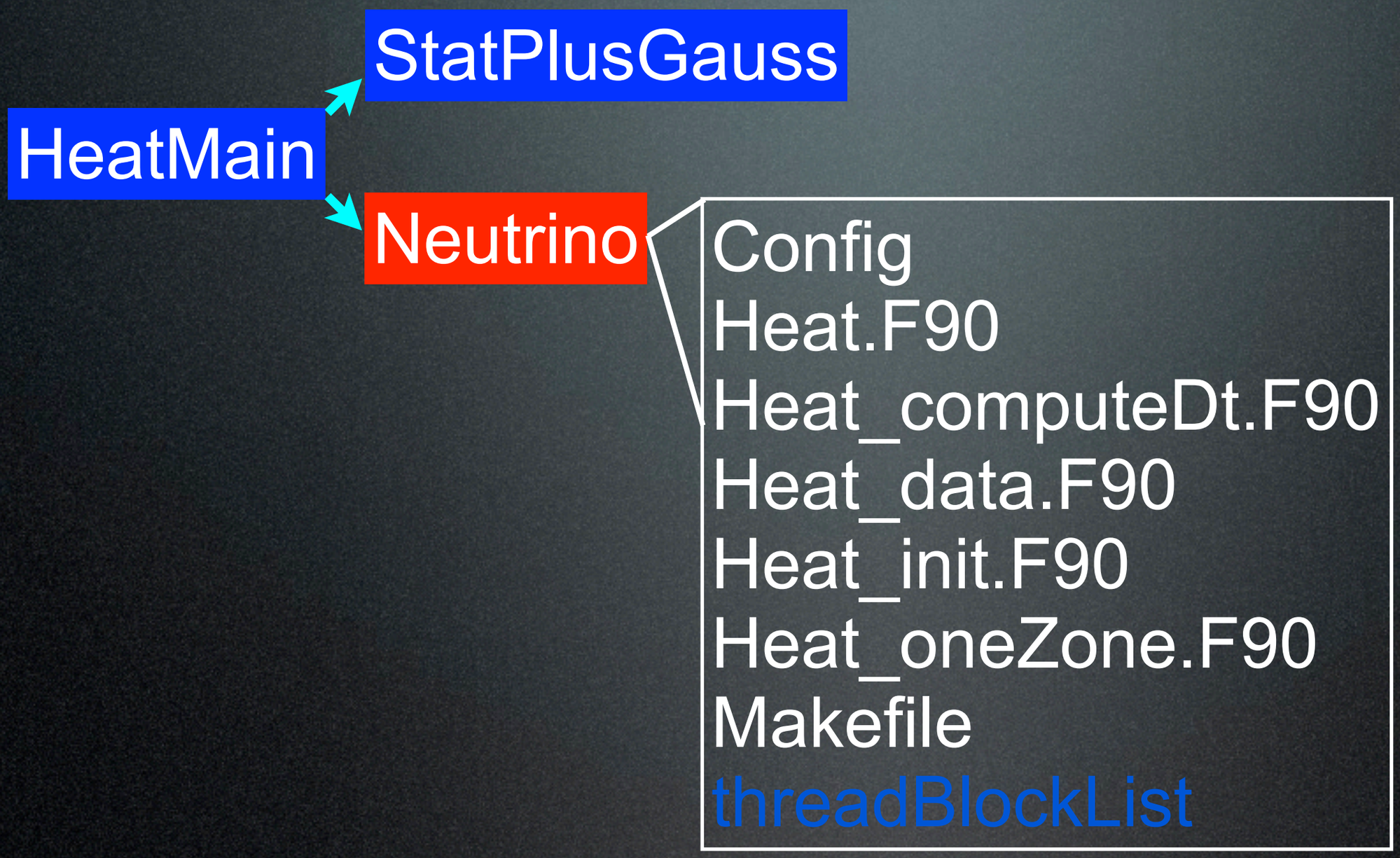
Neutrino Source Terms



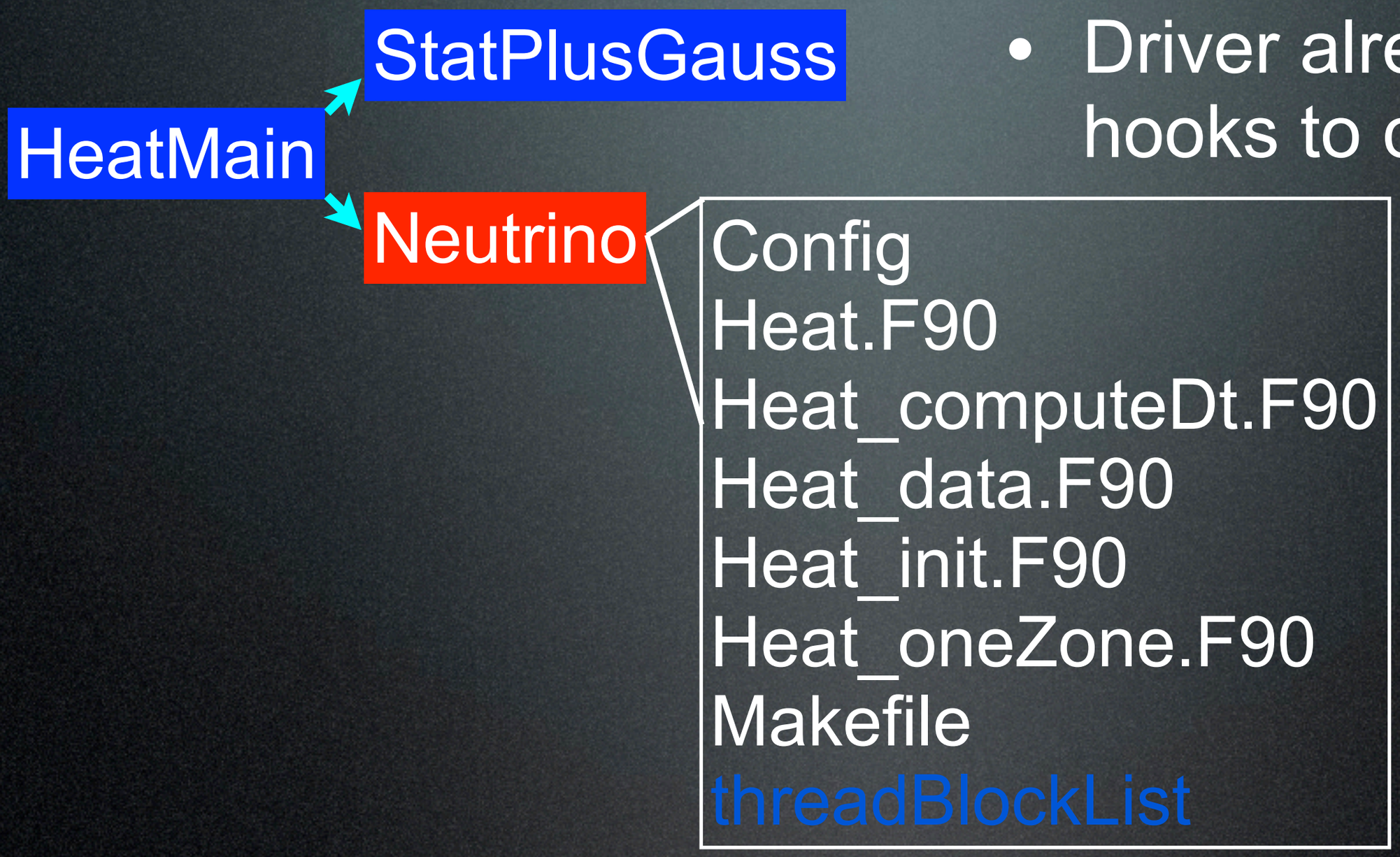
Neutrino Source Terms



Neutrino Source Terms

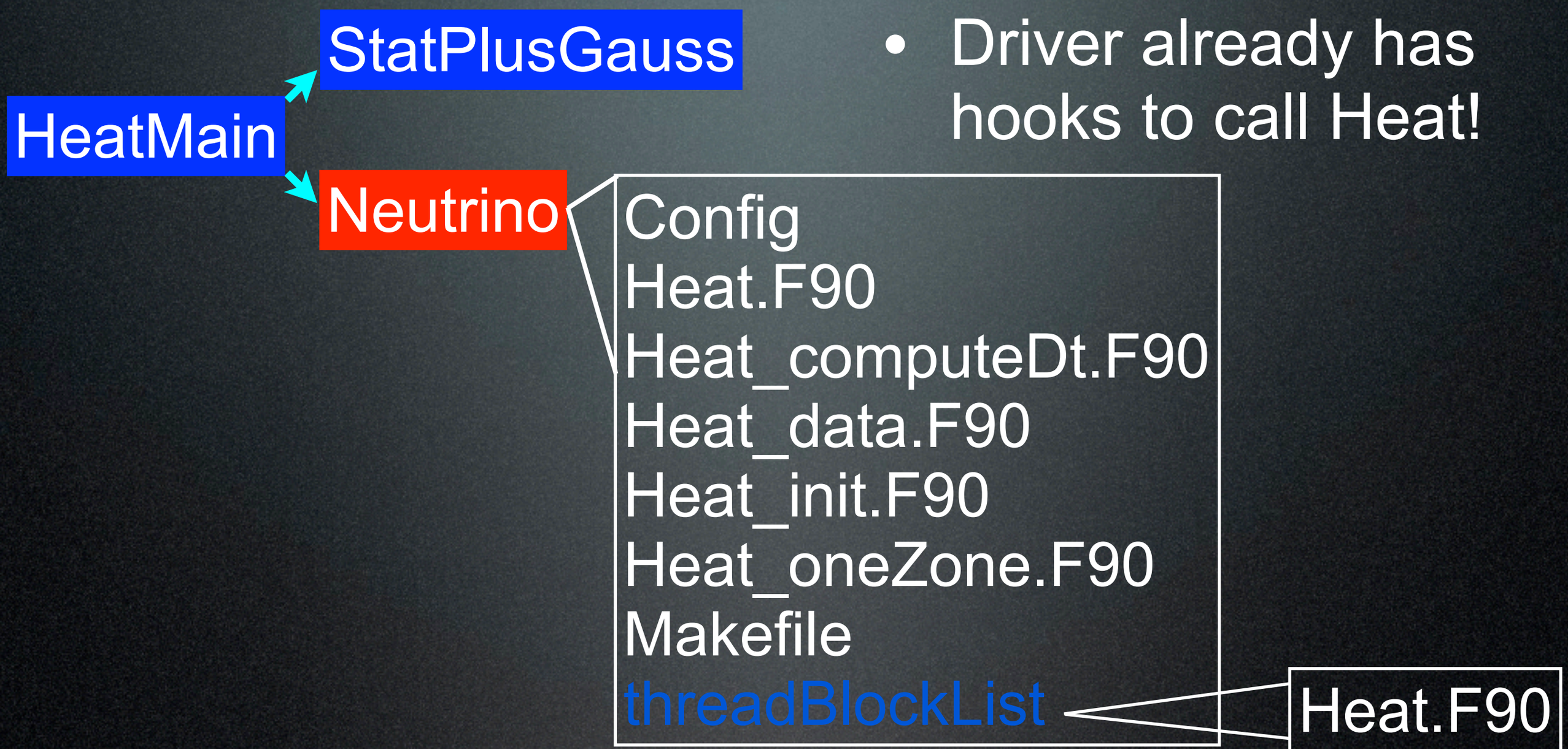


Neutrino Source Terms



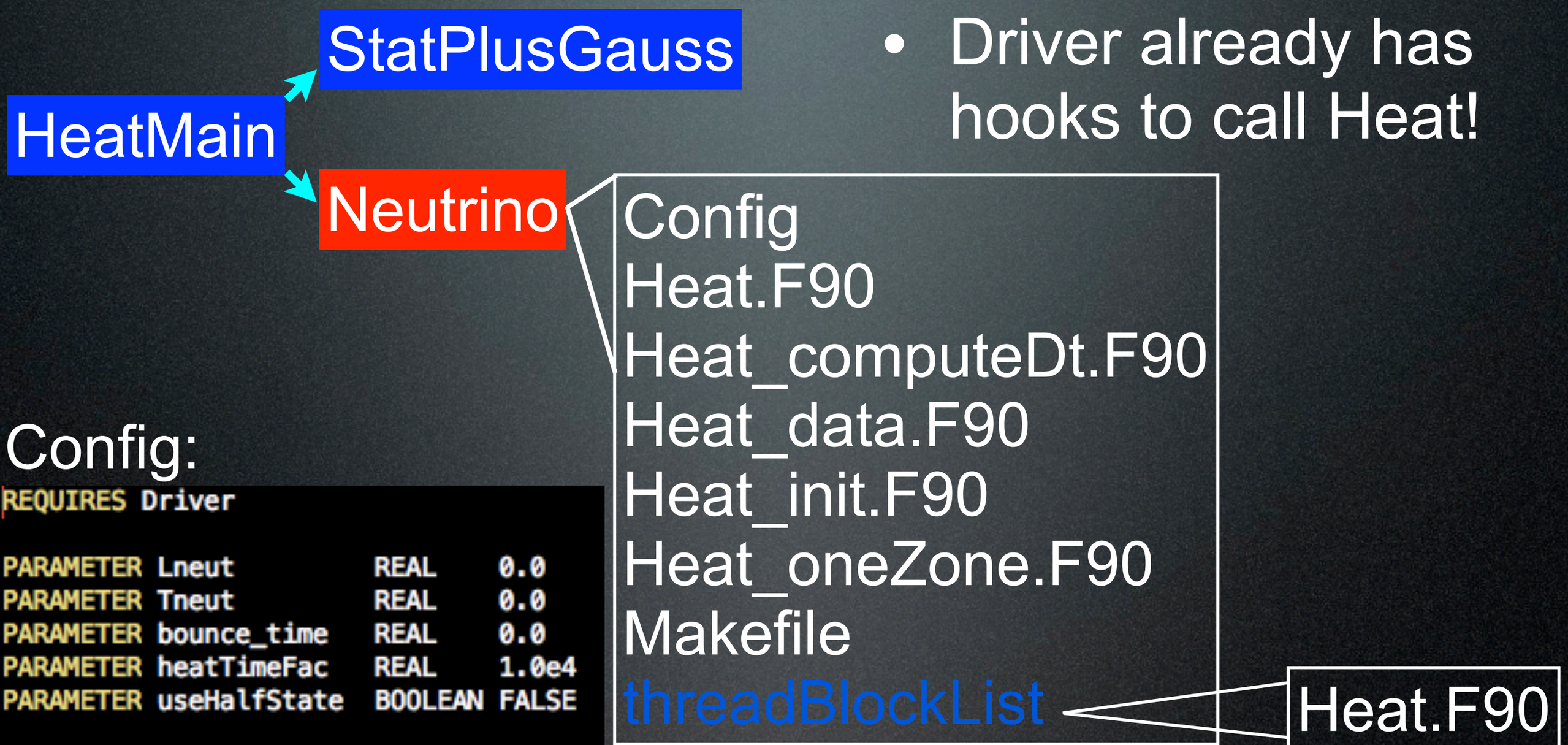
- Driver already has hooks to call Heat!

Neutrino Source Terms



- Driver already has hooks to call Heat!

Neutrino Source Terms



- Driver already has hooks to call Heat!

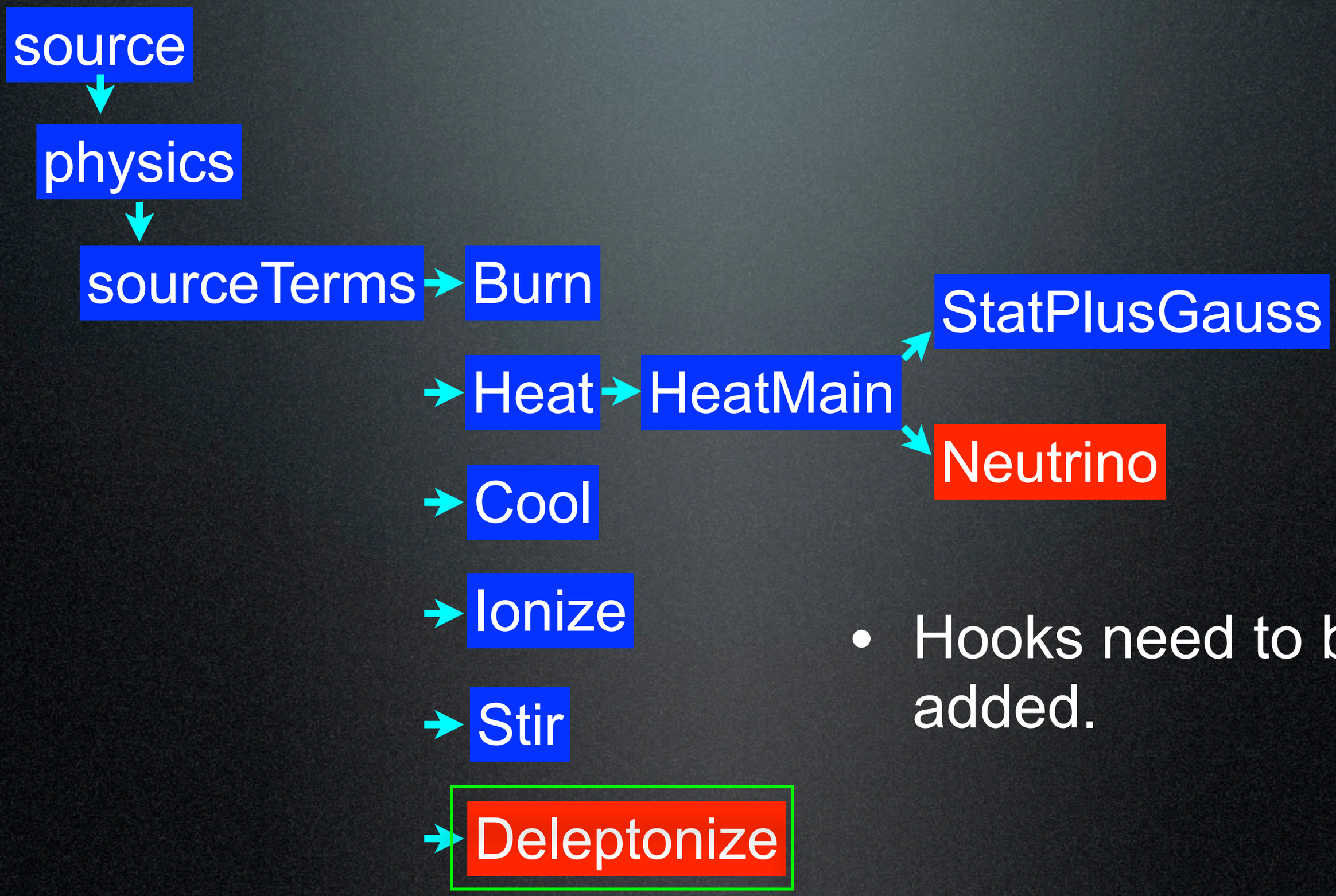
Config:

```
REQUIRES Driver

PARAMETER Lneut          REAL    0.0
PARAMETER Tneut          REAL    0.0
PARAMETER bounce_time    REAL    0.0
PARAMETER heatTimeFac    REAL    1.0e4
PARAMETER useHalfState   BOOLEAN FALSE

USESETUPVARS threadBlockList

IF threadBlockList
  PPDEFINE ST_THREAD_BLOCK_LIST
  REQUIRES ./threadBlockList
ENDIF
```

Neutrino Source Terms

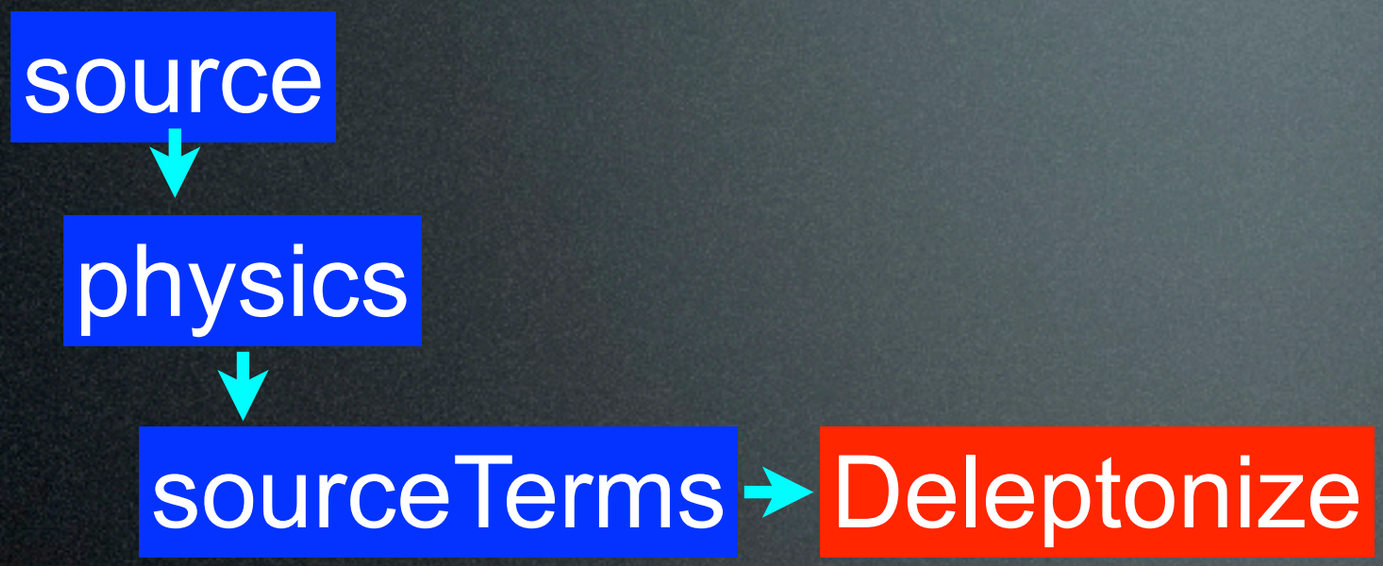
Neutrino Source Terms

Neutrino Source Terms

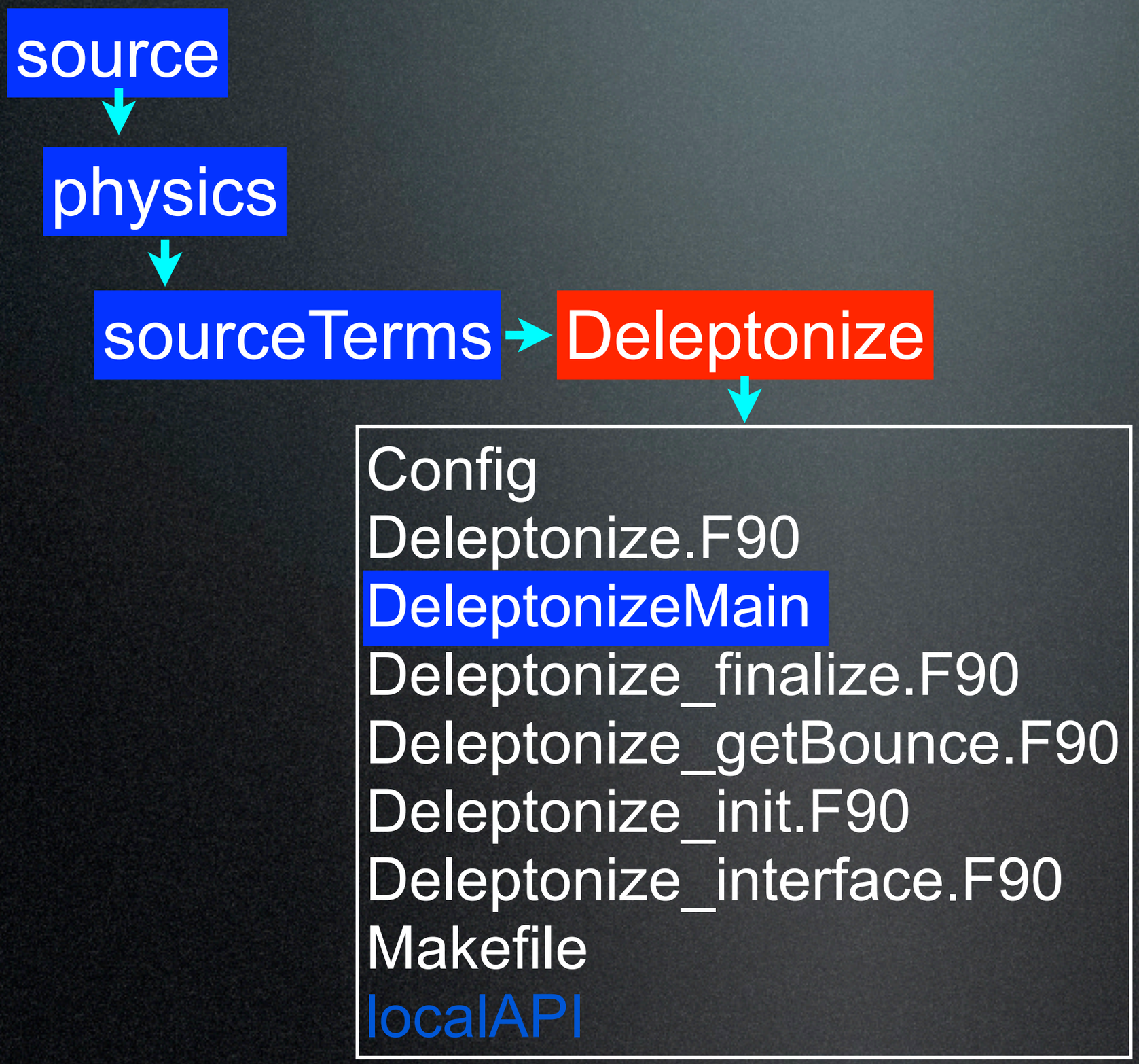


- Hooks need to be added.

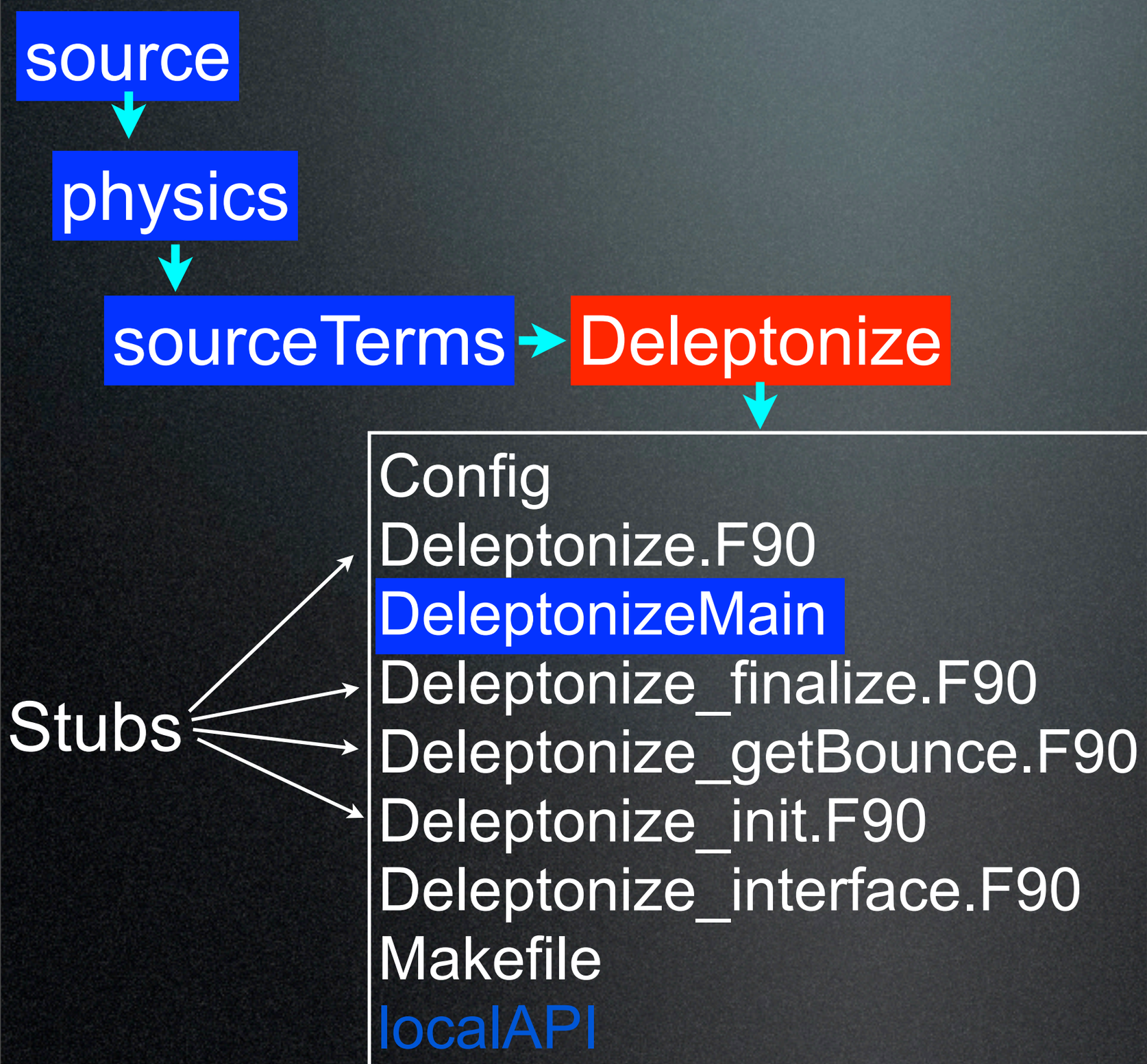
Neutrino Source Terms



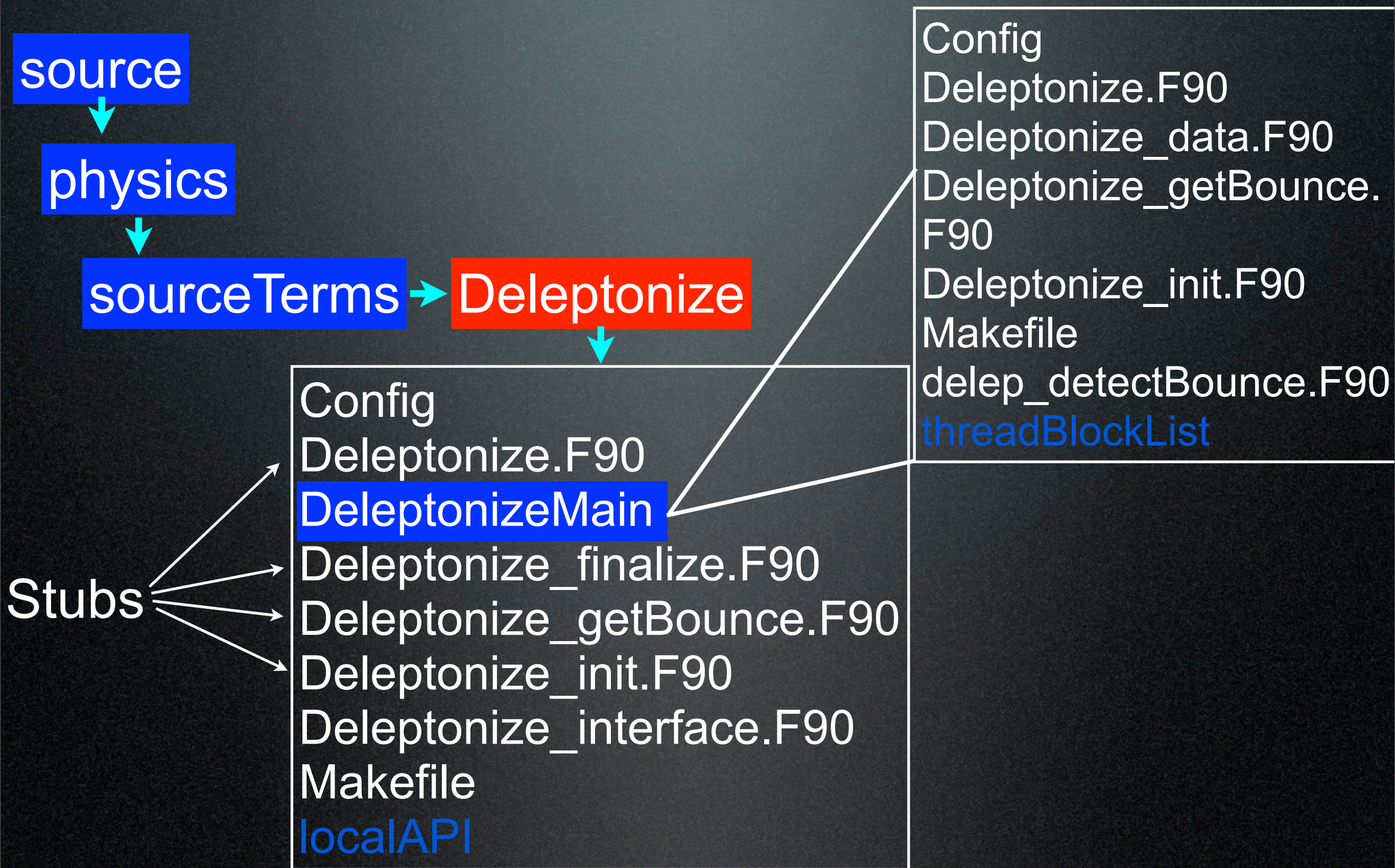
Neutrino Source Terms



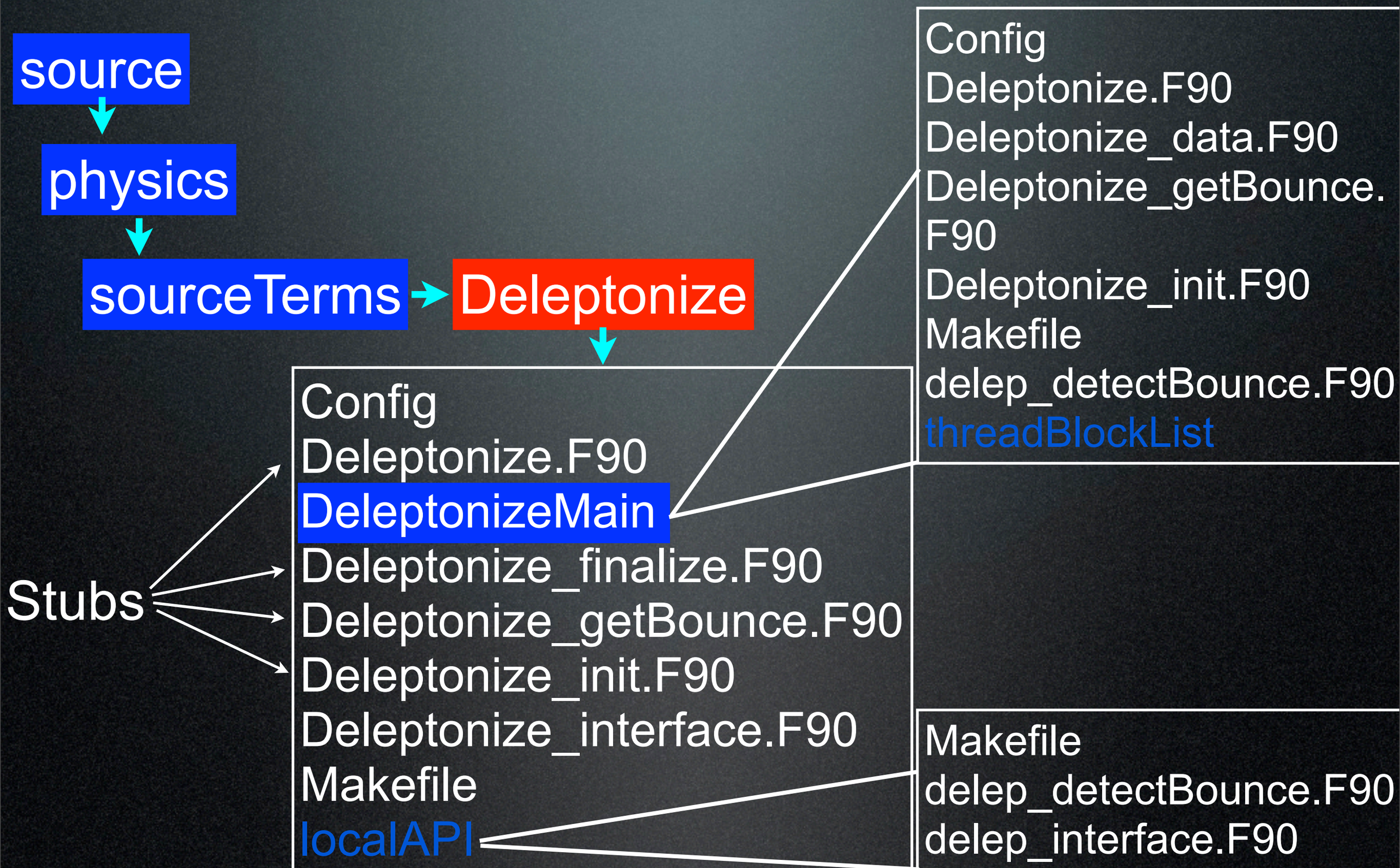
Neutrino Source Terms



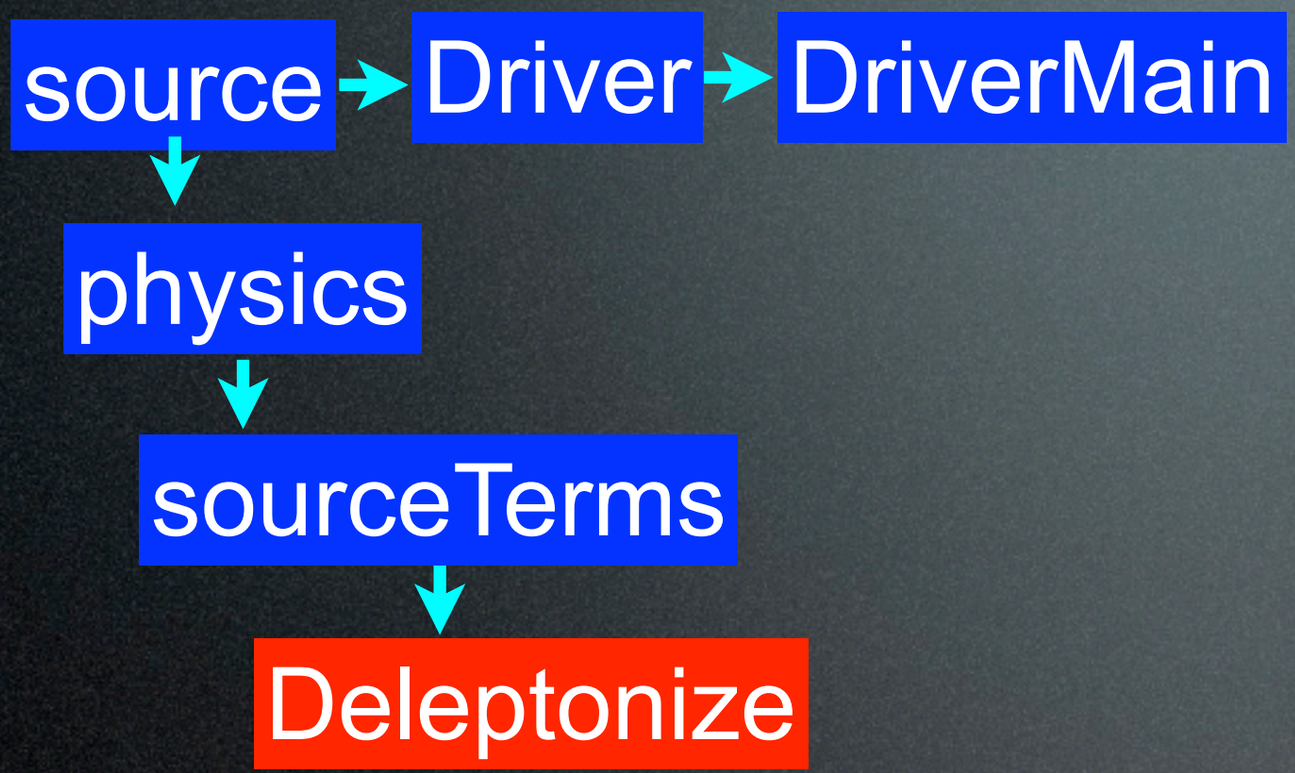
Neutrino Source Terms



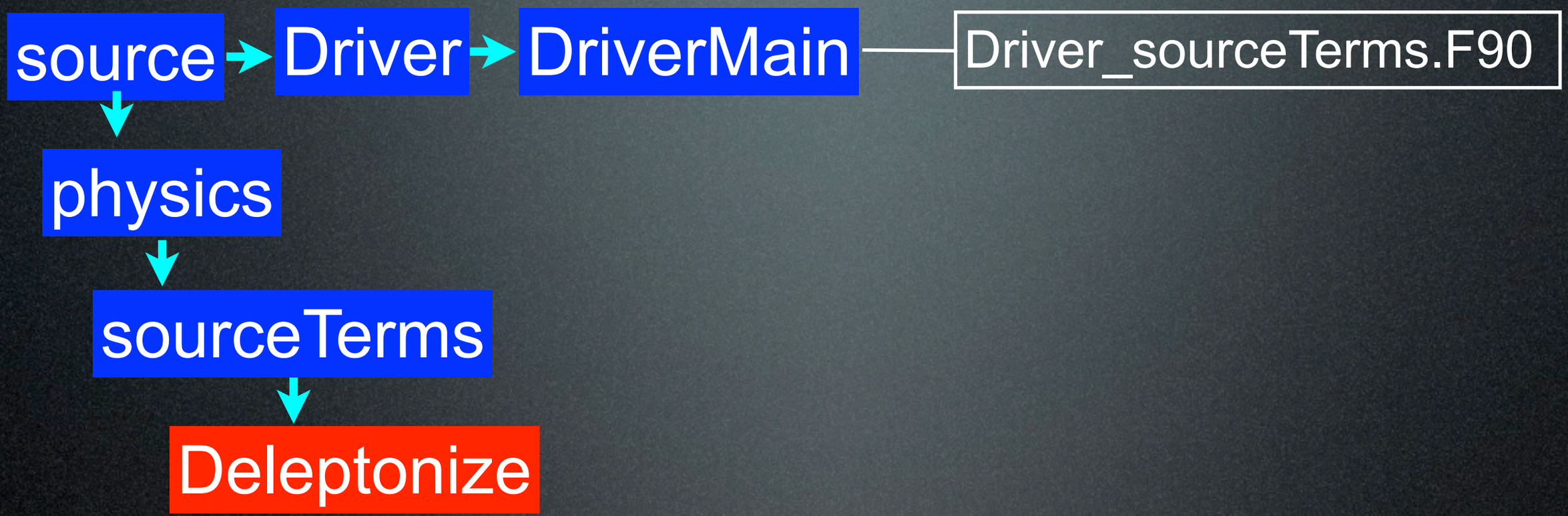
Neutrino Source Terms



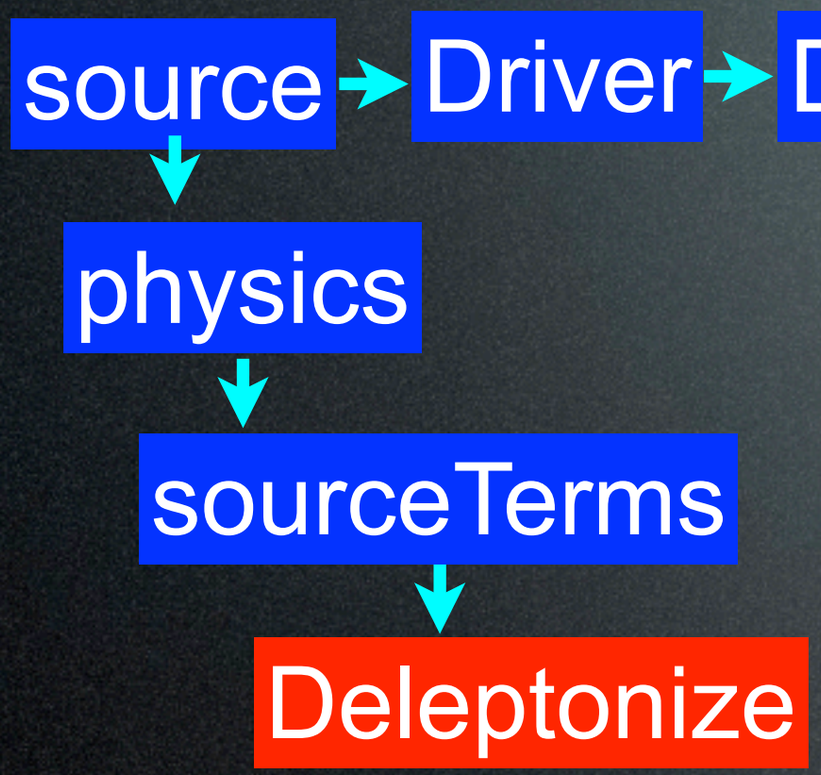
Neutrino Source Terms



Neutrino Source Terms



Neutrino Source Terms



Driver_sourceTerms.F90

```
use Heat_interface, ONLY : Heat
use Heatexchange_interface, ONLY : Heatexchange
use Burn_interface, ONLY : Burn
use Cool_interface, ONLY : Cool
use Ionize_interface, ONLY : Ionize
use EnergyDeposition_interface, ONLY : EnergyDeposition
use Deleptonize_interface, ONLY : Deleptonize

implicit none

real, intent(IN)    :: dt
integer, intent(IN) :: blockCount
integer, dimension(blockCount), intent(IN):: blockList
integer, OPTIONAL, intent(IN):: pass

call Polytrope(blockCount, blockList, dt)
call Stir(blockCount, blockList, dt)
call Flame_step(blockCount, blockList, dt)
call Burn(blockCount, blockList, dt)
call Heat(blockCount, blockList, dt, dr_simTime)
call Heatexchange(blockCount, blockList, dt)
call Cool(blockCount, blockList, dt, dr_simTime)
call Ionize(blockCount, blockList, dt, dr_simTime)
call EnergyDeposition(blockCount, blockList, dt, dr_simTime, pass)
call Deleptonize(blockCount, blockList, dt, dr_simTime)
```

Core-Collapse Requirements

- yes • Higher-order shock-capture hydro/MHD
- yes • Self-gravity
- no • Realistic EOS
- no • Electron capture physics during collapse
- no • Neutrino heating/cooling
- no • Neutrino leakage
- yes • Multiple concurrent setups w/o code duplication

Core-Collapse Requirements

- yes • Higher-order shock-capture hydro/MHD
- yes • Self-gravity
- no • Realistic EOS
- no • Electron capture physics during collapse
- no • Neutrino heating/cooling
- no • Neutrino leakage
- yes • Multiple concurrent setups w/o code duplication

CCSN Equation of State

- Contributions to thermodynamics from baryons, photons, degenerate/relativistic elections & positrons
- Baryons experience phase change at high-density ($>10^{12}$ g cm^{-3}) when strong nuclear force becomes repulsive
- Traditionally handled via table lookup

CCSN Equation of State

- Several nuclear force models; need ability to swap different tables at runtime
- Kernel implementations from collaborators; available at stellarcollapse.org
- Need to interface FLASH EOS Unit with kernel EOS routines

CCSN Equation of State

source → physics → Eos → EosMain → Gamma

→ Multigamma

→ Helmholtz

→ Tabulated

→ Multitemp

CCSN Equation of State

source → physics → Eos → EosMain → Gamma

→ Multigamma

→ Helmholtz

→ Tabulated

→ Multitemp

→ Nuclear

CCSN Equation of State

source → physics → Eos → EosMain → Gamma

→ Multigamma

→ Helmholtz

→ Tabulated

→ Multitemp

→ Nuclear

Config
Eos_nucOneZone.F90
Makefile
eos_initNuclear.F90
eos_lowdens.F90
eos_nuclear.F90
kernel

CCSN Equation of State

source → physics → Eos → EosMain → Gamma

→ Multigamma

→ Helmholtz

→ Tabulated

→ Multitemp

→ Nuclear

Config

Eos_nucOneZone.F90
Makefile
eos_initNuclear.F90
eos_lowdens.F90
eos_nuclear.F90
kernel

CCSN Equation of State

source → physics → Eos → EosMain → Gamma

→ Multigamma

→ Helmholtz

→ Tabulated

→ Multitemp

→ Nuclear

```
#       Config file for the nuclear equation of state
# [physics/Eos/EosMain/Nuclear]

DEFAULT kernel

VARIABLE gamc  EOSMAP:   GAMC  # sound-speed gamma
VARIABLE game  EOSMAP:   GAME  # energy gamma
VARIABLE entr  EOSMAP:  ENTR  # specific entropy in kB per baryon


PARAMETER eos_file  STRING "eosTable"
PARAMETER eos_table_tmax  REAL 250.
```

CCSN Equation of State

source → physics → Eos → EosMain

Eos.F90

```
select case(eos_type)
case(EOS_GAM)
   call eos_idealGamma(mode, vecLen, eosData, vecBegin,vecEnd, massFrac=massFrac
case(EOS_MGAM)
   call eos_mgamma(mode, vecLen, eosData, vecBegin,vecEnd, massFrac=massFrac, ma
case(EOS_HLM)
   call eos_helmholtz(mode, vecLen, eosData, massFrac=massFrac, mask=mask)
case(EOS_TAB)
   call eos_tabulated(mode, vecLen, eosData, massFrac=massFrac, mask=mask)
→ case(EOS_NUC)
   call eos_nuclear(mode, vecLen, eosData, massFrac, mask=mask)
case default
   if (eos_meshMe==MASTER_PE) print*,'[Eos] unrecognized eos_type',eos_type
   call Driver_abortFlash('[Eos] unrecognized eos_type.')
end select
```

Eos_init.F90

```
call eos_initGamma()
call eos_initMgamma()
call eos_initHelmholtz()
call eos_initMtemp()
call eos_initTabulated()
→ call eos_initNuclear()
```

CCSN Equation of State

source → physics → Eos → EosMain → Gamma

→ Multigamma

→ Helmholtz

→ Tabulated

→ Multitemp

→ Nuclear

Config
Eos_nucOneZone.F90
Makefile
eos_initNuclear.F90
eos_lowdens.F90
eos_nuclear.F90
kernel

CCSN Equation of State

source → physics → Eos → EosMain → Gamma

→ Multigamma

→ Helmholtz

→ Tabulated

→ Multitemp

→ Nuclear

Config

Eos_nucOneZone.F90

Makefile

eos_initNuclear.F90

eos_lowdens.F90

eos_nuclear.F90

kernel

Core-Collapse Requirements

- yes • Higher-order shock-capture hydro/MHD
- yes • Self-gravity
- no • Realistic EOS
- no • Electron capture physics during collapse
- no • Neutrino heating/cooling
- no • Neutrino leakage
- yes • Multiple concurrent setups w/o code duplication

Core-Collapse Requirements

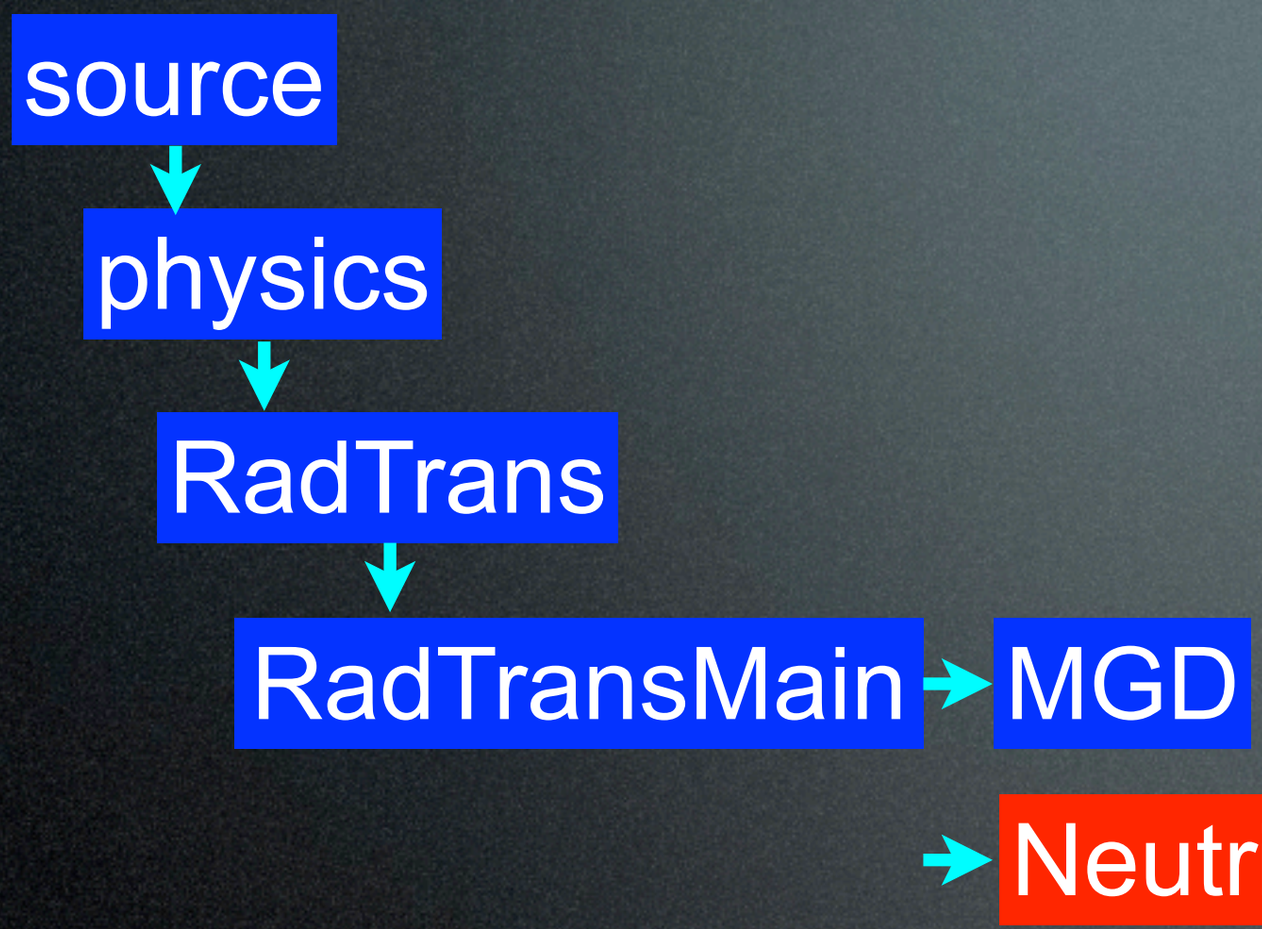
- yes • Higher-order shock-capture hydro/MHD
- yes • Self-gravity
- no • Realistic EOS
- no • Electron capture physics during collapse
- no • Neutrino heating/cooling
- no • Neutrino leakage
- yes • Multiple concurrent setups w/o code duplication

Neutrino Leakage

- Compute approximate transport along radial rays embedded in Cartesian domain
- Memory- and compute-intensive

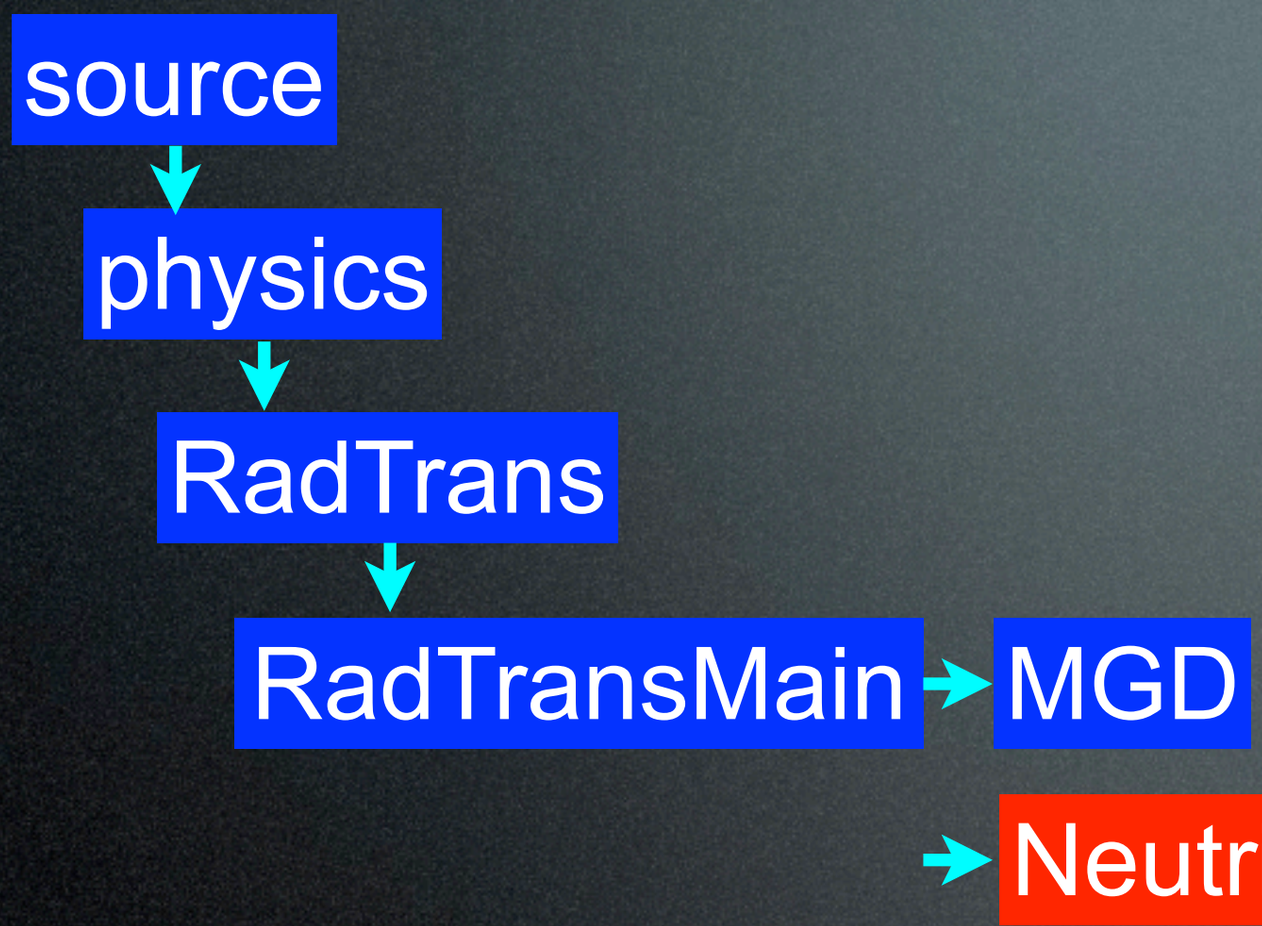
Neutrino Leakage

Neutrino Leakage



Config
Makefile
RadTrans.F90
RadTrans_computeDt.F90
RadTrans_finalize.F90
rt_calcLeak.F90
rt_calcTau.F90
rt_data.F90
rt_init.F90
rt_remapRays.F90
rt_sampleRays.F90
threadBlockList

Neutrino Leakage



- Driver already has hooks to call RadTrans!
- Private routines only code within unit

Config	rt_calcTau.F90
Makefile	rt_data.F90
RadTrans.F90	rt_init.F90
RadTrans_computeDt.F90	rt_remapRays.F90
RadTrans_finalize.F90	rt_sampleRays.F90
rt_calcLeak.F90	threadBlockList

Neutrino Leakage

rt_init.F90:

FLASH architecture allows easy instantiation of alternate communication patterns

```
 call MPI_COMM_SIZE(rt_meshComm, rt_meshNumProcs,error)

 if (mod(rt_meshNumProcs,rt_subCommSize) /= 0 .OR. rt_subCommSize == -1) then
    if (rt_globalMe == MASTER_PE) write(*,*) "RadTrans: setting leak_subCommSize to
meshNumProcs"
    rt_subCommSize = rt_meshNumProcs
!       call Driver_abortFlash("leak_subCommSize must evenly divide meshNumProcs")
 end if

 rt_subMeshMe =  mod(rt_globalMe, rt_subCommSize)
 call MPI_COMM_SPLIT(rt_meshComm, rt_globalMe/rt_subCommSize, rt_subMeshMe, rt_subM
eshComm, error)

   ! Now let's parse up the rays among different processors.
 allocate(rt_recvCnt(rt_subCommSize))
 allocate(rt_dsplCnt(rt_subCommSize))
 rt_recvCnt = 0
 rt_dsplCnt = 0

 numRaysPerProc = rt_leakNumRays / rt_subCommSize
 num = mod(rt_leakNumRays, rt_subCommSize)
 iend = 0
 do i=1,rt_subCommSize
    if (i <= num) then
```

Core-Collapse Requirements

- yes • Higher-order shock-capture hydro/MHD
- yes • Self-gravity
- no • Realistic EOS
- no • Electron capture physics during collapse
- no • Neutrino heating/cooling
- no • Neutrino leakage
- yes • Multiple concurrent setups w/o code duplication

Core-Collapse Requirements

- yes • Higher-order shock-capture hydro/MHD
- yes • Self-gravity
- yes • Realistic EOS
- yes • Electron capture physics during collapse
- yes • Neutrino heating/cooling
- yes • Neutrino leakage
- yes • Multiple concurrent setups w/o code duplication

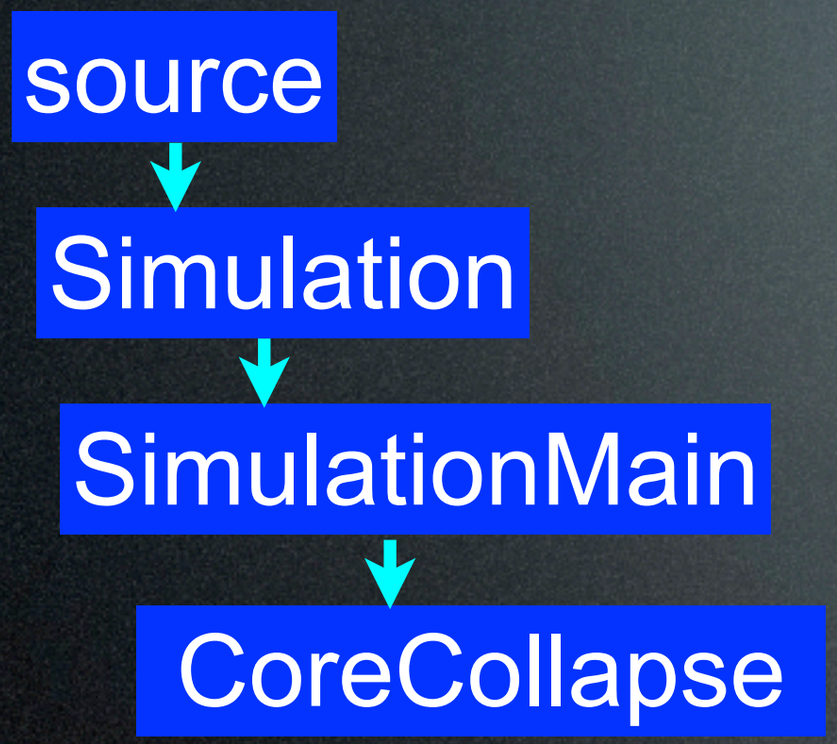
Core-Collapse Requirements

- yes • Higher-order shock-capture hydro/MHD
- yes • Self-gravity
- yes • Realistic EOS
- yes • Electron capture physics during collapse
- yes • Neutrino heating/cooling
- yes • Neutrino leakage
- yes • Multiple concurrent setups w/o code duplication

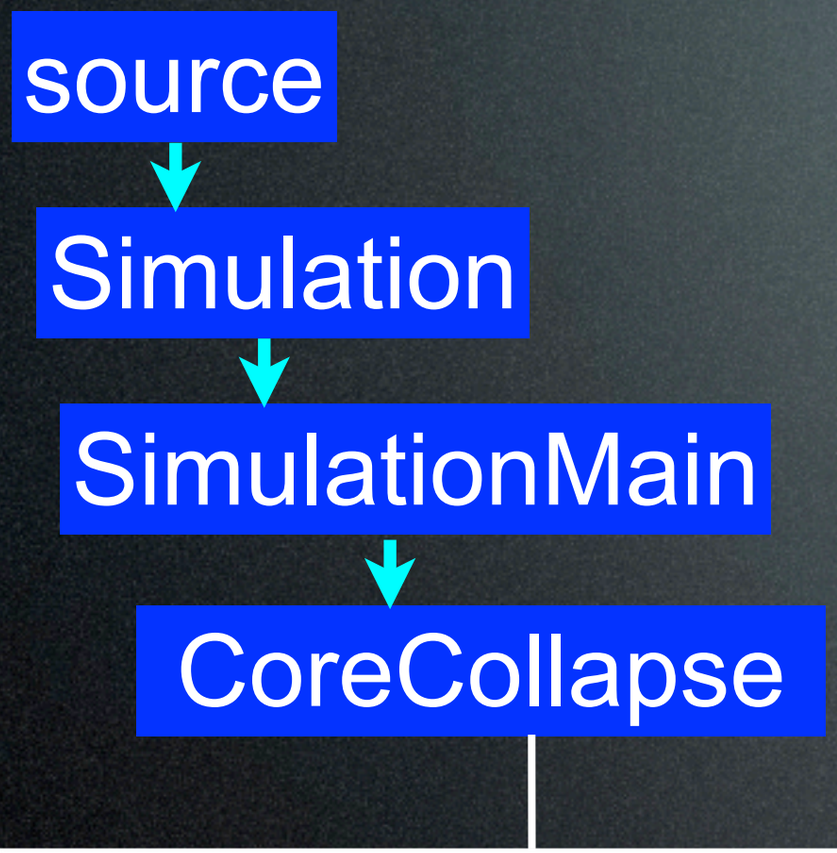
Including New Physics

- Unit rules and inheritance apply to Simulation Unit as well
- Coexistence of multiple application based on same setup feasible by multiple means

Including New Physics



Including New Physics



Config
Grid_bcApplyToRegionSpecialized.F90
Grid_markRefineDerefine.F90
Hydro_detectShock.F90
IO_writeIntegralQuantities.F90
MRITest
Makefile
Perturb
Simulation_data.F90
Simulation_init.F90
Simulation_initBlock.F90
eos_helm.F90
flash.par
Leakage
MagnetoHD

Including New Physics

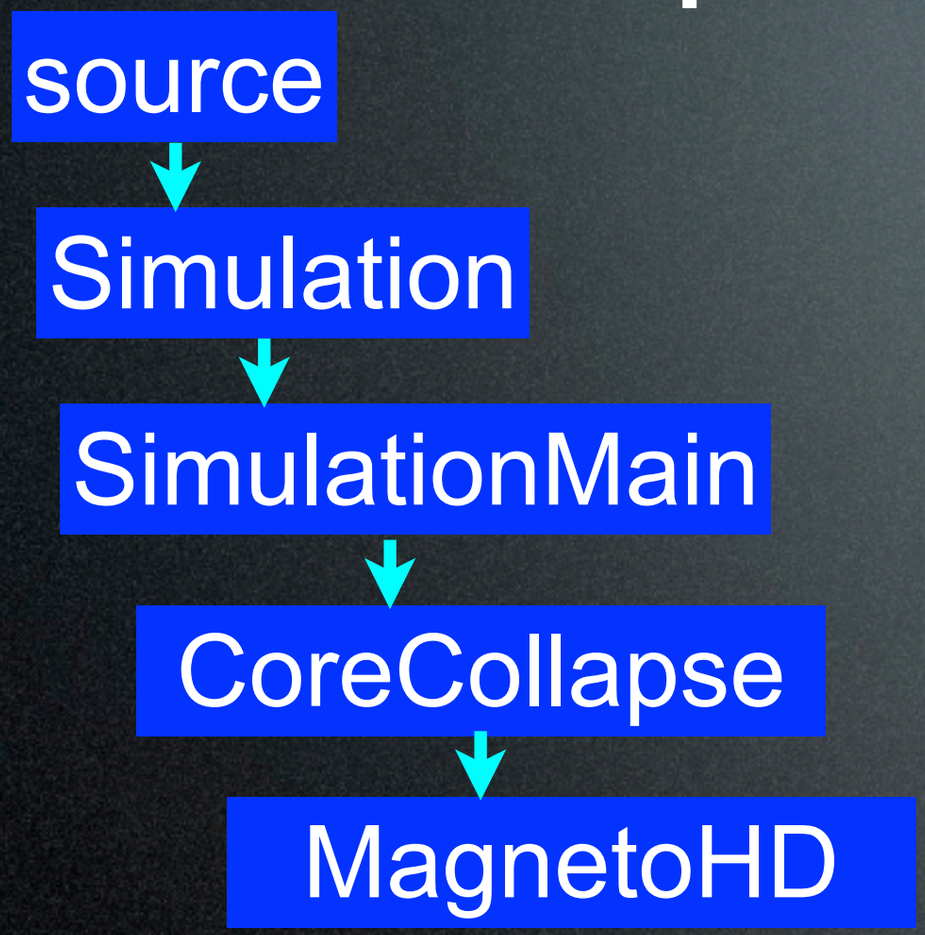
source
↓
Simulation
↓
SimulationMain
↓
CoreCollapse

Config

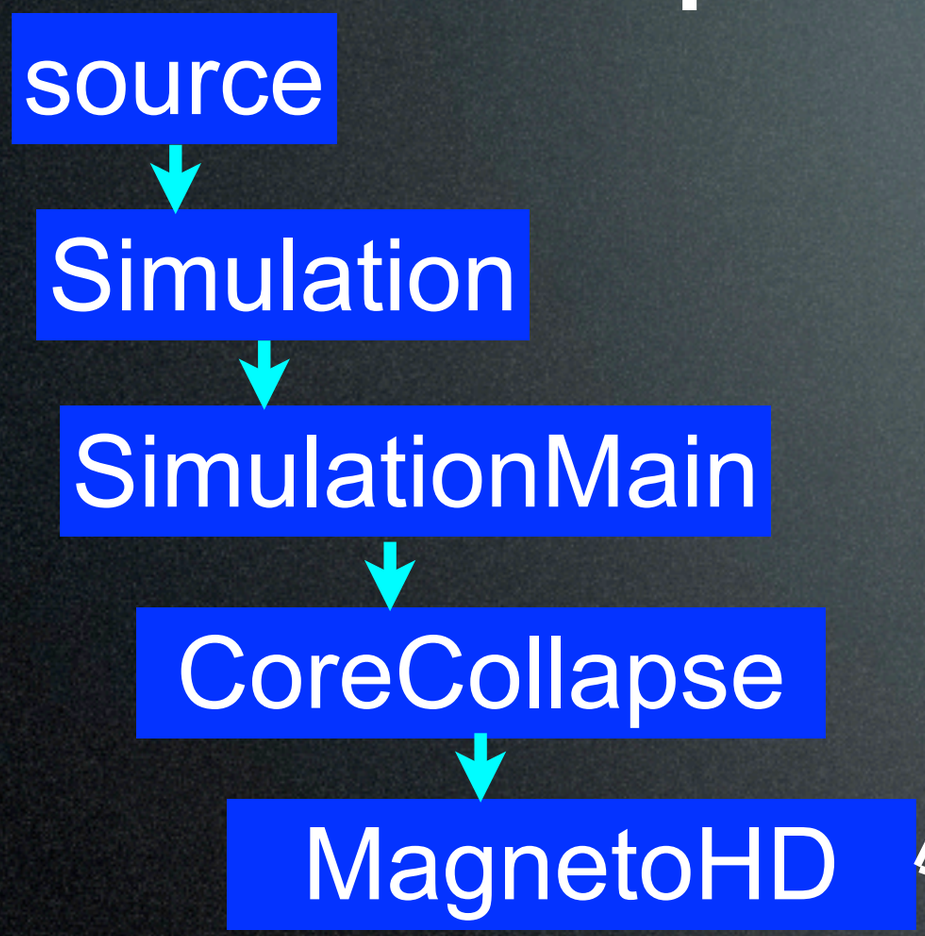
```
# Config file for Core Collapse SN setup.

REQUIRES Driver
REQUIRES Grid
REQUIRES physics/Hydro/HydroMain
REQUIRES physics/Eos/EosMain
REQUESTS physics/sourceTerms/Deleptonize
REQUESTS physics/Gravity/GravityMain/Poisson/Multipole
REQUESTS physics/sourceTerms/Heat/HeatMain/Neutrino


D nsub    number of sub-sampling points for mapping of 1D model
PARAMETER nsub          INTEGER 4
```

Config
Grid_bcApplyToRegionSpecialized.F90
Grid_markRefineDerefine.F90
Hydro_detectShock.F90
IO_writeIntegralQuantities.F90
MRITest
Makefile
Perturb
Simulation_data.F90
Simulation_init.F90
Simulation_initBlock.F90
eos_helm.F90
flash.par
Leakage
MagnetoHD

Multiple Simulation Setups



Multiple Simulation Setups



Config
Driver_evolveFlash.F90
Grid_markRefineDerefine.F90
IO_writeIntegralQuantities.F90
Simulation_customizeProlong.F90
Simulation_initBlock.F90
flash.par

Multiple Simulation Setups

source
↓
Simulation
↓
SimulationMain
↓
CoreCollapse
↓
MagnetoHD

Config
Driver_evolveFlash.F90
Grid_markRefineDerefine.F90
IO_writeIntegralQuantities.F90
Simulation_customizeProlong.F90
Simulation_initBlock.F90
flash.par

```
# Config file for MagnetoHD Core Collapse SN setup.
# This version uses leakage.

REQUIRES physics/RadTrans/RadTransMain/NeutrinoLeakage


#VARIABLE CMRI
VARIABLE LMRI
```

Multiple Simulation Setups

source
↓
Simulation
↓
SimulationMain
↓
CoreCollapse
↓
MagnetoHD

Config
Driver_evolveFlash.F90
Grid_markRefineDerefine.F90
IO_writeIntegralQuantities.F90
Simulation_customizeProlong.F90
Simulation_initBlock.F90
flash.par

```
# Config file for MagnetoHD Core Collapse SN setup.
# This version uses leakage.

REQUIRES physics/RadTrans/RadTransMain/NeutrinoLeakage


#VARIABLE CMRI
VARIABLE LMRI
```

```
#       Config file for Neutrino Leakage RadTrans
REQUIRES physics/sourceTerms/Deleptonize

CONFLICTS physics/sourceTerms/Heat/HeatMain/Neutrino

PARAMETER leak_radMax   REAL 0.0
```

Multiple Simulation Setups

Use of setup syntax

```
./setup CoreCollapse -auto -3d +cube16 +pm4dev +uhdopt
+newMpole threadWithinBlock=True
```

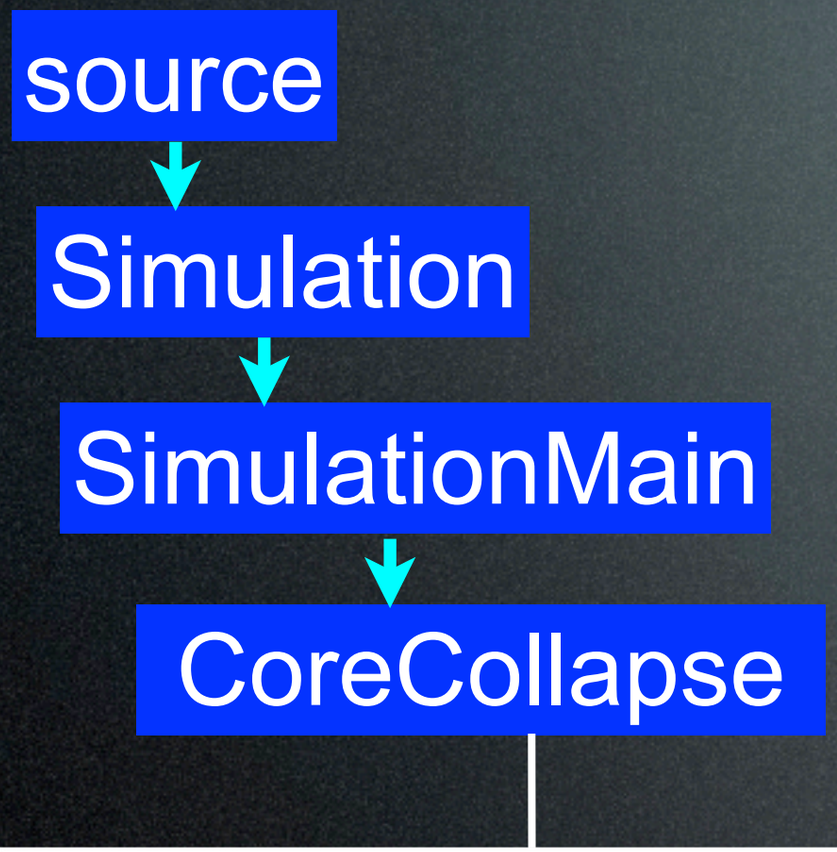
Multiple Simulation Setups

Use of setup syntax

```
./setup CoreCollapse -auto -3d +cube16 +pm4dev +uhdopt
+newMpole threadWithinBlock=True
```

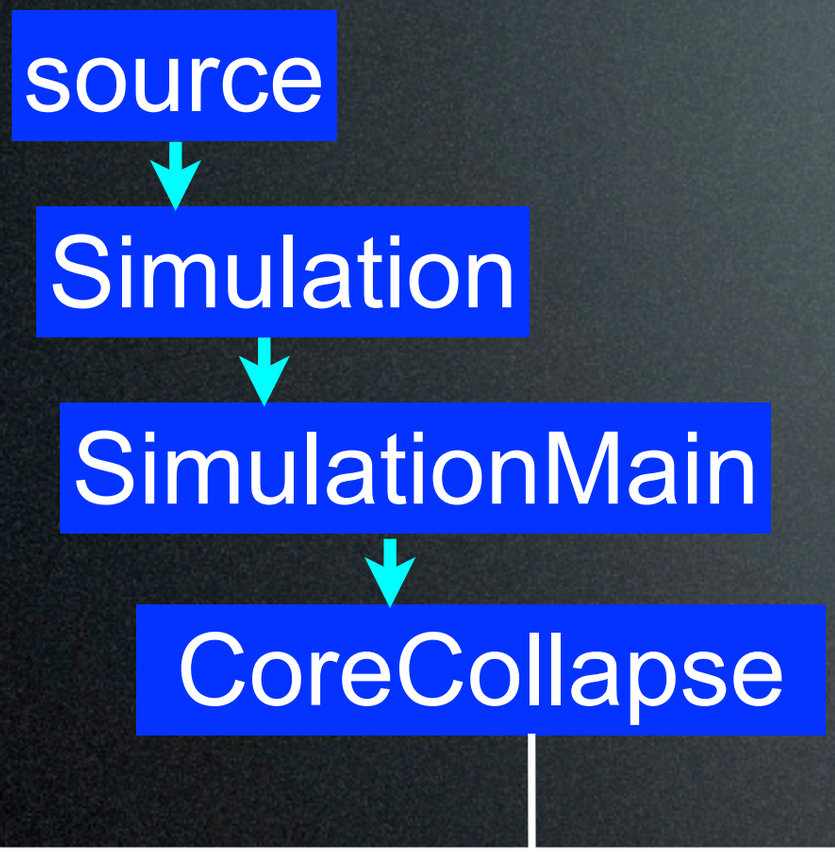
```
./setup CoreCollapse/MagnetoHD -auto -3d +cube16
+pm4dev +uhdopt +newMpole threadWithinBlock=True
--without-unit=physics/sourceTerms/Heat/HeatMain/Neutrino
```

Breaking the Law



Config
Grid_bcApplyToRegionSpecialized.F90
Grid_markRefineDerefine.F90
Hydro_detectShock.F90
IO_writeIntegralQuantities.F90
MRITest
Makefile
Perturb
Simulation_data.F90
Simulation_init.F90
Simulation_initBlock.F90
eos_helm.F90
flash.par
Leakage
MagnetoHD

Breaking the Law



Config
Grid_bcApplyToRegionSpecialized.F90
Grid_markRefineDerefine.F90
Hydro_detectShock.F90
IO_writeIntegralQuantities.F90
MRITest
Makefile
Perturb
Simulation_data.F90
Simulation_init.F90
Simulation_initBlock.F90
eos_helm.F90
flash.par
Leakage
MagnetoHD

Breaking the Law

source
↓
Simulation
↓
SimulationMain
↓
CoreCollapse

```
subroutine IO_writeIntegralQuantities ( isFirst, simTime)

#include "constants.h"
#include "Flash.h"
#include "UHD.h"

  use Simulation_data, ONLY: sim_pointMass, sim_expEner, &
                             sim_shockRadTot, sim_shockRadNum, &
                             sim_massAccRate, sim_massAccNum, &
                             sim_postBounce, sim_bounceTime, sim_maxDens
!!  use EnergyDeposition_data, ONLY : useEnergyDeposition
  use IO_data, ONLY : io_restart, io_statsFileName, io_globalComm
  use Grid_interface, ONLY : Grid_getListOfBlocks, &
    Grid_getBlkIndexLimits, Grid_getBlkPtr, Grid_getSingleCellVol, &
    Grid_releaseBlkPtr, Grid_getCellCoords

  use IO_data, ONLY : io_globalMe
  use Deleptonize_data, ONLY : delep_centralDens, delep_anteSonic, delep_centralEntr,
                               delep_postBounce, delep_bounceTime
  use Deleptonize_interface, ONLY : Deleptonize_getBounce
#ifdef FLASH_HYDRO_UNSPLIT
  use Hydro_data, ONLY : hy_cfl, hy_cfl_original, hy_RiemannSolver
#endif
```

Config
Grid_bcApplyToRegionS
d.F90
Grid_markRefineDerefine.F90
Hydro_detectShock.F90
IO_writeIntegralQuantities.F90
MRITest
Makefile

Simulation_initBlock.F90
eos_helm.F90
flash.par
Leakage
MagnetoHD

The FLASH object Directory

- Collects symbolic links to all the source code that will actually be compiled in your application
- Contains helpful information files about your application (setup_*)
- Highly ‘grep-able’

The FLASH object Directory

```
IncompNS_finalize.F90 -> ../source/physics/IncompNS/IncompNS_finalize.F90
IncompNS_computeDt.F90 -> ../source/physics/IncompNS/IncompNS_computeDt.F90
IncompNS.h -> ../source/physics/IncompNS/IncompNS.h
IncompNS.F90 -> ../source/physics/IncompNS/IncompNS.F90
ImBound_data.F90 -> ../source/physics/ImBound/ImBound_data.F90
IO_writeUserArray.F90 -> ../source/IO/IO_writeUserArray.F90
IO_writeRays.F90 -> ../source/IO/IOMain/hdf5/IO_writeRays.F90
IO_writeParticles.F90 -> ../source/IO/IO_writeParticles.F90
IO_writeIntegralQuantities.F90 -> ../source/Simulation/SimulationMain/CoreCollapse/IO_writeIntegralQuantities.F90
IO_writeCheckpoint.F90 -> ../source/IO/IOMain/IO_writeCheckpoint.F90
IO_updateScalars.F90 -> ../source/IO/IOMain/IO_updateScalars.F90
IO_startRayWrite.F90 -> ../source/IO/IOMain/hdf5/IO_startRayWrite.F90
IO_readUserArray.F90 -> ../source/IO/IO_readUserArray.F90
IO_readParticles.F90 -> ../source/IO/IO_readParticles.F90
IO_readCheckpoint.F90 -> ../source/IO/IOMain/IO_readCheckpoint.F90
IO_outputInitial.F90 -> ../source/IO/IOMain/IO_outputInitial.F90
IO_outputFinal.F90 -> ../source/IO/IOMain/IO_outputFinal.F90
IO_output.F90 -> ../source/IO/IOMain/IO_output.F90
IO_interface.F90 -> ../source/IO/IO_interface.F90
IO_initRPsFromCheckpoint.F90 -> ../source/IO/IOMain/IO_initRPsFromCheckpoint.F90
IO_getPrevScalar.F90 -> ../source/IO/IOMain/IO_getPrevScalar.F90
IO_finalize.F90 -> ../source/IO/IOMain/IO_finalize.F90
IO_data.F90 -> ../source/IO/IOMain/IO_data.F90
Hydro_sendOutputData.F90 -> ../source/physics/Hydro/Hydro_sendOutputData.F90
Hydro_mapBcType.F90 -> ../source/physics/Hydro/HydroMain/Hydro_mapBcType.F90
Hydro_init.F90 -> ../source/physics/Hydro/HydroMain/unsplit_opt/Hydro_Unsplit/Hydro_init.F90
Hydro_finalize.F90 -> ../source/physics/Hydro/HydroMain/unsplit_opt/Hydro_Unsplit/Hydro_finalize.F90
Heatexchange_finalize.F90 -> ../source/physics/sourceTerms/Heatexchange/Heatexchange_finalize.F90
Heatexchange_computeDt.F90 -> ../source/physics/sourceTerms/Heatexchange/Heatexchange_computeDt.F90
Heatexchange.F90 -> ../source/physics/sourceTerms/Heatexchange/Heatexchange.F90
Heat_interface.F90 -> ../source/physics/sourceTerms/Heat/Heat_interface.F90
Heat_finalize.F90 -> ../source/physics/sourceTerms/Heat/Heat_finalize.F90
Grid_updateSolidBodyForces.F90 -> ../source/Grid/Grid_updateSolidBodyForces.F90
```

Alternate Implementations

- Alternate implementations of units and sub-units in FLASH allows flexibility and experimentation
- Can act like pseudo-branching...
- Use of inheritance to reduce code duplication
- But...

Alternate Implementations

- Can lead to code duplication, difficult maintainability, and long lag times in reincorporation
- Examples: optimized unsplit hydro/MHD, threading implementations

Thanks!