```
response = FetchURL("http://www.wavemetrics.com")

// Get a binary image file from a web server and then
// save the image to a file on the desktop.
String url = "http://www.wavemetrics.net/images/tbg.gif"
String imageBytes = FetchURL(url)
Variable error = GetRTError(1)
if (error != 0)
   Print "Error downloading image."
else
   Variable refNum
   String localPath = SpecialDirPath("Desktop", 0, 0, 0) + "tbg.gif"
   Open/T=".gif" refNum as localPath
   FBinWrite refNum, imageBytes
   Close refNum
endif
```

See Also

FTPDownload, **URLEncode**, **URLRequest**

Network Communication on page IV-267, **Network Connections From Multiple Threads** on page IV-271.

FFT

`FFT` `[`***`flags`***`]` ***`srcWave`***

The FFT operation computes the Discrete Fourier Transform of *srcWave* using a multidimensional prime factor decomposition algorithm. By default, *srcWave* is overwritten by the FFT.

Output Wave Name

For compatibility with earlier versions of Igor, if you use FFT with no flags or with just the /Z flag, the operation overwrites *srcWave*.

If you use any flag other than /Z, FFT uses default output wave names: W_FFT for a 1D FFT and M_FFT for a multidimensional FFT.

We recommend that you use the /DEST flag to make the output wave explicit and to prevent overwriting *srcWave*.

Flags

/COLS — Computes the 1D FFT of 2D *srcWave* one column at a time, storing the results in the destination wave.

$$I[t_1][n] = \sum_{k=0}^{N-1} f[t_1][k] \exp\left(i2\pi kn / N\right).$$

You must specify a destination wave using the /DEST flag. No other flags are allowed with this flag. The number of rows must be even. If *srcWave* is a real (NxM) wave, the output matrix will be (1+N/2,M) in analogy with 1D FFT. To avoid changes in the number of points you can convert *srcWave* to complex data type. This flag applies only to 2D source waves. See also the /ROWS flag.

/DEST=*destWave* — Specifies the output wave created by the FFT operation.

It is an error to attempt specify the same wave as both *srcWave* and *destWave*.

The default output wave name is W_FFT for a 1D FFT and M_FFT for a multidimensional FFT.

When used in a function, the FFT operation by default creates a complex wave reference for the destination wave. See **Automatic Creation of WAVE References** on page IV-72 for details.

/FREE — Creates *destWave* as a free wave.

/FREE is allowed only in functions and only if *destWave*, as specified by /DEST, is a simple name or wave reference structure field.

See **Free Waves** on page IV-91 for more discussion.

The /FREE flag was added in Igor Pro 7.00.

/HCC — Hypercomplex transform (cosine). Computes the integral

$$I_c(\omega_1,\omega_2) = \int\int_{-\infty}^{\infty} f(t_1,t_2)\cos(t_1\omega_1)\exp(it_2\omega_2)dt_1dt_2$$

using the 2D FFT (see **Details**).

/HCS — Hypercomplex transform (sine). Computes the integral

$$I_s(\omega_1,\omega_2) = \int\int_{-\infty}^{\infty} f(t_1,t_2)\sin(t_1\omega_1)\exp(it_2\omega_2)dt_1dt_2$$

using the 2D FFT (see **Details**).

/MAG — Saves just the magnitude of the FFT in the output wave. See comments under /OUT.

/MAGS — Saves the squared magnitude of the FFT in the output wave. See comments under /OUT.

/OUT=*mode* — Sets the output wave format.

mode=1: Complex output (default)
mode=2: Real output
mode=3: Magnitude
mode=4: Magnitude square
mode=5: Phase
mode=6: Scaled magnitude
mode=7: Scaled magnitude squared

You can also identify modes 2-4 using the convenience flags /REAL, /MAG, and /MAGS. The convenience flags are mutually exclusive and are overridden by the /OUT flag.

The scaled quantities apply to transforms of real valued inputs where the output is normally folded in the first dimension (because of symmetry). The scaling applies a factor of 2 to the squared magnitude of all components except the DC. The scaled transforms should be used whenever Parseval's relation is expected to hold.

/PAD={*dim1* [, *dim2*, *dim3*, *dim4*]}

Converts *srcWave* into a padded wave of dimensions *dim1*, *dim2*…. The padded wave contains the original data at the start of the dimension and adds zero entries to each dimension up to the specified dimension size. The *dim1*… values must be greater than or equal to the corresponding dimension size of *srcWave*. If you need to pad just the lowest dimension(s) you can omit the remaining dimensions; for example, /PAD=*dim1* will set *dim2* and above to match the dimensions in *srcWave*.

/REAL — Saves just the real part of the transform in the output wave. See comments under /OUT.

/ROWS — Calculates the FFT of only the first dimension of a 2D *srcWave*. It thus computes the 1D FFT of one row at a time, storing the results in the destination wave.

$$N[n][t_2] = \sum_{k=0}^{M-1} f[k][t_2] \exp(i2\pi kn / M)$$

You must specify a destination wave using the /DEST flag. No other flags are allowed with this flag. The number of columns must be even. If *srcWave* is a real (NxM) wave, the output matrix will be (N,1+M/2) in analogy with 1D FFT. To avoid changes in the number of points you can convert *srcWave* to complex data type. See also /COLS flag.

/RP=[*startPoint*, *endPoint*]

/RX=(*startX*, *endX*) — Defines a segment of a 1D *srcWave* that will be transformed. By default the operation transforms the whole wave. It is sometimes useful to take advantage of this feature in order to transform just the defined interval, which includes both end points. You can define the interval using wave point indexing with the /RP flag or using the X-values with the /RX flag. The interval must include at least four data points and the total number of points must be an even number.

/WINF=*windowKind*

Premultiplies a 1D *srcWave* with the selected window function.

If you include the /PAD flag, the window function is applied to the pre-padded data.

See **Window Functions** below for details.

/Z — Disables rotation of the FFT of a complex wave. Igor normally rotates the FFT result (which is also complex) by N/2 so that x=0 is at the center point (N/2). When /Z is specified, Igor does not perform this rotation and leaves x=0 at the first point (0).

Details

The data type of *srcWave* is arbitrary. The first dimension of *srcWave* must be an even number and the minimum length of *srcWave* is four points. When *srcWave* is a double precision wave, the FFT is computed in double precision. All other data types are transformed using single precision calculations. The result of the FFT operation is always a floating point number (single or double precision).

Depending on your choice of outputs, you may not be able to invert the transform in order to obtain the original *srcWave*.

srcWave or any of its intervals must have at least four data points and must not contain NaNs or INFs.

The FFT algorithm is based on prime number decomposition, which decomposes the number of points in each dimension of the wave into a product of prime numbers. The FFT is optimized for primes < 5. In time consuming applications it is frequently worthwhile to pad the data so that the total number of points factors into small prime numbers.

The hypercomplex transforms are computed by writing the sine and cosine as a sum of two exponentials. Let the 2D Fourier transform of the input signal be

$$F[n_1][n_2] = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} f[k_1][k_2] \exp\left(i2\pi k_1 \frac{n_1}{N_1} \right) \exp\left(i2\pi k_2 \frac{n_2}{N_2} \right)$$

then the two hypercomplex transforms are given by

$$I_c[n_1][n_2] = \frac{1}{2}\left(F[n_1][n2] + F[-n_1][n2] \right)$$

and

$$I_s[n_1][n_2] = \frac{1}{2i}\left(F[n_1][n2] - F[-n_1][n2]\right)$$

Window Functions

The /F=*windowKind* flag premultiplies a 1D *srcWave* with the selected window function.

In the following window definitions, $w(n)$ is the value of the window function that multiplies the signal, N is the number of points in the signal wave (or range if /R is specified), and n is the wave point index. With /R, n=0 for the first datum in the range.

Choices for *windowKind* are in bold.

Bartlet:

A synonym for Bartlett.

Bartlett:

$$w(n) = \begin{cases} \dfrac{2n}{N} & n = 0,1,...\dfrac{N}{2} \\ 2 - \dfrac{2n}{N} & n = \dfrac{N}{2},...N-1 \end{cases}$$

Blackman367, **Blackman361**, **Blackman492**, **Blackman474:**

$$w(n) = a_0 - a_1 \cos\left(\frac{2\pi}{N}n\right) + a_2 \cos\left(\frac{2\pi}{N}2n\right) - a_3 \cos\left(\frac{2\pi}{N}3n\right) .$$

$$n = 0,1,2...N-1.$$

windowKind	a_0	a_1	a_2	a_3
Blackman367	0.42659071	0.49656062	0.07684867	0
Blackman361	0.44959	0.49364	0.05677	0
Blackman492	0.35875	0.48829	0.14128	0.01168
Blackman474	0.40217	0.49703	0.09392	0.00183

Cos1, **Cos2**, **Cos3**, **Cos4:**

$$w(n) = \cos\left(\frac{n}{N}\pi\right)^{\alpha},$$

$$n = -\frac{N}{2},...,-1,0,1,...,\frac{N}{2}.$$

windowKind	α
Cos1:	$\alpha = 1$
Cos2:	$\alpha = 2$
Cos3:	$\alpha = 3$
Cos4:	$\alpha = 4$

Hamming:

$$w(n) = \begin{cases} 0.54 + 0.46\cos\left(\dfrac{2\pi n}{N}\right) & n = -\dfrac{N}{2},\ldots,-1,0,1,\ldots,\dfrac{N}{2} \\ 0.54 - 0.46\cos\left(\dfrac{2\pi n}{N}\right) & n = 0,1,2,\ldots,N-1 \end{cases}$$

Hanning:

$$w(n) = \begin{cases} \dfrac{1}{2}\left[1 + \cos\left(\dfrac{2\pi n}{N}\right)\right] & n = -\dfrac{N}{2},\ldots,-1,0,1,\ldots,\dfrac{N}{2} \\ \dfrac{1}{2}\left[1 - \cos\left(\dfrac{2\pi n}{N}\right)\right] & n = 0,1,2,\ldots,N-1 \end{cases}$$

KaiserBessel20, **KaiserBessel25**, **KaiserBessel30:**

$$w(n) = \frac{I_0\left(\pi\alpha\sqrt{1-\left(\dfrac{2n}{N}\right)^2}\right)}{I_0(\pi\alpha)} \quad 0 \le |n| \le \frac{N}{2}.$$

where I_0 is the zero-order modified Bessel function of the first kind.

windowKind	α
KaiserBessel20:	α = 2.
KaiserBessel25:	α = 2.5.
KaiserBessel30:	α = 3.

Parzen:

$$w(n) = 1 - \left|\frac{2n}{N}\right|^2 \quad 0 \le |n| \le \frac{N}{2}.$$

Poisson2, **Poisson3**, **Poisson4:**

$$w(n) = \exp\left(-\alpha\frac{2|n|}{N}\right) \qquad 0 \le |n| \le \frac{N}{2}.$$

windowKind	α
Poisson2:	α = 2.
Poisson3:	α = 3.
Poisson4:	α = 4.

Riemann:

$$w(n)=\frac{\sin\left(\frac{2\pi n}{N}\right)}{\left(\frac{2\pi n}{N}\right)} \qquad 0\le|n|\le\frac{N}{2}.$$

Flat-Top:

The flat-top windows are defined as a sum of cosine terms:

$$w(n)=\sum_{k=0}^{m}c_k\cos(kz), \qquad z=\frac{2\pi j}{N}, \qquad j=0,1,\ldots N-1.$$

Here are the supported flat-top window keywords for use as *windowKind* with the /WINF flag. These keywords require Igor Pro 8.00 or later:

windowKind	**Cosine Terms**
SFT3F	c0=0.26526, c1=-0.5, c2=0.23474.
SFT3M	c0=0.28235, c1=-0.52105, c2=0.19659.
FTNI	c0=0.2810639, c1=-0.5208972, c2=0.1980399.
SFT4F	c0=0.21706, c1=-0.42103, c2=0.28294, c3=-0.07897.
SFT5F	c0=0.1881, c1=-0.36923, c2=0.28702, c3=-0.13077, c4=0.02488.
SFT4M	c0=0.241906, c1=-0.460841, c2=0.255381, c3=-0.041872.
FTHP	c0=1.0, c1=-1.912510941, c2=1.079173272, c3=-0.1832630879.
HFT70	c0=1.0, c1=-1.90796, c2=1.07349, c3=-0.18199.
FTSRS	c0=1.0, c1=-1.93, c2=1.29, c3=-0.388, c4=0.028.
SFT5M	c0=0.209671, c1=-0.407331, c2=0.281225, c3=-0.092669, c4=0.0091036.
HFT90D	c0=1.0, c1=-1.942604, c2=1.340318, c3=-0.440811, c4=0.043097.
HFT95	c0=1.0, c1=-1.9383379, c2=1.3045202, c3=-0.4028270, c4=0.0350665.
HFT116D	c0=1.0, c1=-1.9575375, c2=1.4780705, c3=-0.6367431, c4=0.1228389, c5=-0.0066288.
HFT144D	c0=1.0, c1=-1.96760033, c2=1.57983607, c3=-0.81123644, c4=0.22583558, c5=-0.02773848, c6=0.00090360.
HFT169D	c0=1.0, c1=-1.97441842, c2=1.65409888, c3=-0.95788186, c4=0.33673420, c5=-0.06364621, c6=0.00521942, c7=-0.00010599.
HFT196D	c0=1.0, c1=-1.979280420, c2=1.710288951, c3=-1.081629853, c4=0.448734314, c5=-0.112376628, c6=0.015122992, c7=-0.000871252, c8=0.000011896.
HFT223D	c0=1.0, c1=-1.98298997309, c2=1.75556083063, c3=-1.19037717712, c4=0.56155440797, c5=-0.17296769663, c6=0.03233247087, c7=-0.00324954578, c8=0.00013801040, c9=-0.0000013275.
HFT248D	c0=1.0, c1=-1.985844164102, c2=1.791176438506, c3=-1.282075284005, c4=0.667777530266, c5=-0.240160796576, c6=0.056656381764, c7=-0.008134974479, c8=0.000624544650, c9=-0.000019808998, c10=0.000000132974.