Genie IO Basics

Genie maintains an internal record of the current INPUT and OUTPUT files. Commands such as GET, READ, PUT and WRITE will use these default values when no FILE parameter is given. In the following paragraphs, **bold text** represents commands that can be typed in Genie, and *italics* represents output produced by Open Genie.

Setting The Default Output Device

The default output device is manipulated by using the **SET/OUTPUT** command e.g.

>> SET/OUTPUT "myfile.out"

Default Output: myfile.out

Setting The Default Input Device

This can be set in two ways. One is to use the **SET/INPUT** command e.g.

>> SET/INPUT "hrp08585.raw"

Default input: hrp08585.raw

The other way is to use the ASSIGN command, which takes a RUN NUMBER as its argument and constructs a filename from this e.g.

>> ASSIGN 8585

Default input: /vax/hrpd0/hrpmgr/data/hrp08585.raw

The values used by **ASSIGN** to construct a filename are controlled by the **SET/DISK**, **SET/DIR**, **SET/INST** and **SET/EXT** commands and can be listed with the **SHOW/DEFAULTS** command. If a user program wishes to convert a RUN NUMBER into a String representing the file, the **CFN()** function can be used.

Reading Files

Open Genie provides the generic **GET** function for reading data. This function takes an integer (the spectrum or block number) or string (the item label) argument and returns the variable read. The **GET** function is capable of reading data from Open Genie intermediate files, Genie II intermediate files and ISIS Raw data files - it uses the **FILETYPE** command to determine the appropriate data reading routine to call.

>> PRINTN FILETYPE("hrp08585.raw")

raw

For ASCII data in column format, use the **READ/ASCII/COLUMNS** command. The following will read in three columns of data into a workspace W, creating three array fields called "X", "Y" and "E"

>> W = READ:ASCII:COLUMNS("test.dat", "x,y,e")

Blank lines and those beginning with a "#" are ignored by the reading routine, so files may contain comments.

An alternative ASCII format that can read in any type of genie variable is available via the **READ:ASCII:STREAM** command. This format is provided so that programs external to Open Genie can write a file that may later be loaded in; it is a little complicated and may not contain comment characters (yet). To get an idea of what an input file should look like, use the **WRITE/ASCII/STREAM** command to WRITE out a variable.

Assuming W1 and W2 are Genie variables

>> WRITE/ASCII/STREAM "test.dat" W1

>> U = READ:ASCII:STREAM("test.dat")

If more than one variable is to be READ/WRITTEN from the file, the procedure is a little more complicated:

>> H = OPEN:STREAM:NEW("test.dat") # create stream for new file

>> WRITE/ASCII/STREAM H W1 # add variable to stream

>> WRITE/ASCII/STREAM H W2

>> CLOSE/STREAM H # close stream

>> H = OPEN:STREAM:READ("test.dat") # re-open stream for reading

>> R1 = READ:ASCII:STREAM(H) # read in first variable (W1)

>> R2 = READ:ASCII:STREAM(H) # read in second variable (W2)

>> CLOSE/STREAM H # all done, close stream

The **WRITE/ASCII/STREAM** command is analogous to the Genie II SHOW/DATA command.

Writing Files

Open Genie provides a powerful platform independent intermediate file based on the XDR[1] format. As well as being able to store any type of Genie variable (including recursive structures, such as Workspaces containing Workspaces), variables can be assigned symbolic tags and comment string. The intermediate file also records the creation date and name of the author of any block added (see the **DIR** command).

Example session

>> SET/INPUT "hrp008585.raw"

Default input: hrp008585.raw

>> W = GET(3) # read spectrum 3

GET: hrp08585.raw is of type raw

GET: Reading spectrum 3 from hrp08585.raw

>> U = GET(4)

1 External Data Representation, RFC 1014

GET: hrp08585.raw is of type raw

GET: Reading spectrum 4 from hrp08585.raw

>> SET/OUTPUT "myfile.int"

Default output: myfile.int

>> PUT/NEW W "W" "Workspace W" # create a new file, or overwrite existing one

PUT: Writing variable to "myfile.int" in Open Genie format

PUT: Label = "w", Comment = "Workspace W"

1 blocks now in myfile.int

>> PUT U "U" "Workspace U" # append U to the file

PUT: Writing variable to "myfile.int" in Open Genie format

PUT: Label = "u", Comment = "Workspace U"

2 blocks now in myfile.int

>> DIR/OUT/FULL

Block = 1, Label = "w", Type = GXWorkspace

Created: Mon 18 Mar at 22:42:00 WET by System PRIVILEGED Account <root@isds9.nd.rl.ac.uk>

Comment: Workspace W

Block = 2, Label = "u", Type = GXWorkspace

Created: Mon 18 Mar at 22:42:48 WET by System PRIVILEGED Account <root@isds9.nd.rl.ac.uk>

Comment: Workspace U

>> T = GET("U", "myfile.int") **#** read block labelled "u"

GET: myfile.int is of type G3 intermediate

GET: Reading block "u" from Open Genie intermediate file myfile.int

2 blocks in Open Genie intermediate file myfile.int

Array may be written out in ASCII stream format (see the **WRITE/ASCII/STREAM** example above), or in column format. If W is a workspace obtained from an ISIS raw file, then the following code will write an "X Y E" file:

>> WRITE/ASCII/COLUMNS "test.dat" W.X W.Y W.E

To add the comment "My test data" to the head of the created file, use:

>> WRITE/ASCII/COLUMNS "test.dat" "My test data" W.X W.Y W.E