If *x1* or *x2* are not within the X range of *XWaveName*, faverageXY limits them to the nearest X range limit of *XWaveName*.

faverageXY returns the area divided by (*x2* -*x1*).

If any values in the X range are NaN, faverageXY returns NaN.

Reversing the order of *x1* and *x2* does not change the sign of the returned value.

The values in *XWaveName* may be increasing or decreasing. faverageXY assumes that the values in *XWaveName* are monotonic. If they are not monotonic, Igor does not complain, but the result is not meaningful. If any X values are NaN, the result is NaN.

The faverageXY function is not multidimensional aware. See Chapter II-6, **Multidimensional Waves** for details on multidimensional waves, particularly **Analysis on Multidimensional Waves** on page II-95.

See Also

Integrate, **area**, **areaXY**, **faverage** and **Poly2D Example 3**

FBinRead

FBinRead [***flags***] ***refNum, objectName***

The FBinRead operation reads binary data from the file specified by *refNum* into the named object.

For simple applications of loading binary data into numeric waves you may find the GBLoadWave operation simpler to implement.

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 read and how the bytes are interpreted.

- *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.

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

Details

If *objectName* is the name of a string variable then /F does not apply. The number of bytes read is the number of bytes in the string *before* the FBinRead operation is called. You can use the **PadString** function to set the size of a string.

The binary format that FBinRead uses for a numeric variable depends on the /F flag. If you omit /F, the native data type of the variable, which is 8-byte double-precision floating point, is used. So, when reading into a real numeric variable, depending on /F, FBinRead reads 1, 2, 4, or 8 bytes from the file, converts those bytes to double-precision floating point if necessary, and stores the resulting value in the variable. When reading into a complex numeric variable, this process is repeated twice, once for the real part and once for the imaginary part.