KOFIA
Kaon Offline Interactive Analysis

Morgan Burke (TRIUMF) **1992-Dec-18**

1.1 Introduction

What is KOFIA?

`KOFIA` is the Kaon Offline Interactive Analysis program, written, maintained, and used by Brookhaven Experiment 787. It is used to process and analyze data from the 787 detector, and provides an assortment of basic analysis functions, including histogramming, event sorting and skimming, and graphical event displays. Other capabilities can be introduced by using optional analysis libraries and custom routines.

Actually, `KOFIA` is little more than a collection of libraries and routines which are linked together by the user, along with his own custom routines, to produce an executable program which is especially tailored for a specific analysis task. The various libraries contain routines grouped by either general function (*eg.* data unpacking, data calibration) or by physical association in the detector (*eg.* drift chamber, target). In addition, there are a number of general support libraries which contain necessary and useful subroutines and functions, but which were not specifically written for 787.

Data from the E787 detector is encoded in series of banks, managed by a software system called YBOS, which comes from the CDF computing group at Fermilab[1]. The primary job of `KOFIA` is to read and translate these YBOS banks into a more accessible format, and then to process the translated data according to analysis instructions provided by the user. Finally, data which is deemed satisfactory can be re-written to files in YBOS form, for deeper analysis by other versions of `KOFIA`. The YBOS bank structure is extremely flexible, and allows the user to create and delete banks of data as he or she sees fit.

`KOFIA` can also be configured to be used online, to pre-process the data coming from the detector in real time. In this mode, `KOFIA` is used to perform substantial amounts of data reduction before recording the results on tape, by compacting data and rejecting events that don't meet the experimenters' criteria for meaningful data.

Where is KOFIA?

Before you can use `KOFIA`, you will have to define various required logical names by typing the command:

```
$ SETUP KOFIA [version]
```

where `version` is the version number (*eg.* 1.4, 1.5) of `KOFIA` that you want to use. The version numbers increase approximately once per year, partially to reflect detector hardware changes in each year's data-taking run. The year-version correlations are as follows:

Year	Version
1989	1.3
1990	1.4
1991	1.5
1992	1.6
1993	2.0

Versions are moderately backwards-compatible, except for certain libraries. Leaving out the `version` specification in the `SETUP KOFIA` command will set up the current production version of `KOFIA`, usually corresponding to the previous year's data-taking run.

Note that the `SETUP` command will not work unless the user has already entered the command:

```
$ @PRVO:[TRIUMF]SETUP
```

Most users place this command in their `LOGIN.COM` file, so that it is automatically executed whenever they log in.

[1] see the document *YBOS at TRIUMF,* available from Corrie Kost, room Chem 209.

Simple versions of KOFIA are available for public use, although their analysis capabilities are limited. They are usually used to look at raw data, perform simple data-skimming tasks, or look at event displays. These public executables are:

E787K$MAIN:KOFIA.EXE, and
E787K$PHOTO:PUBLIC_PHOTO.EXE

The former is a bare-bones version of KOFIA, while the latter can be used to generate photos, or graphical event displays.

The various KOFIA libraries and source files are located in a complex of directories, shown below:

KOFIA Library	Function
KOFIA$BW:BWPC	Beam Chamber Analysis
KOFIA$CAL:KOFIA_CAL	Data Calibration
KOFIA$CCD:KOFIA_CCD	Charge Coupled Devices
KOFIA$DC:DC_DECODE	Drift Chamber Analysis
KOFIA$DC:DCTRACK	Track Fitting
KOFIA$DUMMY:KOFIA_DUMMY	Dummy Routines
KOFIA$IWC:KOFIA_IWC	Inner Wire Chamber Analysis
KOFIA$MAIN:KOFIA_MAIN	KOFIA User Interface and Drivers
KOFIA$PASS1:PASS1	First Pass Analysis
KOFIA$PHOTO:PHOTO	Graphical Event Displays
KOFIA$RS:RANGE	Range Routines
KOFIA$TARGET:TARGET	Target Analysis
KOFIA$TD:KOFIA_TD	Transient Digitizer Analysis
KOFIA$TT:KOFIA_TD	New Target Data Unpacking
KOFIA$UNP:KOFIA_UNP	Data Unpacking

Additional Support Libraries:

BRU$LIBRARY:BRU	Utility Routines
CERN$LIBRARY:CERNLIB	Utility Routines
CFM$MAIN:CFM	Calibration File Management
EDGR$DIR:EDGR	Graphical Editor
FIOWA$DIR:FIOWA	Histogramming Package
IO$DIR:LIB	Magnetic Tape Handling
KOST$DIRECTORY:LIB	Utility Routines
REPLAY$DIR:LIB	Histogram Displays
YBOS$LIBRARY:YBOS	Data Management Routines

Which libraries the user chooses to link to depends on the analysis task to be performed. As a minimum, one must link KOFIA$MAIN:KOFIA.OBJ to the KOFIA_MAIN and KOFIA_DUMMY libraries, and several additional support libraries. For example, the following command will link a basic version of KOFIA:

```
$ LINK KOFIA$MAIN:KOFIA,KOFIA_MAIN/LIB,CFM$MAIN:CFM/LIB,BRU$LIBRARY:BRU/LIB,-
_$ FIOWA$DIR:FIOWA/LIB,KOFIA$DUMMY:KOFIA_DUMMY/LIB,KOST$DIR:LIB/LIB,-
_$ YBOS$LIBRARY:YBOS/LIB,IO$DIR:LIB/LIB,CERN$LIBRARY:KERNLIB/LIB
```

All other libraries are optional, since they are only used in specific analysis applications. A standard link command file, E787K$MAIN:KOFIA.LNK, which can be executed in place of the above command, or copied and edited first, is available. In addition there is a similar command file for linking a version with photos: E787K$PHOTO:PUBLIC_PHOTO.LNK.

1.2 Customizing KOFIA

The various KOFIA libraries comprise a large toolbox of analysis routines. Experimenters are expected to write their own routines to utilize and tie together all these tools in a way that is suitable to them. Various empty ("dummy") routines are placed at strategic points throughout KOFIA, expressly so that they may be replaced with custom user code for this purpose. It is also possible to replace any standard routines or libraries with custom versions, although users who attempt this should take care to ensure that the replacements produce proper results.

Custom user routines can be used to perform a special analysis of the data, monitor the data or analysis, build histograms, or do just about anything else the user desires. By accessing the common blocks where `KOFIA` stores its raw and processed data, these routines can have access to all the information available on the current event. Of course, calls to other custom-written routines, as well as standard analysis routines, may be made from the user routines, which allows the level of complexity of the customizations to become as high as necessary. The user routines called directly by `KOFIA` are as follows:

`DEFINE:` This routine is used to define and initialize all FIOWA[2] histograms and scatterplots which will be filled in later analysis.

`DPLOT:` This routine is called after `KOFIA` has extracted an event from a file of raw data, and performed its own analysis, if any. Further analysis of the data, and the histogramming of the results[2] is normally done from here.

`EXEC_ON_EXIT:` This routine is called just before `KOFIA` finishes. It can be used to clean up and finish the analysis, or display a summary of the work that was done over the course of the session.

`FUNC:` This is the only user routine to take an argument (one argument of type `INTEGER*4`). It is called when a user gives the interactive command "`FUNC` n" to `KOFIA`, and the number n is passed as the argument to `SUBROUTINE FUNC(n)`. `FUNC` can be used for various purposes, such as additional analysis, displaying results, or writing output to disk. Depending on the value of n, it can even be configured to perform several completely different tasks.

`MY_BEGIN_RUN:` This routine is called when a `BEGIN_RUN` record is read from the data file. The begin run record signals that a new set of events is about to follow, possibly taken with different triggers and calibrations. This routine is commonly used to reset counters and other variables.

`MY_END_RUN:` Similar to `MY_BEGIN_RUN`, `MY_END_RUN` is called at the end of a run of data. It is commonly used to compile and/or display statistics gathered on the run just analyzed.

`MY_END_SPILL:` This routine is called following an `NSPILL` record. A spill can be described as a short burst of beam, which produces a collection of events. It is following each such cluster that `MY_END_SPILL` is called; otherwise it is similar to `MY_END_RUN`.

`MY_CMC_STATUS:` This routine is called when a CMC record is read from the input file. Otherwise, it is similar to `MY_BEGIN_RUN`.

Unless otherwise stated, the user routines do not take arguments. The dummy equivalents of these routines are part of the `KOFIA_MAIN` library.

1.3 Running KOFIA:

Before `KOFIA` can begin analysis, it needs to know the names of various data files, and other information about the user's session. This information can be passed to `KOFIA` through logical names that the user defines before he or she begins the `KOFIA` session. In many cases, the information can be given to `KOFIA` after a session is started, but it is usually more desirable to pre-define them, so the definitions aren't lost after the user exits. The logicals that `KOFIA` uses are as follows:

`KOFIA$HIS:` This is the name of a file to which all histogram data can optionally be written. If undefined, `KOFIA` invents its own file name, and locates the file in the directory defined by the logical `KHIST`.

`KOFIA$INIT:` This is the name of a file containing a list of `KOFIA` commands that should be automatically executed on start-up. File name extensions of "`.KCM`" are optional.

`KOFIA$KCMDIR:` This logical stores the name of a directory in which to search for `KOFIA` command files, if they are not found in the default directory. Users who have a number of standard command files

[2] consult the *FIOWA* users' guide, available from Corrie Kost in room Chem 209

may want to define this logical so they don't always have to type in the name of the directory the command files are stored in.

`KOFIALOG:` `KOFIA` keeps a log of all terminal input and output during the user's session. This log is written to a file given by the definition of `KOFIALOG`. The default is "`KOFIALOG.DAT`".

`MTIN:` This is the file which contains your input data. `KOFIA` will not be able to do much until this is defined, and there is no default. Tape drives can be specified by defining `MTIN` to be the tape device name.

`MTOUT:` This is the name of your "skim" file, or the file to which selected data is output when in single skimming mode (*cf.*). This data file is identical in format to that of `MTIN`, except that it only contains those events you have elected to keep. There is no default output file.

`TRIUMF_PLOTTER_TYPE:` This logical gives the type of plotter you will be getting graphics hardcopy from. This information is required by the photo routines to know what commands to send to the hardcopy device. Possible definitions include "`HP`", "`QMS`", and "`ZETA`". The default plotter type is `HP`.

`TRIUMF_TERMINAL_TYPE:` This logical gives the type of terminal you are running on. This information is required by the photo routines to know what kind of graphics your terminal can display. Possible definitions include "`VT640`", "`CIT467`", and "`DECWINDOWS`". The default terminal type is `VT640`.

To define these outside `KOFIA`, use one of the commands:

> `$ DEFINE` [*logical_name definition*]
> `$ ASSIGN` [*definition logical_name*]

where *logical_name* is one of the above logical names, and *definition* is your definition of the logical. Inside `KOFIA`, `MTIN` and `MTOUT` can be defined by merely typing one of:

> `KOFIA> MTIN` [*definition*]
> `KOFIA> MTOUT` [*definition*]

1.4 KOFIA Commands

`KOFIA` has a command line interface, which supports over 40 commands, many of which have numerous VMS-style keywords and qualifiers for additional variety. If the user has linked in the `REPLAY` histogram display package, `KOFIA` will also accept `REPLAY` commands.

`KOFIA` *command summary*:

@file_name	Execute `KOFIA` command file
ANALYZE	Analyze events
CFM	Open the CFM databases
CLEAR	Clear text and graphics screens
CLOSE	Close skim output device
DCL	Spawn a subprocess
DEFINE	Redefine FIOWA histograms and scatterplots
DUMP	Produce dump of YBOS banks
EVENT	Analyze specfied event
EXIT	Leave `KOFIA`
FILES	Specify how many files to analyze
FUNC	Call user supplied routine
HELP	Get online help with KOFIA commands
ICFAIL	Show failure codes
INIT	Initialize ANALC and ANALYZE routines
KHIST	Define `KHIST` directory
LIST	Display analysis results
MAP	Map FIOWA common blocks
MTIN	Assign input device
MTOUT	Assign skim output device
NAMELIST	Access namelist variables

NOTRAP	Turn OFF trap handler
PHOTO	Graphically display current event
PLTR	Set plotter type
PRINT	Print specified histograms
QUIT	Exit program
RELEASE	Close `KOFIALOG`, and open a new one
RESTORE	Restore FIOWA histograms and scatterplots
REWIND	Rewind input device to load point
RUNS	Specify which runs to analyze
SAVE	Save FIOWA histograms and scatterplots
SET	Set various flags
SHOW	Show contents of specified namelist
SKIM	Set up multi-channel skimming
SKIP	Skip specified number of events or files
SUM	ACP histogram summing
TERMTYPE	Set terminal type
TRAP	Set trap handler ON
UNMAP	Unmap FIOWA common blocks
USE	Use a non-CFM calibration file
WIDTH	Set screen width
ZERO	Zero histograms and statistics

`KOFIA` uses a command recall shell that allows users to retrieve previously typed commands using the up and down arrow keys, and define "hot" keys to input repeated commands that would be a nuisance to type over and over. Hit the `PF2` key while on the `KOFIA` command line for help on using these and other features of this shell. Type `CTRL-K` to diable the shell and enter a trivial command line editing mode. This can be useful if you are running `KOFIA` over a slow network link and want to speed up your terminal response.

Each command may be abbreviated down to a few characters, to save typing. In the command descriptions that follow, the minimum abbreviation of each command is shown in `UPPER CASE`. Optional arguments and qualifiers are enclosed in brackets.

KOFIA Command Descriptions

@

Syntax: `@file_name[.kcm]`

`KOFIA` commands will be read from the specified file and executed. The default extension for KOFIA command files is ".KCM". Command files may be nested up to 10 deep. If KOFIA is running in BATCH mode, fatal errors will cause the program to stop execution immediately.

WARNING: due to terminal handling differences between `KOFIA` and the command interpreters in the Photo routines, errors may occur if photos are plotted from within command files.

ANALYZE

Syntax: `ANalyze [n]`

`KOFIA` will analyze the next `n` events from the run(s) specified by the `RUNS` or `FILE` command. If `n` is not specified, `KOFIA` will analyze up to the next end-of-file marker.

CFM

Syntax: `CFM`

Open the CFM databases. Normally the databases are opened automatically when `KOFIA` starts up. If they cannot be opened, this command can be used to try again, once the problem has been corrected.

CLEAR

Syntax: `CLear`
The screen is cleared of both graphics and text.

CLOSE

Syntax: `CLOse`

This will write two end-of-file markers to the current output file (*ie.* **MTOUT**), and close the file. The **SKIM** flag will also be turned off. The output file is normally closed when you exit **KOFIA**, or when an end-of-device is encountered on the input file.

DCL

Syntax: `DCL`

This commands spawns a DCL subprocess, from which any DCL commands may be entered, including requests to run other programs. All files that were opened by **KOFIA** will not be useable, however. Furthermore, device allocations specific to the original process do not apply to the subprocess. To return to **KOFIA**, log out of the subprocess with the DCL command **LOGOUT**. This command is only available on VMS implementations of **KOFIA**.

DEFINE

Syntax: `DEFine`

Re-define FIOWA histograms and scatterplots. The FIOWA commons must be zeroed, first (see the **ZERO** command). This can be useful if the histograms and scatterplots have interactively-defined parameters (probably via the **HCUTS** namelist) which have been changed during the course of the **KOFIA** session.

DUMP

Syntax: `DUmp[/format][/I2][/LIST] [bank_name[[,]bank_name]...]`

This command displays the specified YBOS banks on the terminal (and log file), according to the format specified with the `/format` qualifier. If no bank names are specified then all of the banks are dumped.

`/format` can be one of **ASCII**, **DECIMAL**, **HEX**, **OCTAL**, or **REAL**. **DECIMAL** is the default. The qualifier **I2** displays the data as two-byte integers, and can be used in conjunction with the **OCTAL** or **HEX** qualifiers. The **LIST** qualifier lists all YBOS banks names in the A list of the current event. Consult the YBOS manual for more information.

Example: `DUMP/HEX TDPT,TDDT`

EVENT

Syntax: `EVent n`

KOFIA will search for the given event number, and **ANALYZE** it, if found. **KOFIA** will only search the data file in a forward direction (*cf.* **REWIND**).

EXIT

Syntax: `EXIT`

Terminate the **KOFIA** session.

FILES

Syntax: `FILes n`

Process the next `n` files of data without stopping between runs, when the next **ANALYZE** command is given.

FUNC

Syntax: `FUnc [n]`

Call the **FUNC** user routine, and pass it the **INTEGER*4** argument `n`. If `n` is not specified, it defaults to `0`. **FUNC** does nothing unless the user provides a routine to define it (*cf.* Customizing **KOFIA**, above).

HELP

Syntax: `HELP [command]`

Invoke the VMS Help facility to provide online help on **KOFIA** concepts and commands. This command is only available on VMS machines.

ICFAIL

Syntax: `ICFail`

Display the ICFAIL statistics.

INIT

Syntax: `INIt`

Perform the initialization part of the `ANALYZE` command only. This causes some failure descriptions to get set up, and the routine `FIOWA_INIT` to get called.

KHIST

Syntax: `KHIST`

Define the directory where FIOWA save files will be stored (*cf.* `SAVE` command). The default save directory is taken from the logical name `KHIST`.

LIST

Syntax: `LIst [/[NO]ECHO] [/HARD] [/ss [/data] ... ] [itrack]`

List analysis results. The `ECHO` and `NOECHO` qualifiers control whether or not the listing is written to the screen and log file (`ECHO` is the default). The `HARD` qualifier can be used to send output to a text file. The listing will contain information referred to by the qualifier(s) `data`, from the subsystem(s) `ss`. The subsystems `BW` and `DC` are recognized. `Data` can be one or more of the following:

`BOUNDS`	cell bounds
`DIP`	dip angle corrections
`DV`	drift velocities
`FD`	field distortions
`LA`	lorentz angles
`PED[/ALL]`	pedestals
`POS`	position resolutions
`RAW`	raw TDC information
`SEC`	corrections for sector-dependence of momentum
`STAG`	wire staggers
`TA`	$T \cdot A$ values
`WIRE`	wire positions at $Z = 0$
`XY`	X-Y fit specifics
`XYZ`	X-Y-Z fit specifics
`Z`	Z fit specifics

Of these, only `RAW`, `TA`, and `XY` can be used with the `BW` subsystem. The argument `itrack` is only meaningful when used with the `/DC/XY`, `/DC/XYZ`, or `/DC/Z` options; if not given, data for all tracks is listed.

All meaningful permutations and combinations of the qualifiers will be processed. For example, the command "`LIST/DC/XY/RAW/BW/PED 1`" is equivalent to all of the following commands together:

```
LIST/DC/PED
LIST/DC/RAW
LIST/DC/XY 1
LIST/BW/RAW
LIST/BW/XY
```

MAP

Syntax: `MAP global_section_name`

Allow other processes to access the FIOWA common blocks concurrently with `KOFIA`. Normally this feature is used to allow REPLAY to display histograms and scatterplots while `KOFIA` is still filling them. Any name can be provided for the global_section_name, and a file with this name (plus the extension ".DAT") will be created in the default directory (it can be deleted after the user finishes the session).

The REPLAY commands "/MAP global_section_name" and "/UNMAP" can be used to select and deselect these FIOWA common blocks for display, and hop between several different running KOFIA sessions.

The KOFIA command UNMAP should be used before the user exits the KOFIA session.

MTIN

Syntax: `MTIn[/qualifier] file_name`

Define the input device/file containing the data to be analyzed. The qualifiers REWIND (the default) and NOREWIND are accepted. The input device defaults to the value of the logical name MTIN.

MTOUT

Syntax: `MTOut MTOut file_name`

Define the output device/file to which single skimming data is to be written. The output device defaults to the value of the logical name MTOUT. MTOUT cannot be reassigned until the previous version is closed (*cf.* CLOSE).

NAMELIST

Syntax: `NAMElist [name]`

Invoke the namelist editor which can be used to interactively display and alter the values of variables that control KOFIA analysis and control flow. Some of these variables serve as fine-tuning parameters that experimenters can use to tweak their analysis, while others serve to turn various optional KOFIA functions on and off. Only variables in the namelist name can be changed. If name is not provided, the names of all the known namelists are displayed, along with a brief description of their purpose. As of this writing, the following namelists were recognized:

PFLAG	DC information output units
DCPHYS	DC physical constants
HCUTS	DPLOT user cuts
DCXYZPAR	DC TDC data cuts
DCCIRPAR	DC circle fit and Z fit parameters
DCLINPAR	DC line fit parameters
PHOPAR	PHOTO flags
DCCUTS	DC data cuts
DCFIT	DC analysis flags
DSPPAR	Event display parameters
PHYSPAR	Detector physical parameters
BWCUTS	Beam MWPC time cuts
BWTIME	Beam MWPC time values
KPUNIT	KOFIA print units
KAHOST	ACP KOFIA Host flags
KANODE	ACP KOFIA Node flags
TDPI	TD PULSE INTEGER parameters
TDPR	TD PULSE REAL parameters
INTC	INTIME cuts
KFLAGS	KOFIA execution control flags
REG	KOFIA_UTILITY parameters
IWFIT	IWC fit flags

To view the variables (and their values) in each of these namelists, use the command SHOW NAMELIST name from the KOFIA prompt, or type "?" from within the namelist editor.

To change the value of a variable from within the namelist editor, type:

```
VARIABLE=new_value[,VARIABLE=new_value[,...]]  <CR>
```

Keep entering such changes until you are done with the namelist, and then type an ampersand ("`&`") to exit the editor.

WARNING: the namelist editor command "?" does not work properly on VMS computers. This bug seems to be the fault of VAX FORTRAN, and it is uncertain whether or when it will be fixed.

Example:

```
KOFIA> NAME PHOPAR
Enter PHOTO parameters:
IPIC=6
DURATION=2.0
&
```

NOTRAP

Syntax: `NOTRap`

Disable the error trapper. All subsequent system errors encountered by `KOFIA` will result in a stack dump, and possibly termination of the session. This command may have to be used when debugging `KOFIA` routines.

PHOTO

Syntax: `PHOTo`

Produce a graphical display of the current event and enter PHOTO command mode (*cf.* the Photo subpackage, below).

PLTR

Syntax: `PLTR plotter_type`

Define the plotter type to be used for outputting hardcopies of photos. The default plotter type is taken from the logical name `TRIUMF_PLOTTER_TYPE`, or is set to "`HP`" if this logical is also undefined. The plotters that `KOFIA` knows about are `HP`, `PSCRIPT` (PostScript), `QMS`, and `ZETA`.

PRINT

Syntax: `PRint[/OUT=filename] [specification]`

Display histograms, scatterplots and their respective indices. Output is ASCII, and is sent to the screen and log file. If the `OUT` qualifier is used, the output is also written to the given file (or logical unit number). In the absence of a `specification`, all indices, scatterplots, and histograms that were set up in the `DEFINE` routine are printed.

A `specification` consists of the letter "`H`" followed by the numbers of the desired histograms, the letter "`S`" followed by the numbers of the desired scatterplots, or the letter "`I`" followed by the desired indices. Combinations of these letters, and lists and ranges of numbers can also be specified.

Examples:

`PRINT S 5-8` — print scatterplots 5 through 8

`PRINT/OUT=HIST.TXT H,S 1,2` — print histograms and scatterplots 1 & 2 to the file `HIST.TXT`

QUIT

Syntax: `QUit`

Terminate the `KOFIA` session.

RELEASE

Syntax: `RELease`

Close the current `KOFIALOG` file, and start a new one. The new one will have the same name, but a new version number. WARNING: on operating systems without file version numbers (*eg.* Unix) , this command will overwrite the first log file.

RESTORE

Syntax: `REStore [/YBOS] file_name`

Restore the specified FIOWA save file. The restored histograms and scatterplots can then be examined and information about them extracted without having to redo the analysis which produced them in the first place. User routines can be used to process these plots, and additional data from other analysis jobs can be added to them. If the `YBOS` qualifier is used, `KOFIA` will assume that the histograms are saved in machine-independent YBOS format (*cf.* `SAVE/YBOS` command).

REWIND

Syntax: `REWind [keyword]`

Reposition the input stream (*ie.* `MTIN`) to the load point by default. The following optional `keyword`s can be used to control where the input is rewound to:

`RUN`	rewind to beginning of run
`FILE`	rewind to beginning of file
`TAPE`	rewind to beginning of tape or disk file

WARNING: avoid using `REWIND FILE` and `REWIND RUN` on disk-based files.

RUNS

Syntax: `RUNs[/NOSKIP] [run1[-run2][, ...]]]`

Select only the given runs for analysis on subsequent `ANALYZE` commands. By default, `KOFIA` searches for `BEGIN_RUN` records to find the requested runs. If the `NOSKIP` qualifier is used, the run number of each event is checked instead, which is slower, but can be useful if the `BEGIN_RUN` records have been skimmed out of the data file. The `RUNS` command with no arguments resets `KOFIA` to process all encountered runs.

Examples:	`RUN 102-106,110`	select runs 102 to 106 and run 110 for analysis
	`RUN`	select all runs for analysis

SAVE

Syntax: `SAVE [/HBOOK] [/LOGICAL] [/YBOS]`

Save FIOWA histograms and scatterplots to a file. This file can be `RESTORE`d later, or in another analysis, or can be used as input to the REPLAY program. By default, `KOFIA` composes the save file name from a string entered by the user and the current run number, with the file extension "`.HIS`". This file is stored in the directory referenced by the logical name `KHIST`. If the `LOGICAL` qualifier is used, however, the histograms are saved to the file name given by the logical `KOFIA$HIS`. The `YBOS` qualifier saves the histograms in machine-independent YBOS format. This is useful if the histograms need to be transferred between different computer architectures. Users who are using HBOOK to build histograms can save using the `HBOOK` qualifier.

SET

Syntax: `SET flag[/option] [=] ON|OFF`

The specified analysis `flag` is turned `ON` or `OFF`, as requested. Accepted `flags` and `options` are:

`ALL`	all sub-system decode and analyze flags
`BPRINT`	???
`DECODE`	all sub-system decode flags
`PHOTO`	graphical event display for each analyzed event
`SKIM`	single-stream skimming to `MTOUT` (only turned on at the beginning of a run)
`TRACK`	all sub-system analyze flags

	if turned on, all decode flags are also turned on
`ss[/ANALYZE]`	a subsystem analyze flag
`ss/CUTS`	a subsystem cuts flag
`ss/TRACK`	a subsystem tracking flag

`ss` is a two letter sub-system code, for example `BW` or `DC`.

SHOW

Syntax: `SHow keyword [/qualifier] [argument]`

Display the current values associated with the specified `keyword`. Known `keywords` and their accepted qualifiers/arguments are:

- `CDF` calibration files used for this run
- `ERRORS` error messages processed by `KERROR`
- `FLAGS` `ON/OFF` setting of flags set by the `SET` command
- `[IC]FAIL` sub-system and event failure statistics; (same as command `ICFAIL`)
- `KHIST` current histogram save directory
- `LOGICALS` current `KOFIA` logical name definitions
- `MTIN` input device
- `MTOUT` output device
- `NAMELIST [name]` namelist descriptions
- `RUNS` runs that have been selected for analysis
- `SCALERS [/HARD]` scaler information (`HARD` qualifier sends output to a text file)
- `SUBSYSTEMS [/ORDER]` list sub-system codes and descriptions; (`ORDER` qualifer sorts the list alphabetically)

SKIM

Syntax: `SKIM [/ON|OFF] [/operation[/operation]] [record_type] [channel(s)] [file]`

Control multi-skimming behaviour. Multi-skimming, unlike single-skimming (which is invoked by the command `SET SKIM ON`), allows users to control several output channels, and select which data records (not just which events) are sorted into each channel. Up to ten output channels may be simultaneously controlled.

The qualifiers `ON` and `OFF` start and stop multi-skimming. Every other `SKIM` command describes a certain set of `operation`s which should be performed on the given `channel`s or `record_type`s. The permitted `operation`s are:

- `ASSIGN` assign a channel number to a file
- `CLOSE` close the given channel(s)
- `OPEN` open the channel(s) when multi-skimming is turned on
- `REOPEN` same as `OPEN`, except that the device is positioned at LEOD when opened
- `SHOW` display the current skim devices
- `SKIP` do not output the given record type to the given channel(s)
- `WHAT` show current record type output assignments
- `WRITE` output the given record type to the given channel(s)

The default operation is `ASSIGN`. The `ASSIGN` operation may be combined with the `OPEN` and `REOPEN` operations. The known `record_type`s are:

`BEGIN_FILE`	`END_RUN`
`BEGIN_RUN`	`END_SPILL`
`CMC_STATUS`	`EVENT`

COMMENT	FILE
END_FILE	UNKNOWN

By default, all **record_type**s except **EVENT** are selected.

The skim **channel** is an integer from 1 to 10. Comma-seperated lists of **channel**s and ranges of **channel**s are also accepted. By default, all channels are selected for the given operation.

Examples:	`SKIM/OPEN 1 SCR0:[JOEUSER]SKIM_1.DAT`	
		assigns skim channel 1 to the given file, and opens the file when skimming is started
	`SKIM/SKIP END_SPILL`	do not write `END_SPILL` records to any channels
	`SKIM/WRITE EVENT 1,2`	write `EVENT` records to channels 1 and 2 only
	`SKIM/ON`	starts multi-skimming

SKIP

Syntax: `SKIp[/qualifier] n`

Skip over the next `n` files or events, without analyzing them. Accepted qualifiers are `FILES` and `EVENTS` (the default). The number of events to skip must be positive, but files may be backwards-skipped by specifiying a negative number.

Examples:	`SKIP/FILES -1`	skip backwards one file
	`SKIP 15`	skip the next 15 events

SUM

Syntax: `SUM`

Sum histograms and scatterplots. This command is only available on ACP implementations of `KOFIA`.

TERMTYPE

Syntax: `TERMtype terminal_type`

Define the terminal type to be used for displaying graphics. The default is taken from the logical name `TRIUMF_TERMINAL_TYPE`, or is set to `VT640` if this logical is undefined. Known terminal types are: `VT100`, `VT640`, `CIT467`, `PT100G`, `TK4107`. `TK4010`, `VS11`, `VT241`, `GR1105`, `CVAXSTA` (colour VAXstation running UIS), `MVAXSTA` (monochrome VAXstation running UIS), and `DECWIND` (any DECwindows or X-windows device). Some of these terminal types are not capable of displaying graphics (*eg.* `VT100`), but will still enable users to enter the photo package and obtain hardcopies.

TRAP

Syntax: `TRAp`

Enable the error trapper. All subsequent system errors and interrupts (including CONTROL-C typed at the keyboard) encountered by `KOFIA` will break the analysis and return the user to the `KOFIA` command line. This feature is enabled automatically when `KOFIA` starts up. It can be disabled with the `NOTRAP` command. The error trapper is only available on VMS machines.

UNMAP

Syntax: `UNMap`

Unmap the FIOWA common blocks. See the `MAP` command for more.

USE

Syntax: `USE component attribute [filename]`

Use the specified calibration file in place of the standard one given by CFM, for the calibration type given by `component` and `attribute`. If `filename` is not given, any previous `USE` command which affected the given calibration type is cleared.

WARNING: `USE` definitions can only be cleared if the executable is linked to CFM version 2.0 or higher.

Examples:	`USE DC PED SCRO:[JOEUSER]DC_PED.99999`	
		use the given file instead of the `DC_PED` file in the CFM databases
	`USE DC PED`	use the normal CFM file for `DC_PED`

WIDTH

Syntax: `WIDth n`

Change screen width to `n` columns, where `n` can be up to 132. This can be useful for wide-format data displays, including the printing of histograms.

ZERO

Syntax: `ZEro`

Clear the run table, event counters, statistics, and all histogram and scatterplot bins.

1.5 The Photo subpackage

The Photo subpackage is an interactive environment that can be reached from `KOFIA`, if the `PHOTO` library has been linked in. To view photos, the user must request them explicitly, using the command:

```
KOFIA> PHOTO
```

or implicitly (*ie.* automatically on each event), with the command:

```
KOFIA> SET PHOTO ON
```

It is also possible for Photo to be invoked from within a user's own routines, with the call:

```
CALL PHOTO
```

Photo can draw up to twelve different views of the E787 detector, although as of this writing, only 7 of these views contained useful information. The various picture numbers are:

1. **DC X-Y:** shows the end view of the drift chamber (looking upstream) with the hit points and fitted track(s) projected onto the X-Y plane at $Z = 0$.
2. **DC Z:** shows the hit points and fitted track(s) of the drift chamber projected onto a "radius versus Z-direction" plane.
3. **DC X-Z and Y-Z:** shows the X-Z and Y-Z projections of the hit points and fitted track(s) in the drift chamber at $Y = 0$ and $X = 0$, respectively.
4. **General Event Display:** shows an X-Y end view and R-Z side view of the detector at a scale of 15:1. In the end view, outlines of the target and drift chamber active volumes are shown. In the target, range stack, barrel veto, and end cap, hit cells above a given threshold are outlined. Colour terminals can display a spectrum of energy thresholds in different colours. All drift chamber and IWC hits are displayed (provided that DC tracking has been enabled). The side view shows the Z-information derived from the drift chamber stereo layers.
5. **Time Display:** shows an end view of the detector, similar to that shown in picture 4, except that illuminated detector elements and colours are determined by the times of their hits, rather than the energy deposited in them. A colour legend is displayed on the right hand side of the screen.
6. **IWC Exploded View:** shows an exploded view of the inner wire chamber. The chamber is viewed from the inside, and is opened up from a cylinder to a rectangle, by slitting it along the intersection of the chamber with the X-axis. Upstream is at the bottom of the display. Hit anode wires appear as vertical lines, while hit cathode strips appear as diagonal lines.

7-11. Unused.

12. **Full Detector:** shows the full E787 detector, with all detector elements illuminated. This display does not contain any event-specific information.

The default picture number is 4. This may be changed by altering the value of the variable `IPIC` in the namelist `PHOPAR`.

Photo can be configured to automatically cycle through the events it draws, pausing for a certain amount of time before moving on to the next one. To set up this feature, increase the value of the variable `DURATION` in the namelist `PHOPAR` to be equal to the number of seconds to pause between photos. If `DURATION` is set less than or equal to zero, Photo will stop after each picture and enter its interactive command line mode.

Photo has a simple command-line interface that allows users to select the pictures they want to view, change the scale at which the pictures are drawn, add more information to the pictures, and modify various `KOFIA` namelists. The prompt in Photo changes to "`Kofia/PHOTO>`" to remind users that `KOFIA` is now accepting a different set of commands.

Note that before pictures can be viewed on a terminal, the user will have to define his or her terminal type appropriately. See the command `TERMTYPE` for more.

ACLEAR

Syntax: `AClear`

Clear the screen of text, and place the prompt at the bottom of the screen.

BROWSE

Syntax: `BROWSE [/RS | | /TG | /TT]`

Displays the selected subsystem, and allows the user to select hit components using the terminal crosshairs or mouse. The signals in the selected components will be plotted in windows on the right side of the screen. Upstream channels are selected by default, while downstream channels are selected by pressing the "D" key. Press "H" to get hardcopy, "T" to enter the TDPHOTO package (using the selected channels), and "Q" to quit.

CAPTION

Syntax: `CAPtion`

Add a text string to the photo. The user will be prompted for a text string, and then will be asked to select a screen location to write the text to.

CCD

Syntax: `CCD`

Invokes the CCDPHOTO subpackage for producing quality graphs of CCD signals. See the section on CCDPHOTO and TDPHOTO for more details.

CLEAR

Syntax: `CLear`

Clear the screen of graphics and text.

CROSS

Syntax: `CRoss`

Invokes crosshair mode. Positioning the crosshairs and pressing a key or mouse button returns the physical detector coordinates of the selected position. On X-windows devices, clicking and dragging with the mouse will open a zoom window, displaying the selected portion of the photo.

DCXY

Syntax: `DCXY`

Draw picture 1.

DCXZ, DCYZ

Syntax: `DCXZ` or `DCYZ`

Draw picture 3.

DCZ

Syntax: `DCZ`

Draw picture 2.

DISPLAY

Syntax: `DIsplay n`

Draw picture number `n`. See above for descriptions of picture numbers.

DUMP

Syntax: `DUmp[/format][/I2][/LIST] [bank_name[[,]bank_name]...]`

Same as `KOFIA` command of the same name.

EXPAND

Syntax: `EXpand [C|n]`

Expand the time scale in picture number 5. If the argument "`C`" is given, the new time scale can be selected with the graphics cursor. Otherwise, the parameter `n` must ge given.

FACTOR

Syntax: `FActor y_factor`

Redraw the current picture, after rescaling the screen size in the Y-direction by `y_factor`. The changed screen dimensions are retained for subsequent pictures, until another `FACTOR` command is given. This command can be used to correct rectangular aspect ratios that are seen on some terminals. For example, the command "`FACTOR 1.2`" will correct the aspect ratio on many VT640 terminals.

FIT

Syntax: `FIT [itrack]`

Interactively modify what points make up the specified track in the drift chamber. If no argument is specified, then the current track is used. The track number, number of points, chi value and radius of the track are displayed at the top of the screen.

A list of the original point numbers is displayed on the screen and the new list is initially set equal to these points. The user is prompted for the point numbers which will make up the new fit. An input value of 0 at the beginning of an input line will cause the new list to be emptied. Positive (unsigned) input values are added to the list. Negative input values are deleted from the list. Input values are integers separated by commas. An empty carriage return causes the original "best" final fit information to be replaced with the new final fit information and an attempt to fit with these new points is made. If the fit is successful, the new track is drawn; if not, the original track is redrawn.

FULL

Syntax: `FUll`

Redraw the current picture. This command comes in handy when plots have been overwritten, improperly edited, or partially erased.

HARD

Syntax: `HArd`

Produce a hardcopy file of the current picture. The file will be named "`PHOTO.HRD`" for Printronix printers, or "`HPLASER.PLT`" for HP Laserjet printers.

HELP

Syntax: `HELP [command]`

Invoke the VMS Help facility for online help on Photo commands.

INFO

Syntax: `INfo [analysis [analysis [...]]]`

Display additional information on the picture, which reflects the `analysis` that has been done on the event. If `analysis` isn't specified, "buttons" appear at the top of the picture that can be selected with the graphics cursor to chosse from the various options available. The following `analysis` options can be selected: `INTIME`, `RSTRAK`, `TGDCVT`, `TGP`, `TGK`, and `TRKTIM`. Information will only be added to the plot if the user has already performed the given analysis on the event.

Because `PHOTO` is normally called before the user's `DPLOT` routine, much if not all of this extra analysis may not be complete. Set the variable `PHOTO_FIRST` in the namelist `NLKFLAGS` to `.FALSE.` in order to have `DPLOT` called before `PHOTO`. Then the `INFO` command can be used to draw any information dug up by the `DPLOT` routine.

LINE

Syntax: `LINe`

Draw a line on the picture. The user will be prompted to select the starting and ending points of the line with the graphics cursor. This command can be used to perform some rudimentary editing of a picture (such as adding labels and pointers) before printing it.

LIST

Syntax: `LIst [/[NO]ECHO] [/HARD] [/ss [/data] ... ]  [itrack]`

Same as the `KOFIA` command of the same name.

MORE

Syntax: `MOre`

Adds information to the current picture. Values will be written on each above-threshold illuminated detector element on the first call, and on all hit elements on the second call. The values will be energies in picture 4, and times in picture 5. The `MORE` command has no meaning for other pictures.

NAMELIST

Syntax: `NAMElist [name]`

Same as the `KOFIA` command of the same name.

NEXT

Syntax: `Next`

Move on to the next event, returning to the default picture. If there are no more events to be drawn, the user is returned to the `KOFIA` command line.

NOPHOTO

Syntax: `NOphoto`

Exit the Photo subpackage. The graphics screen is cleared, and the `PHOTO` flag is turned `OFF`. Any remaining events are analyzed without displaying photos.

OFF

Syntax: `OFf`

Same as `NOPHOTO` command.

PAGE

Syntax: `PAGE`

Toggle automatic screen clearing on and off. After each picture is drawn, the text screen is cleared by scrolling all text off the top of the screen. On workstation terminal types, the text screen is not cleared, since text and graphics appear in different windows. The `PAGE` command can be used to change these default settings.

PLOT

Syntax: `PLOT`

Generate plotter output. The format of the plotter commands depends on the type of plotter that has been selected using the `PLTR` command or the logical `TRIUMF_PLOTTER_TYPE`. For some plotters, a menu of choices will be displayed on the screen, that allows the user to start the plot immediately or save the commands in a disk file.

PLTR

Syntax: `PLTR plotter_type`

Same as `KOFIA` command of the same name.

PRTR

Syntax: `PRTR printer_type`

Define the printer type to be used for printing photos. The following printer types are recognized: `PX`, `HPLASER`, `HPTHINK`, `LA100`, `HPPAINT`. The default is `HPLASER`.

TD

Syntax: `TD`

Invokes the TDPHOTO subpackage, for generating quality plots of transient digitizer signals. See the section "CCDPHOTO and TDPHOTO" for details.

TERMTYPE

Syntax: `TERMtype terminal`

Same as the `KOFIA` command of the same name.

TIME

Syntax: `TIME [n]`

Draw picture 5. If the argument `n` is provided, the display will be expanded (*cf.* `EXPAND`) by 10^n before being plotted.

ZOOM

Syntax: `Zoom [x y]|[C] [X factor]|[scale]|[C]`

Zoom or pan in pictures 4 and 5. The first argument(s) define the new center of the picture, by supplying the (real detector) coordinates (`x` and `y`) or by marking the desired center with the graphics cursor. The argument `factor` magnifies the picture by the given amount (and must be preceeded by the letter "`X`"). The argument `scale` changes the magnification to be equal to `1:scale`. If "`C`" is provided as the last argument, then the graphics cursor can be used to select the new edge of the screen, implicitly determining the new scale.

Examples:	`ZOOM 0. 0. 1.`	display the center of the detector at 1:1 scale
	`ZOOM C C`	interactively choose the new center and edge of the picture

1.6 CCDPHOTO and TDPHOTO

The CCDPHOTO and TDPHOTO subpackages comprise an interactive environment for generating GPLOT-type graphs of CCD and TD signals. They include functions to alter axis-ranges, point styles, the number of plots per page, and to scan and display comprehensive listings of the data.

These packages may be started up from within the Photo package, by using one of the commands:

`Kofia/PHOTO> CCD/PHOTO` or,
`Kofia/PHOTO> TD`

They can also be started up from within a user's own routines with one of the calls:

`CALL CCDPHO` or,
`CALL TDPHO`

Once begun, two default plots are displayed. The first two hit channels are shown in CCDPHOTO, and the stopping counters (both ends) are shown in TDPHOTO. The user is then presented with the prompt "`Kofia/Photo/CCDPHO>`" or "`Kofia/Photo/TDPHO>`", and can enter any of the following commands:

CLEAR

Syntax: `CLEar`

Clear the graphics screen.

COUNTS

Syntax: `COUnts [mode]`

Alter Y-axis ranges. `Mode` can be one of "`FIXED`" or "`AUTO`". `FIXED` mode prompts the user for the minimum and maximum axis values, while `AUTO` mode autoscales the axis to make the largest pulse in the given time range nearly fill the available space.

DISPLAY

Syntax: `DISplay`

Draw the selected channels.

DRAW

Syntax: `DRAw n`

Draw the first `n` of the selected channels. The maximum number of channels that can be plotted at once is 12. The number of channels requested will be rounded up to the nearest number that will fit evenly on the screen (*ie.* 1, 2, 4, 6, 9, or 12).

EDGE

Syntax: `EDGe`

Toggle the display of leading edge times. This feature is normally shut off, but when turned on, the leading edge time, peak pulse height, pedestal, and half-max pulse height will all be marked with line segments. This command is only available in TDPHOTO.

HARD

Syntax: `HARd`

Generate a printout of the current display.

HELP

Syntax: `HELp`

Show a brief summary of the accepted commands.

LIST

Syntax: `LISt [n]`

List all the hit channels. In TDPHOTO, the map labels of the hit channels are also listed, while in CCD-PHOTO, a summary of the first pulse in each hit channel is also given. The optional argument `n` (only used in CCDPHOTO) will produce a more comprehensive listing of the data in channel `n`.

NEXT

Syntax: `NEXt`

Exit CCDPHOTO or TDPHOTO. "EXIT" and "QUIT" also work.

PLOT

Syntax: `PLOt`

Generate a hardcopy plot of the picture.

RAW

Syntax: `RAW`

Display a comprehensive listing of TD data for all selected channels. This command is only available in TDPHOTO.

SAVE

Syntax: `SAVE`

Saves the current plot settings and channels as the default for subsequent calls to TDPHOTO. Subsequent SAVE commands toggle this feature off and on.

SET

Syntax: `SET`

Select a series of channels for drawing. The user will be prompted to enter a list of comma-separated channel numbers. No more than 12 channels will be accepted.

STYLE

Syntax: `STYle P|L`

Draw the signals as a series of points (if argument is "`P`") or as a series of connected line segments (if argument is "`L`").

SUMMARY

Syntax: `SUMmary`

List data for the selected channels only.

TIMES

Syntax: `TIMes [mode]`

Alter Y-axis ranges. `Mode` can be one of "`FIXED`", "`PROMPT`", or "`AUTO`". `FIXED` mode prompts the user for the minimum and maximum axis values, while `AUTO` mode autoscales the axis to include all pulses in the channel. "`PROMPT`" mode restores the default prompt time window (only in TDPHOTO).

1.7 Skimming

Skimming is the process of selectively "editing" data files to contain only the data that the user is interested in. Depending on what kind of skimming is used, the user can control what events, and even what data records are deleted and which ones are retained. Note that the orginal data file (*ie.* MTIN) is never altered; all changes are written to seperate files.

"Single-skimming" is the easiest skimming mode to use. It has only one output stream, and the user only selects which events are kept and which ones are thrown out. Before the user can single-skim, he or she must define the output device or file (MTOUT), and turn on the SKIM flag (use the command "SET SKIM ON"). Skimming will start after KOFIA encounters the next BEGIN_RUN record (in other words, you can't start skimming in the middle of a run). All "good" events are written to the output. By default, *all* events are good, unless the the variable ICFAIL (INTEGER*4) in the common block /CFAIL/ is set to a non-zero value at some point in the analysis. This may be done by KOFIA, or the user may do it from within his or her own analysis code.

In addition, the user can control which YBOS banks of those in the good events are retained. All banks in the YBOS 'A' list are normally written (use the KOFIA command "DUMP/LIST" to see what banks are normally in this list). By deleting banks from this list, or adding his or her own banks to this list, the user can customize the data that is skimmed. Consult the YBOS manual for details on these operations.

"Multi-skimming" is a more complex skimming mode that permits much more flexibility and power in the user's control of the skimming process. It works in a similar fashion to single-skimming, but the user can control up to 10 independent output streams, and send different events and data record types to each, or throw out certain record types altogether. The user must use the "SKIM" command (*cf.*) to set up the various output channels and determine which record types should be written to each. Once skimming is turned on (using the "SKIM/ON" command), the data read from the input tape is sorted and written to the various output channels as the user has requested.

1.8 Writing Your own KOFIA Code

This section outlines some of the guidelines that should be followed when users are writing their own analysis routines. These guidelines exist for several reasons:

- to improve code portability between machine architectures
- to ensure compatibility with the various KOFIA packages and libraries
- to make your routine easier for others to understand

Code that does not follow these guidelines will not be accepted for inclusion into official KOFIA libraries.

The EXPAND Preprocessor

Write all routines in the EXPAND preprocessor format (with the ".CDF" file extension instead of the normal ".FOR" extension). Consult the EXPAND documentation for details on using EXPAND. The following EXPAND statements should be used in place of their FORTRAN counterparts (EXPAND commands always begin in column 1):

- $$IMPLICIT (instead of IMPLICIT NONE)
- $$INCLUDE

Always include the line:

```
$$INCLUDE 'KOFIA$LIB:KALIAS.INC'
```

as the first line of any KOFIA routine.

To write machine-dependent code, use the following EXPAND options in the order given:

ACP_NODE: code for 68020-based ACP nodes.

BEND: code for big-endian computers (including 68020-based ACP nodes, IBMs, and Silicon Graphics Iris computers).

DECS3100: code specifically for Digital RISC/Ultrix computers.

IRIS: code specifically for Silicon Graphics Iris computers.

UNIX: file-handling code for UNIX-based systems (including DECstations and Iris computers), and code to deal with MIPS compiler quirks.

IBM: code specifically for IBM computers.

`VAX`: the default environment, used for VMS-specific code, or for non-file handling code on MIPS-based computers such as DECstations and Irises.

The above options should be used with the EXPAND `$$IF`, `$$ELSE IF`, `$$ELSE`, and `$$ENDIF` commands to surround code for the various architectures that the code is meant to be run on. Consult the EXPAND documentation for more details on how EXPAND processes these options.

Statement Ordering

Specification and Data statements must obey the Fortran 77 conventions for their ordering. They must appear in the following order:

1) IMPLICIT NONE Statement
2) Type Statements (REAL, INTEGER, etc.)
3) PARAMETER Statements
4) DIMENSION Statements
5) COMMON Statements
6) EQUIVALENCE Statements
7) SAVE Statement
8) DATA Statements

Variable Declarations

Always use `$$IMPLICIT` and declare all variables. Do not limit yourself to 6-character variable names, as most modern versions of FORTRAN can handle symbols of 31 or more characters, and the EXPAND preprocessor can deal with those that cannot. If you must initialize variables, use `DATA` statements, and do not initialize them within the variable declaration statement. Use the `SAVE` statement in your routines, and place it before all `DATA` statements.

Common Blocks and Include Files

All `KOFIA` common blocks, as well as parameter definitions that are shared by more than one file, are placed in include files which are found in the directory `KOFIA$LIB`. To use these structures, include the files using the EXPAND statement "`$$INCLUDE 'filename'`". Avoid using nested include statements, unless the include files are also in EXPAND format.

When creating your own common blocks, place them is similar files, and make sure that the variables are declared before the `COMMON` statement. Do not use `DATA` statements to initialize common block variables (use a `BLOCK DATA` subprogram instead). Do not allow common block variables to cross 32-bit word boundaries.

Logical Units

Do not use logical units indiscriminately! `KOFIA` uses many logical units for its own internal purposes, and will get very upset if users steal them away. If you are just opening a file temporarily (for example, with calibration files) always use the logical unit `LCDF`, defined in the include file `KOFIA$LIB:LUNS.CMN`. The following logical units are sacred, and should never be used explicitly:

1-15	`KOFIA` I/O
20-22	old TD fitter
39	TD fiducial bifurcation file
41-45	CFM databases and lock files
50-59	KOFIA command files
80-84	new TD fitter

Comments

Adequately comment your code! Do not over-comment, since this can make code harder to read; make your code self-documenting by choosing self-explanatory variable names. Do not comment blank lines, since it is not necessary in modern versions of FORTRAN, and only adds visual clutter to the listing. Include header comments in all routines that descibe the use of the routine, and its various arguments. Also comment all include files, describing where they are used, and the function of the variables contained within it. Always add a short comment to any official `KOFIA` routine that you have modified, explaining the changes.

Other Stylistic Suggestions

You can improve the legibility of your code by using both upper and lower case, since newer versions of FORTRAN are case-insensitive. For example, you can make all FORTRAN keywords UPPER CASE and all symbols lower case. Always use indentation to make DO and IF blocks more readable. Use both horizontal and vertical spacing to improve legibility and delineate different sections of a line or routine.