tively small FIFO. Note that if you change the size of the disk cache you may have to reboot for the change to take effect.

When the stop command is given to a running FIFO then it goes into review mode and remains valid. If the FIFO is attached to a file then the entire contents of the file can be reviewed or be transferred to a wave using the FIFO2Wave command.

The act of attaching a FIFO to an existing file for review using the rfile keyword of the CtrlFIFO command reads in the file contents and sets itself up for review. You should not use the NewFIFOChan command or any of the other CtrlFIFO keywords except size. Here is all that is required to review a preexisting file:

```
Variable refnum
Open/R/P=mypath refnum as "my file"
NewFIFO dave
CtrlFIFO dave,rfile=refnum
```

If any chart controls have been set up to monitor FIFO dave then they will automatically configure themselves to display all the channels of dave using default parameters.

The connection between FIFOs and chart controls relies on Igor's dependency manager. The dependency manager does not automatically run during function execution — you have to explicitly call it by executing the DoUpdate command.

The dependency manager sends messages to a chart control when:

- A FIFO is created
- A FIFO is killed
- A FIFO becomes valid (start command)
- Data is added to a FIFO

In particular, if inside a user function, you kill a FIFO and then create it again you should call DoUpdate after the kill so that the chart control notices the kill and can get ready for the creation.

FIFO File Format

This section is provided for programmers who want to create FIFO files as input for Igor with their own programs or to analyze data stored in Igor-generated FIFO files. Such applications require familiarity with C programming.

The FIFO file uses C structures named FIFOFileHeader, ChartChunkInfo, ChartChanInfo, and FIFOSplitFileHeader. These are defined in the file NamedFIFO.h which is shipped in "Igor Pro Folder\Miscellaneous\More Documentation". NamedFIFO.h is the primary documentation for the FIFO file format.

The FIFOFileHeader, ChartChunkInfo, and FIFOSplitFileHeader structures contain version fields. These structures may evolve in the future, so your code should check them and fail gracefully if you encounter an unexpected value.

Igor supports integrated FIFO files, in which the header and data are in the same file, and split FIFO files, in which the header and data are in separate files.

An integrated FIFO file consists of a FIFOFileHeader structure followed by a ChartChunkInfo structure and finally by chunks of data until the end of the file. A user-defined function that can parse an integrated FIFO file can be found in the "FIFO File Parse" example experiment.

The split format uses the FIFOSplitFileHeader structure to allow the raw data to reside in its own file rather than having to be in the same file as the header. This facilitates the use of Igor to review large binary files generated by third-party programs.

See the "Wave Review Chart Demo" example experiment for sample code for both the integrated and split FIFO file formats.