Reading real waves works like reading real variables except that a real wave has multiple elements each of which is 1, 2, 4, or 8 bytes depending on the wave's data type. For each element of a real wave, FBinRead reads the number of bytes implied by /F or by the wave's native data type, converts those bytes to the wave's data type if necessary, and stores the resulting value in the corresponding wave element. When reading into a complex wave, this process is repeated twice, once for the real part of each element and once for the imaginary part.

Reading structures is different. The /F flag has no effect. FBinReads reads the number of bytes required to fill the structure which depends on the sizes of the individual fields and the fact that Igor uses 2-byte structure alignment. After the bytes are read from the file into the structure, FBinRead byte-swaps the individual fields if you include the /B flag.

The FBinRead operation is not multidimensional aware. See **Analysis on Multidimensional Waves** on page II-95 for details.

See Also
FBinWrite, **Open**, **FGetPos**, **FSetPos**, **FStatus**, **GBLoadWave**

FBinWrite

FBinWrite [*flags*] *refNum, objectName*
The FBinWrite operation writes the named object in binary to a file.

Parameters
refNum is a file reference number from the **Open** operation used to open the file.

objectName is the name of a wave, numeric variable, string variable, or structure.

Flags

/B[=*b*]   Specifies file byte ordering.

b=0:	Native (same as no /B).
b=1:	Reversed (same as /B).
b=2:	Big-endian (Motorola).
b=3:	Little-endian (Intel).

/F=*f*   Controls the number of bytes written and how the bytes are formatted.

f=0:	Native binary format of the object (default).
f=1:	Signed byte; one byte.
f=2:	Signed 16-bit word; two bytes.
f=3:	Signed 32-bit word; four bytes.
f=4:	32-bit IEEE floating point; four bytes.
f=5:	64-bit IEEE floating point; eight bytes.
f=6:	64-bit integer; eight bytes. Requires Igor Pro 7.00 or later.

/P   Adds an IgorBinPacket to the data. This is used for PPC or Apple event result packets (*refNum* = 0) and is not normally of use when writing to a file.

/U   Integer formats (/F=1, 2, or 3) are unsigned. If /U is omitted, integers are signed.

Details
A zero value of *refNum* is used in conjunction with Program-to-Program Communication (PPC) or Apple events (*Macintosh*) or **ActiveX Automation** (*Windows*). The data that would normally be written to a file is appended to the PPC or Apple event or ActiveX Automation result packet.

If the object is a string variable then /F doesn't apply. The number of bytes written is the number of bytes in the string.

The binary format that FBinWrite uses for numeric variables or waves depends on the /F flag. If no /F flag is present, FBinWrite uses the native binary format of the named object.