MOFIA Tutorial

- MOFIA Manual
- FORTRAN 90
  - Advantages
  - Structures
  - Modules
- MOFIA Source Code Organization
- Compiling MOFIA
- Running MOFIA
- MOFIA Commands
- Namelists
  - Definition and coding
  - Description
  - Contents and assignments
- MOFIA Functions
- Command Files
- Calibration File Manager (CFM)
  - Defined types
  - Runs and Sets
  - Over-riding CFM
- Example of a Command File
- Histogramming
  - MOFIA Initialization Branch
  - MOFIA Histogramming Branch
- Histogramming Example
- Using the Data Structures Figures

MOFIA 2.2 Manual

X. Appendecies

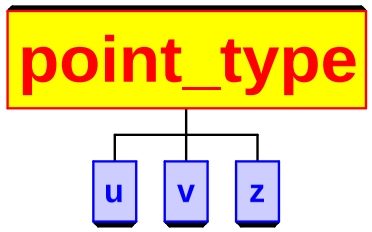
FORTRAN 90 Advantages

- Derived types (structures)
  - containing standard types
  - containing derived types
- Sharing data through modules
  - Variable declarations
  - Module subroutines
- Controlling shared variables
  - PUBLIC & PRIVATE statements
  - USE only statement
- Compact array operations
  - Array assignments with no DO loops
  - Array calculations with no DO loops

- Many other useful & powerful features:
  - Pointers
  - Allocatable arrays
  - Intent attributes
  - Recursion
  - Operator overloading
  - Free source form
  - New control structures (CASE, DO WHILE
  - EXIT & CYCLE statements
  - Interface blocks
  - Declarations may contain attributes and initializations.

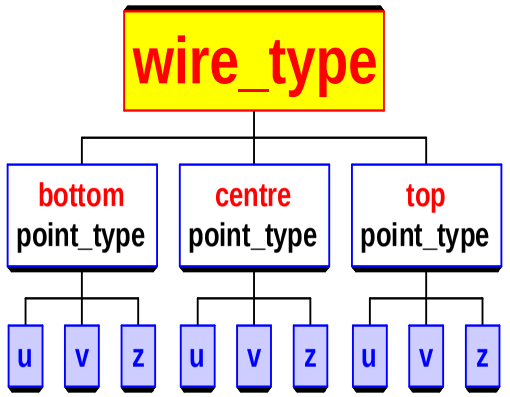
FORTRAN 90 Structures

- Define a structure for a point (u,v,z)




```
TYPE,PUBLIC::point_type
    REAL:: u, v, z
END TYPE point_type

REAL:: u1
REAL, POINTER:: v2P
TYPE(point_type)::  point1
TYPE(point_type)::  point2
```

- Define a structure wire locations




```
TYPE,PUBLIC::wire_type
        TYPE(point_type):: top, center, bottom
END TYPE wire_type

REAL:: WireU
TYPE(wire_type), DIMENSION(44,80)::
    WirePosition
```

- Assign the u-component of point1 to u

```
u1 = point1%u
v2P => point2%v
```

- Assign the u-component of the center of wire 5 in plane 23 to Wire U

```
WireU = WirePosition(5,23)%center%u
```

Modules

- A module is a programming unit
  - Contains derived types (structures) definitions
  - Variable declarations
  - Procedures (functions and subroutines)
- Why use modules?
  - Convenient way to package and or organize data
    - A module should contain declarations of variables and procedures used to manipulate these variables.
  - Facilitates sharing of data
    - Modules can use information from other modules through the FORTRAN 90 “use” statement.

```
MODULE kalman_mod
  USE matrix_mod
  USE param_mod

  TYPE hit_type
    REAL, DIMENSION(5,1):: Xp,Xf,Xs
    REAL, DIMENSION(5,5):: Cp, Cf, Cs
    REAL, DIMENSION(2,1):: Xm
  END TYPE hit_type
  TYPE(hit_type) ,DIMENSION(MAX_HITS):: KalHit

CONTAINS
  SUBROUTINE KalFit
    ------
  END SUBROUTINE KalFit

  SUBROUTINE KalInit
    ------
  END SUBROUTINE KalInit

  SUBROUTINE KalFilter
    ------
  END SUBROUTINE KalFilter

  SUBROUTINE KalSmoother
    ------
  END SUBROUTINE KalSmoother
END MODULE kalman_mod
```

Use statements

Type & variable declarations

Module procedures

Source Code Organization

- MOFIA is the TWIST analysis package
- Source code path

  ../e614soft/triumf/mofia/2.2/source

  ../e614soft/triumf/mofia/devel/source
  - **Subdirectory defined by MOFIA_SOURCE.**
  - **Example: cd $MOFIA_SOURCE.**

- Source code subdirectory

  ../source/main
  - **Code from experiment 787.**
  - **FORTRAN Fixed format (extention .F).**
  - **Subdirectory defined by MOFIA_MAIN.**

  ../source/dummy
  - **Dummy modules and subroutines.**
  - **Allow debugging only main for debugging purposes.**
  - **Subdirectory defined by MOFIA_DUMMY.**

  ../source/mainf90
  - **TWIST code that is fairly stable.**
  - **FORTRAN Free format (extension .f90) organized in modules.**
  - **Subdirectory defined by MOFIA_MAINF90.**

  ../source/user
  - **TWIST code under development and testing.**
  - **FORTRAN Free format (extension .f90).**
  - **Subdirectory defined by MOFIA_USER.**

  ../source/photo
  - **Event display code.**
  - **C++ and FORTRAN free format.**
  - **Subdirectory defined by MOFIA_USER.**

  ../source/include
  - **Include files, common blocks, parameter files, and interface blocks.**
  - **Extensions .inc, .cmn, .par, and .ifb.**
  - **Subdirectory defined by MOFIA_INCLUDE.**

  ../source/modules
  - **Module files used for compiling.**

Compiling MOFIA

- Compiling scripts
  - Compiling scripts are in the directory
    …/e614soft/triumf/mofia/2.**2**
  - To compile from scratch (or clean everything and compile again)
    - Use the script make_all by issuing the command make_all in that directory
- Compiling code in the *…/source/user* subdirectory
  - Often the user changes code only in the …./source/user subdirectory
  - Issue the command make from the *…/source/user* subdirectory
- Compiling code in the *.../source/mainf90*
  - Issue the command make from the *…/source/mainf90* subdirectory
  - Issue the command make from the *…/source/user* subdirectory
- Dependencies
  - Some files use information from other files through the use statement
  - The "used" files have to be compiled before the files that use them
  - The directory compiling order is
    - …/source/dummy
    - …/source/main
    - …/source/mainf90
    - …/source/photo
    - …/source/user
  - Similarly file order in the Makefile is important
    - Used files should precede those that use them

Running MOFIA

- ***To run MOFIA issue the command mofia or photo from the .../source/user subdirectory***

```
[quraan@maher user]$ mofia
 ------------------------------------------------------------ --------------
                          MOFIA 2.2
                     TWIST: TRIUMF E614
 ------------------------------------------------------------ --------------
 MODE: interactive
 NAME: mofia
 DATE: Thu May  9 16:57:39 2002
 TIME:
 MTIN: *** UNDEFINED ***
%CFM-I-CFMGET, CFM databases opened for reading
%CFM-I-CFMGET, CDTs obtained from CFM
 MOFIA>
```

- ***MOFIA commands can be entered on the MOFIA> command line***

MOFIA Commands

- **Assigning the input file:**

  MOFIA> MTIN "*argument*"

  - **Assign the input file/device to argument.**
  - **Example: MTIN "/twist/data3/datalinks/2100/run02121.ybs"**

- **Analyzing data:**

  MOFIA> analyze *argument*

  - **Start the analysis process. The number of events to be analyzed may be used as an argument**
  - **Example analyze 1000 analyzes 1000**
  - **If no argument, analyzes the entire file**

- **Analyzing a specific event:**

  MOFIA> event *argument*

  - **Move FORWARD to the event specified by "argument" and analyze it.**
  - **Example: event 237 moves forward to event 237 and analyzes it.**

- **Displaying information:**

  MOFIA> show *argument*

  - **Shows the current contents of "argument".**
  - **Examples: show MTIN**
  - **If no argument is provided, a listing of possible arguments is displayed.**

- **Exiting MOFIA:**

  MOFIA> exit

  - **Exit MOFIA**

- **Help:**

  MOFIA> help

  - **Run the MOFIA help facility.**

Namelists

- Namelist definition & commands
  - A collection of variables that can be referenced to by the group name.
  - Define groups referring to a set of cuts or settings for a detector subsystem, etc.
    - **Detector subsystems are parts of the detector such as drift chamber (DC).**
  - Example:
    - **Namelist /DCset/ FirstPlaneDC, LastPlaneDC**
    - **READ(*,NML = DCset)**
    - **WRITE(*,NML=DCset)**
- source/mainf90/namelist_mod.f90
  - Contains namelist definitions
  - Code to read them in
  - Code to print out their values and descriptions
- To access the namelists from a given subroutine/module
    - **USE namelists_mod**
  - Values for the namelist variables can be assigned from the MOFIA command line or read in from a command file (*.kcm file).

Namelists Description

MOFIA> show namelist

NAMELIST	DESCRIPTION
BATCH	BATCH LOG control parameters
HCUTS	DPLOT user cuts
DCSET	Drift Chamber SETtings
PCSET	Proportional Chamber SETtings
HIST	HISTogramming parameters
PHOTO	PHOTO flags
DCCFLAGS	Drift Chamber Calibration FLAGS
SCSET	SCintillator SETtings
SCCUTS	SCintillator CUTS
RFCUTS	RF CUTS
DCCUTS	Drift Chamber CUTS
PCCUTS	Proportional Chamber CUTS
KPUNIT	MOFIA print units
SCFLAGS	SCintillator FLAGS
KFLAGS	MOFIA execution control flags
PRCNTL	Print control flags
GLOBAL	GLOBAL settings
QOD	QOD Monitor params
STRSET	STR SETtings
KalmanCuts	Kalman Tracking Cuts
Alignment	Alignment parameters
FirstGuess	First Guess parameters
HelixFit	HelixFit parameters
TimeZero	Time zero fit settings
Efficiency	Efficiency settings
REZS	REZolutionS controls

Namelists Contents & Assignment

- Displaying contents:
  MOFIA> show name DCSET

  NameList DCSET: Drift Chamber SETtings

  FirstPlaneDC = 1 (dflt = 1): First DC plane
  LastPlaneDC = 44 (dflt = 44): Last DC plane

- Making assignments:
  MOFIA> namelist DCset
  Enter Drift Chamber SETtings:
  ?
  &NLDCSET
  FirstPlaneDC = 1,
  LastPlaneDC = 44,
  /
  LastPlaneDC = 22
  &
  MOFIA>

- Making assignments:
  MOFIA> namelist DCset
  Enter Drift Chamber SETtings:

  LastPlaneDC = 22
  &
  MOFIA>

- Making assignments:
  MOFIA> namelist Dcset LastPlaneDC = 22
  MOFIA>

Namelists Contents Description

MOFIA> show name hist

NameList HIST: HISTogramming parameters

nEVTprocessed	= -1 (dflt = -1,OFF):Write hists every nEVTprocessed events
FillRawHist	= T (dflt = T) : Fill raw histograms
FillHist	= T (dflt = T) : Fill histograms after initial filtering
FillTrackHist	= T (dflt = T) : Fill tracking histograms
FillPatternHist	= T (dflt = T) : Fill pattern recognition histograms
PulserHistToggle	= T (dflt = F) : Fill histograms with random pulser data(T) or normal triggers(F)
FillPhysicsHist	= T (dflt = T) : Fill physics histograms
FillFirstGuessHist	= F (dflt = T) : Fill helix first guess histograms
FillXtalkHist	= T (dflt = T) : Fill cross talk histograms
PlaneHists	= T (dflt = T) : Generate individual plane histograms
Globalmem_toggle	= T (dflt = T) : Turn Global Memory Section on (T) /off (F)
TDC_MIN	= 0. (dflt = 0.) : Lower limit on TDC spectra
TDC_MAX	= 6000. (dflt = 6000.) : Upper limit on TDC spectra
Raw_XMI_tdcslot	= 0. (dflt = 0.) : Lower limit on RAW TDC plots
Raw_XMA_tdcslot	= 30000. (dflt = 30000.) : Upper limit on RAW TDC plots
Raw_NX_tdcslot	= 3 (dflt = 3) : Number of RAW x Channels for TDC plots

MOFIA> show name DCSET

NameList DCSET: Drift Chamber SETtings

FirstPlaneDC = 1 (dflt = 1): First DC plane
LastPlaneDC = 44 (dflt = 44): Last DC plane

MOFIA> show name PCSET

NameList PCSET: Proportional Chamber SETtings

FirstPlanePC = 1 (dflt = 1): First PC plane
LastPlanePC = 12 (dflt = 12): Last PC plane

MOFIA> show name DCCFLAGS

NameList DCCFLAGS: Drift Chamber Calibration FLAGS

FindPlanePos = F (dflt = F) : Find plane positions
FindWirePos = F (dflt = F) : Find wire positions
FindPlaneRot = F (dflt = F) : Find plane rotations
FindTDC0 = F (dflt = F) : Find TDC time zero
FindResolution = F (dflt = F) : Find drift chamber resolution
FindbeamAngle =

MOFIA> show name dccuts

NameList DCCUTS: Drift Chamber CUTS

DC_MAXTDC_CUT = 6000.00 (dflt = -1, OFF): Max TDC channel cut
DC_MINTDC_CUT = 0.00 (dflt = -1, OFF): Min TDC channel cut
DC_MAXWTDC_CUT = 150.00 (dflt = -1, OFF): Max TDC width cut
DC_MINWTDC_CUT = -1.00 (dflt = -1, OFF): Min TDC width cut
DC_MAX_HITS_IN_PLANE = -1 (dflt = -1, OFF): Max hit wires in plane cut
DC_MIN_PLANES = -1 (dflt = -1, OFF): Min planes cut
DC_XTALK_WCUT = -1.00 (dflt = -1, OFF): Min cross talk TDC width cut
DC_NOISE_WCUT = -1.00 (dflt = -1, OFF): Min noise TDC width cut

MOFIA> show name pccuts

NameList PCCUTS: Proportional Chamber CUTS

PC_MAXTDC_CUT = 6000.00 (dflt = -1, OFF): Max TDC channel cut
PC_MINTDC_CUT = 0.00 (dflt = -1, OFF): Min TDC channel cut
PC_MAXWTDC_CUT = 150.00 (dflt = -1, OFF): Max TDC width cut
PC_MINWTDC_CUT = -1.00 (dflt = -1, OFF): Min TDC width cut
PC_MAX_HITS_IN_PLANE = -1 (dflt = -1, OFF): Max hit wires in plane cut
PC_MIN_PLANES = -1 (dflt = -1, OFF): Min planes cut
PC_XTALK_WCUT = -1.00 (dflt = -1, OFF): Min cross talk TDC width cut
PC_NOISE_WCUT = -1.00 (dflt = -1, OFF): Min noise TDC width cut

MOFIA> show name scset

NameList SCSET: SCintillator SETtings

S1_PEAK = 1000.00 (dflt = 1000) : Scintillator 1 peak position
S2_PEAK = 1000.00 (dflt = 1000) : Scintillator 2 peak position
S3_PEAK = 1000.00 (dflt = 1000) : Scintillator 3 peak position

MOFIA> show name sccuts

NameList SCCUTS: SCintillator CUTS

S1_MAX_NHITS = -1 (dflt = -1, OFF): Scint 1 max number of hits cut
S2_MAX_NHITS = -1 (dflt = -1, OFF): Scint 2 max number of hits cut
S3_MAX_NHITS = -1 (dflt = -1, OFF): Scint 3 max number of hits cut
S1_WIDTH_CUT = -1.00 (dflt = -1, OFF): Scint 1 peak width cut
S2_WIDTH_CUT = -1.00 (dflt = -1, OFF): Scint 2 peak width cut
S3_WIDTH_CUT = -1.00 (dflt = -1, OFF): Scint 2 peak width cut
S1_MAX_TDC = ***** (dflt = -1, OFF): Scint 3 peak width cut**
S2_MAX_TDC = ***** (dflt = -1, OFF): Scint 1 max TDC channel cut**
S3_MAX_TDC = 2500.00 (dflt = -1, OFF): Scint 2 max TDC channel cut
S1_MIN_TDC = -1.00 (dflt = -1, OFF): Scint 1 min TDC channel cut
S2_MIN_TDC = -1.00 (dflt = -1, OFF): Scint 2 min TDC channel cut
S3_MIN_TDC = -1.00 (dflt = -1, OFF): Scint 2 min TDC channel cut
S1_MAX_ADC = 5000.00 (dflt = -1, OFF): Scint 1 max ADC channel cut
S2_MAX_ADC = 5000.00 (dflt = -1, OFF): Scint 2 max ADC channel cut
S1_MIN_ADC = -1.00 (dflt = -1, OFF): Scint 1 min ADC channel cut
S2_MIN_ADC = -1.00 (dflt = -1, OFF): Scint 2 min ADC channel cut

MOFIA> show name scflags

NameList SCFLAGS: SCintillator FLAGS

S1_signal = T (dflt = TRUE) : Require a signal in S1
S2_signal = T (dflt = TRUE) : Require a signal in S2
S3_signal = T (dflt = TRUE) : Require a signal in S3

MOFIA> show name kalmancuts

NameList KalmanCuts: Kalman tracking cuts

ChiDiffCluster (def = 1.E-02) = 0.10E-01 Chi2 convergence level for cluster iteration
ChiDiffTime (def = 1.E-04) = 0.10E-03 Chi2 convergence level for timing iteration
Chi2sCutCluster (def = 1.E05) = 0.10E+06 Chi2 cut for cluster iteration
Chi2sCutTime (def = 1.E05) = 0.10E+06 Chi2 cut for timing iteration
MaxIterateCluster (def = 50) = 50 Maximum number of iterations for cluster fit
MaxIterateTime (def = 50) = 50 Maximum number of iterations for timing fit
EnableKalman (def = TRUE) = T Execute Kalman filterng code
SwitchLR = (def = TRUE) = T Attempt reducing the value of Chi2 by switching left and right
NoiseExcludeMax (def = 2) =

MOFIA> show name strset

use_cos_increments (def= T) T
str_angle_inc (def= 0.05) 5.000000E-02
str_upper_angle_limit (def= 85. degrees) 85.0000

MOFIA> show name scflags

NameList SCFLAGS: SCintillator FLAGS

S1_signal = T (dflt = TRUE) : Require a signal in S1
S2_signal = T (dflt = TRUE) : Require a signal in S2
S3_signal = T (dflt = TRUE) : Require a signal in S3

MOFIA> show name efficiency

Namelist Efficiency: Efficiency calculation parameters

RadiusCutDense = 5.000 (dflt = 15 cm): DC dense stack radius cut
RadiusCutSparse = 15.000 (dflt = 15 cm): DC sparse stack radius cut
CellCut = 2 (dflt = 2): Max number of adjacent cells to investigate on each side of the cell expected to have a hit
CellCut = 10 (dflt = 10): Minimum number of hit planes
FindEff = F (dflt = .FALSE.): Calculate chamber efficiency

MOFIA> show name rfcuts

NameList RFCUTS: RF CUTS

RF_MIN_TDC = -1.00 (dflt = -1, OFF) : RF min TDC cut
RF_MAX_TDC = -1.00 (dflt = -1, OFF) : RF max TDC cut

MOFIA> show name global

BField (def = 0.) 0.000000
UnpackMC = F

MOFIA> show name rezs

Namelist REZS: Resolutions

nEventsMax = 5000 (dflt = 5000): # Events per Resolution Iteration.
MinHistFitEntries = 1000 (dflt = 1000): # histogram entries necessary to do fit.
nResidualBins = 300 (dflt = 300): # Bins in residual histograms.
MinResidualValue = 0.30 (dflt = -0.3): Min histogram channel (cm).
MaxResidualValue = 0.00 (dflt = 0.3): Max histogram channel (cm).
calcResidualMethod = * (dflt = 3): Residual calculation method.**
1=> Gaussian Fit to residual histograms.
2=> SquaredSum calculation.
3=> FWHM calculation.
peakTOL =

MOFIA> show name alignment

NameList Alignment: Alignment parameters

nEventsPlane (def = 10000) = 10000 Total number of events per iteration for calculating plane positions
nEventsWire(def = 100000) = 100000 Total number of events per iteration for calculating wire positions
AlignAngleU (def = 0.0) = 0.000 Angle of the U planes transverse-alignment line with respect to the beam
AlignAngleV (def = 0.0) = 0.000 Angle of the V planes transverse-alignment line with respect to the beam
FixPlanes (def = FALSE) = F Choose a line for transverse alignment defined by any 2 U and 2 V planes
FixedPlane1 (def = 1) = 1 Plane number for one of the four fixed planes used to define the alignment line
FixedPlane1 (def = 1) = 2 Plane number for one of the four fixed planes used to define the alignment line
FixedPlane1 (def = 1) = 7 Plane number for one of the four fixed planes used to define the alignment line
FixedPlane1 (def = 1) = 8 Plane number for one of the four fixed planes used to define the alignment line

MOFIA Functions

MOFIA> fun

Description of FUNC:

1 Turn ON printing for list of wire HITS
2 Turn OFF printing " "
3 Turn ON printing for TDCUNP debugging
4 Turn OFF printing " "
5 Write HBOOK histograms to disk
6 Print geometry files
7 Print mapped variables for current event
9 Reset all histograms
10 Initialize cross talk counters
11 Print cross talk counters
12 Print efficiency counters
13 Print residuals
14 Initialize efficiency counters
20 Determine and output t_0

Example: MOFIA> fun 12

Command Files

- Mofia> @filename
  - Execute the command file specified by filename.
  - Command files may include any MOFIA command that may otherwise be issued at the MOFIA command line.
  - If the file extension is not specified, the extension .kcm is assumed. Command files may also be nested (so that a command file may call another) up to 10 layers deep

- To keep a record of cuts/settings in mofialog.dat
  - show namelist “name”
  - make assignments to namelist variables
  - show namelist “name”

Calibration File Manager

- CFM allows the user to associate run numbers with calibration files.
- CFM> show types

1:DC_PPC	**Plane Position Corrections**	**10:DC_WZC**	**DC Wire Z position Corrections**
2:DC_WPC	**Wire Position Corrections**	**11:DC_WRC**	**DC Wire Rotation Corrections**
3:DC_RES	**DC Resolution**	**12:DC_STR**	**DC Space Time Relations**
4:DC_PRC	**Plane Rotation Corrections**	**13:PC_T0**	**PC Time Zero**
5:DT_GEO	**Detector Geometry**	**14:DC_T0**	**DC Time Zero**
6:FBC1_MAP	**Fast Bus Map, Crate 1**	**15:SC_T0**	**SC Time Zero**
7:FBC2_MAP	**Fast Bus Map, Crate 2**	**16:PC_ADC**	**PC ADC Calibrations**
8:FBC3_MAP	**Fast Bus Map, Crate 3**	**17:SC_ADC**	**SC ADC Calibrations**
9:DC_PZC	**DC Z Position Corrections**		

Calibration File Manager (CFM)

CFM>show runs

Runs	Total	CDF Set
1 : 4	4	1
5 : 6	2	2
7 : 20	14	3
21 : 499	479	--
500 : 800	301	4
801 : 922	122	5
923 : 924	2	6
925 : 932	8	7
933 : 955	23	8
956	1	9
957 : 1311	355	8
1312 : 1500	189	12
1501 : 1693	193	--
1694 : 2501	808	13
2502 : 2658	157	14
2659 : 3158	500	15
3159 : 3226	68	16
3227 : 3231	5	17
3232 : 5000	1769	18

CFM>show set 17

CDF Set	Run Total	Run Range
17	5	3227: 3231

CDF Names:

DC_PPC.00003	DC_STR.00010
DC_WZC.00003	FBC2_MAP.00018
PC_T0.00004	DC_PRC.00003
DC_T0.00006	DT_GEO.00024
FBC3_MAP	SC_ADC.00001
DC_PZC.00003	DC_WPC.00003
FBC1_MAP.00018	PC_ADC.00001
SC_T0.0001	DC_RES.00004
DC_WRC.00003	

- All CFM files sit in the directory defined by $CAL_DB
  - cd $CAL_DB
- All files are in ascii
  - ls dc_ppc.00003

Over-riding CFM

- MOFIA will use the calibration files associated with a run number as specified in CFM

- To over ride CFM (i.e.use a different calibration file) at the MOFIA command line enter

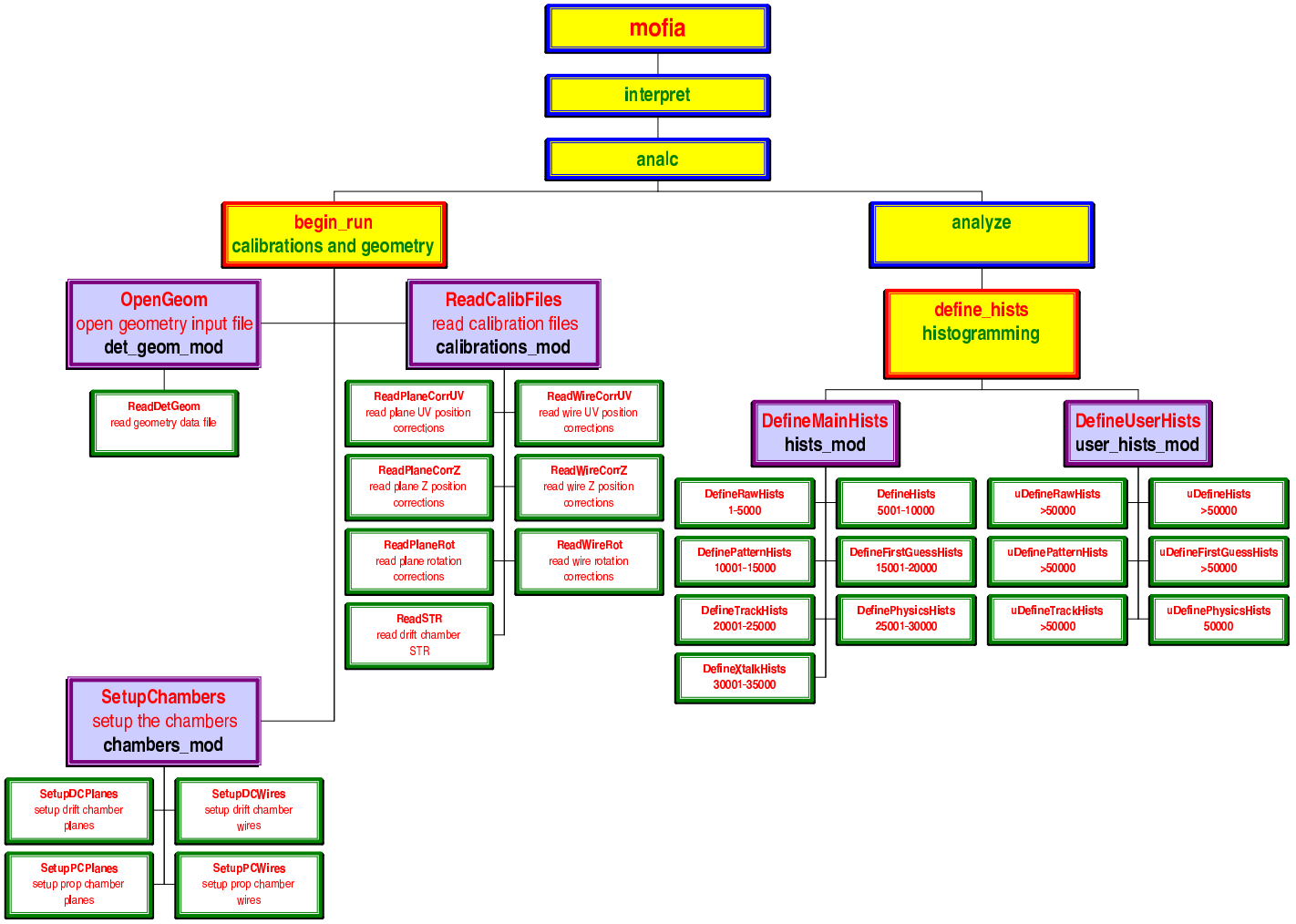
  MOFIA> use dc ppc “testfile.dat”

  – As usual, can do this in the command file as well
- CFM is not installed in Monte Carlo
  – Calibration files have to be explicitly specified in the .kcm file

Example of a command file

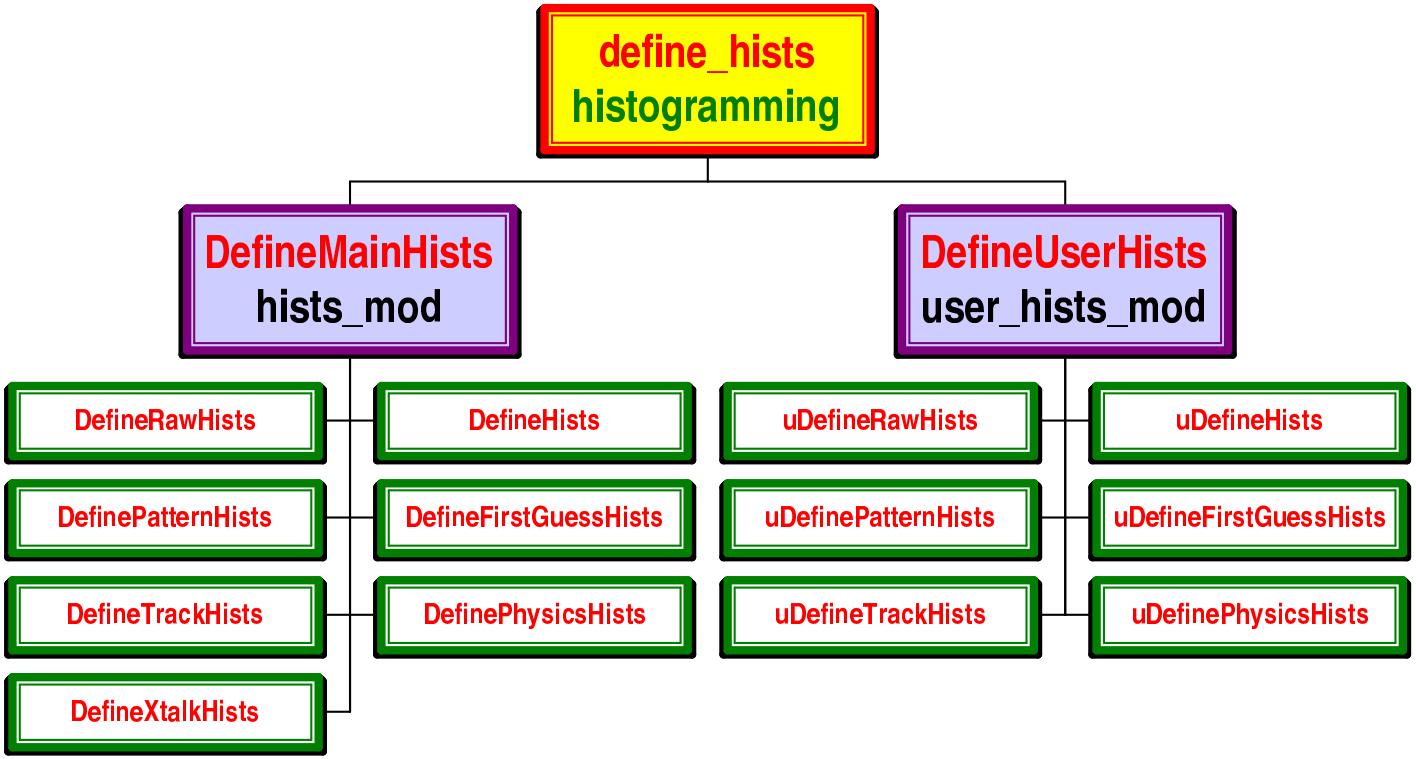
```
use dc str "/home/e614/e614soft/caldb_ascii/dc_str.00010"
use pc t0 "/home/e614/e614soft/caldb_ascii/pc_t0.00004"
use dc ppc "dc_ppc.fall"
use dc res  "dc_res.fall”

set dc/cuts on
set pc/cuts off
set sc/cuts off
show flags

namelist DCCFLAGS
FindResolution = .FALSE.
FindPlanePos = .FALSE.
FindWirePos = .FALSE.
FindPlaneRot = .FALSE.
FindTDC0 = .FALSE.
&
show name DCCFLAGS

namelist hist
NEVTPROCESSED = 10000
TDC_MIN = -100
TDC_MAX = 900
PlaneHists = .TRUE.
FillRawHist = .TRUE.
FillHist = .TRUE.
FillPatternHist = .TRUE.
FillFirstGuessHist = .TRUE.
FillTrackHist = .TRUE.
FillPhysicsHist = .TRUE.
FillXtalkHist = .TRUE.
&
show name hist

name dccuts
DC_MAXTDC_CUT = 800.
DC_MINTDC_CUT = 0.
DC_MAXWTDC_CUT    = -1.
DC_MINWTDC_CUT    = -1.
DC_MAX_HITS_IN_PLANE = 1
DC_MIN_PLANES      = 14
DC_XTALK_WCUT    = 100.
DC_NOISE_WCUT    = 100.
&
show name dccuts

!PC_MAXTDC_CUT = 100000.
!PC_MINTDC_CUT = -100000.

name pccuts
PC_MAXWTDC_CUT    = -1.
PC_MINWTDC_CUT    = -1.
PC_MAX_HITS_IN_PLANE = 3
PC_MIN_PLANES      = 2
PC_XTALK_WCUT    = 0.
PC_NOISE_WCUT    = 0.
&
show name pccuts

name dcset
FirstPlaneDC = 23
LastPlaneDC = 44
&
show name dcset
```

MOFIA Initializarion Branch



MOFIA Histogramming Branch

- **Two histogramming modules**
  - **…/source/mainf90/hists_mod.f90**
  - **…/source/user/user_hists_mod.f90**



Histogramming Example

- **ID declaration**

  INTEGER(i4), PARAMETER:: IDH_TDC_ALL = 3

  Advantages:

  1. **Need to only change Ids in one place.**
  2. **Reduces potential for define/fill mismatches.**
  3. **Easy to avoid conflicts if done in order.**

- **Defining a raw 1-D histogram**
  - **Define in subroutine DefineRawHists in module hists_mod.f90**

  ! TDC time (total)

  CALL HBOOK1(IDH_TDC_ALL, 'DC TDC TIME', TDC_MAX-TDC_MIN+1,REAL(TDC_MIN), REAL(TDC_MAX),0.0)

  - **First parameter: histogram ID.**
  - **Second parameter: histogram label (to appear on the figure).**
  - **Third parameter: number of bins (can be an expression).**
  - **Fourth parameter: lower limit.**
  - **Fifth parameter: upper limit.**

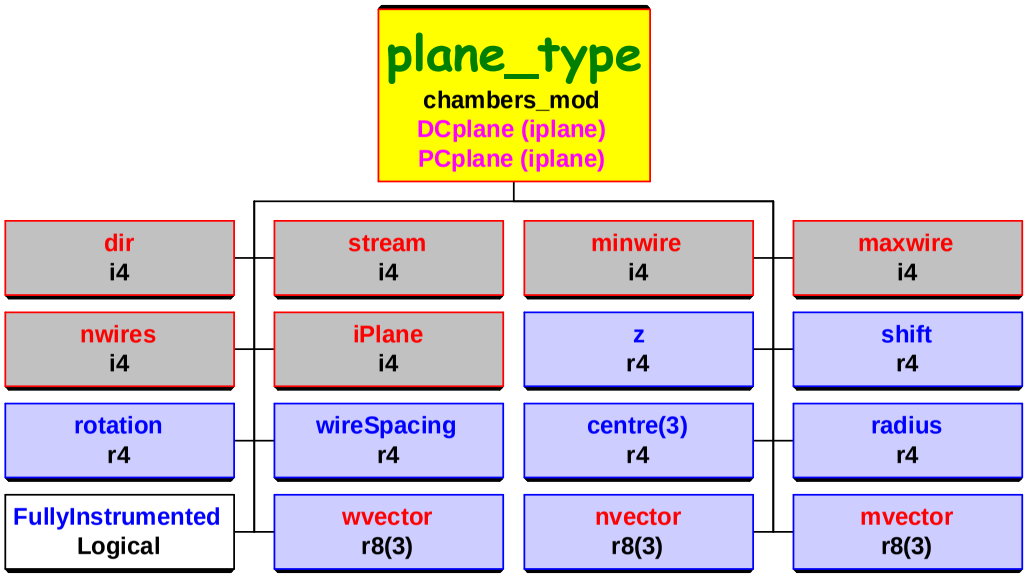
- **Filling a raw 1-D histogram**
  - **Define in subroutine FillRawHists in module hists_mod.f90**

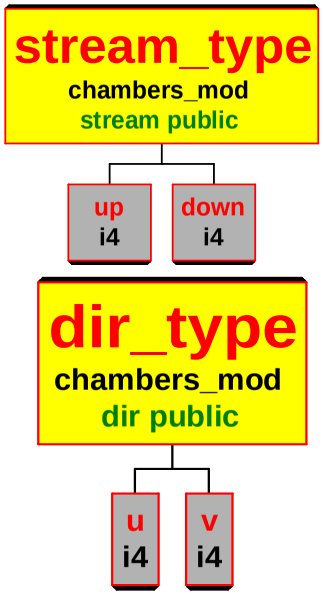
  ! Raw TDC time (total)

  CALL HFILL(IDH_TDC_ALL, time, 0.0, 1.0)

  - **First parameter: histogram ID.**
  - **Second parameter: variable containing TDC time (must be real). If time is an integer send REAL(time) instead.**
  - **Third parameter: second variable (non-zero for a 2-D histograms)**
  - **Fourth parameter: weight factor.**

Using the Data Structures Figures






```
TYPE(PLANE_TYPE), POINTER:: planeP
PlaneLoop: DO iPlane = FirstPlaneDC, LastPlaneDC
     planeP => DCplane(iPlane)
      DO iWire = planeP%MinWire, planeP%MaxWire
         IF(.NOT.planeP%FullyInstrumented) THEN
            IF(planeP%stream = = stream%up) &
               PRINT*, 'Plane',iPlane,'is in the upstream dense stack'
         ENDIF
      END DO
END DO PlaneLoop
```

More Data Structures