

---

# *xspectrum Commands and Symbols*

Last updated: 9/22/98

---

---

## *Introduction*

This chapter contains a only description of commands and symbols for `xspectrum`. For a full description of file formats and menu items, refer to the `xspectrum` chapter in the *waves+ Manual*.

`xwaves` can be controlled by manually entered commands, commands in a command file or by commands from the `send_xwaves` facility. Commands from these sources may be passed through to `xspectrum` via the `xwaves send` command (see the *waves+ Manual* and “`xwaves Global Commands`” on page 2).

The syntax of the *sending* commands is:

```
waves send function xspectrum op <command for xspectrum>
```

The syntax of the *received* commands is:

```
receiver command [keyword value] [keyword value] ...
```

The first item, *receiver*, is either the attachment name `xspectrum` or the name of a display object. The second item, *command*, is the actual command name. Just as for `xwaves` there is a distinction between global `xspectrum` commands and object commands. The global commands *always* specify as *receiver* the word `xspectrum`. The other commands specify as

## xspectrum Commands and Symbols

---

*receiver* the name of the object to which the spectrum window is attached. This is the name of the *xwaves* object.

---

### *xspectrum Commands*

#### Object Commands

The following commands are recognized by display objects; this means the *receiver* of a *send* comand is the name of a display object.

**kill** *remove xspectrum display*

This removes the indicated *xspectrum* display object and frees all resources associated with it.

**mark** *compute a new spectrum*

**file** the file name (loaded in *xwaves*) to be analyzed.

**time** The center of the analysis window. This causes *xspectrum* to generate a spectrum based on a window centered at time with window size given by the current setting of **size (sec)**: in the control panel, if *xspectrum* is operating in the **Cursor +- size/2 Window Limits** mode. When operating in the **Markers** mode, the time parameter is ignored.

**rstart** The start of an analysis window (must be accompanied by a setting for *rend*). Only valid in **Markers Window Limits** mode of *xspectrum*.

**rend** The end of an analysis window (must be accompanied by a setting for *rstart*). When operating in the **Markers Window Limits** mode (this equals setting *xspectrum\_window\_limits* to 1) *xspectrum* generates a spectrum starting at *rstart* and ending at *rend*. Then the value specified for *time*, if any, is ignored. When operating in the **Cursor +- size/2** mode, *rstart* and *rend* are ignored.

Regardless of the **Window Limits** mode, `xspectrum` sends a message back to `xwaves` setting the markers to show the window limits used to compute the spectrum.

**print** *print graphics*

This outputs the contents of the label display window as **PostScript** or **PCL** to a file or the line printer. The output attributes may be established using the `xwaves` print setup widget as explained in the *waves+ Manual*. The print attributes described in the *waves+ Manual* may also be set as `xspectrum` variables via `send` commands.

**save\_spectrum** *output spectral values*

**output** Name of a file to receive the computed spectral values. Output is an ASCII file containing header information followed by frequency-value pairs, one to a line.

Only the most recently created “master” spectrum can be saved using this command. In future releases the spectrum output function will be enhanced.

### Global Commands

The following commands are recognized by the object `xspectrum`; this means `xspectrum` is the *receiver* in the `send` command.

**make** *create a display and/or generate a spectrum*

**name** name of an `xwaves` display object to synchronize the displays with.

**file** the file name (loaded in `xwaves`) to be analyzed.

**time** the center of the analysis window. This causes `xspectrum` to generate a spectrum based on a window centered at time with window size given by the current setting of **size (sec)**: in the control panel, if `rstart` and `rend` are *NOT* specified. `xspectrum` sends a message back to `xwaves` setting the markers to show the window limits.

**rstart** the start of an analysis window (must be accompa-

## xspectrum Commands and Symbols

---

nied by a setting for `rend`).

**rend** the end of an analysis window (must be accompanied by a setting for `rstart`). When `rstart` and `rend` are set in the `make` command, `xspectrum_window_limits` is automatically set to 1 (**Markers**). `xspectrum` generates a spectrum based on the data starting at `rstart` and ending at `rend`. The value specified for `time`, if any, is ignored. `xspectrum` sends a message back to `xwaves` setting the markers to show the window limits.

If the display object already exists, then the `make` command simply computes a new spectrum with the indicated parameters.

**quit** *exit xspectrum*

`xspectrum` detaches itself from `xwaves` and removes all of its displays. No data is automatically saved.

**save\_globals** *output values of global variables*

**output** name of a file to receive the values of the global variables. Output is a list of keyword-value pairs in a format suitable for a `.wave_pro`. If `output` is not specified, `stdout` is used. The current settings of all control panel items are reflected in the saved values.

**set** *set globals for use by xspectrum*

**symbol** variables from the list in the “`xspectrum symbols`” section below.

Multiple items can be set with a single command. Spectra are recomputed and the display is updated after any `set` command. All these items can be initialized in the `.wave_pro` file. For compatibility with previous versions of `xspectrum`, some of these variables are accepted without the `xspectrum_` prefix. However, use of the shorter forms in `.wave_pro` files is not recommended, as it is likely to lead to confusion in the long run.

### *xpectrum symbols*

The following symbols are available for `xpectrum`. For an explanation of the type specifiers after the symbols, see Table 1, “List of types for settable symbols,” on page 40.

#### `debug_level` I

Sets the verbosity of debugging messages.

#### `xpectrum_a_autozoom` B

If the value of `xpectrum_a_autozoom` is nonzero, the display is automatically zoomed in amplitude when the horizontal markers are moved. Zero disables automatic amplitude zooming.

#### `xpectrum_analysis_type` I

Specifies the analysis method as follows:

- 0: DFT
- 1: DFTR
- 2: AUTO
- 3: COV
- 4: BURG
- 5: MBURG
- 6: FBURG
- 7: VBURG
- 8: STRCOV
- 9: STRCOV1
- 10: CEPST

#### `xpectrum_cep_cutoff` D

Gives the nominal cutoff frequency (in seconds) for the CEPST analysis type.

#### `xpectrum_cep_liftering` I

Selects liftering operation for the CEPST analysis type:

## **xspectrum Commands and Symbols**

---

- 0: **None**
- 1: **Lowpass**
- 2: **Highpass**

When **None** is selected, the **CEPST** method is equivalent to the **DFT** method.

### **xspectrum\_cep\_trans D**

Gives the duration of the frequency transition region (in seconds) for the **CEPST** analysis type.

### **xspectrum\_ctlwin\_x I**

The  $x$  position of the `xspectrum` control window.

### **xspectrum\_ctlwin\_y I**

The  $y$  position of the `xspectrum` control window.

### **xspectrum\_datwin\_forward I**

If non-zero (default), the `xspectrum` plot window will be brought to the foreground every time a new spectrum is computed.

### **xspectrum\_datwin\_height I**

The height of the `xspectrum` data plot window (pixels). If this value is greater than or equal to 0, upon resize of a spectrum display window the new height will become the value of this symbol. This has the effect that if a spectrum display window opened for a new display object is opened, it will automatically take the height of the last resized spectrum display window.

### **xspectrum\_datwin\_width I**

The width of the `xspectrum` data plot window (pixels). If this value is greater than or equal to 0, upon resize of a spectrum display window the new width will become the value of this symbol. This has the effect that if a spectrum

display window opened for a new display object is opened, it will automatically take the width of the last resized spectrum display window.

### **xspectrum\_datwin\_x** I

### **xspectrum\_datwin\_y** I

If there is only one display object, these two globals control the initial position of the `xspectrum` data plot window. Positive values for `xspectrum_datwin_x` or `xspectrum_datwin_y` specify respectively the *x* and *y* position of the `xspectrum` plot window. If there is more than one display object, the `xspectrum` plot window is aligned with the first display window of the corresponding `xwaves` display object, with the `xspectrum` display directly underneath the left side of the `xwaves` window. If `xspectrum_datwin_x` is negative, the tops of all `xspectrum` plot windows are positioned directly underneath the first `xwaves` display window of the corresponding object. If `xspectrum_datwin_y` is negative, the left side of all `xspectrum` windows are lined up (vertically) with the left side of the first `xwaves` display of the corresponding object.

### **xspectrum\_f\_autozoom** B

If nonzero, the display is automatically zoomed in frequency when the vertical markers are moved. Zero disables automatic frequency zooming.

### **xspectrum\_formants** I

Toggles printing of formant estimates (for LPC-based analysis types):

0: Off

1: On

## **xspectrum Commands and Symbols**

---

### **xspectrum\_harmonic\_cursors I**

Toggles whether harmonic cursors are used:

0: **Off**

1: **On**

### **xspectrum\_horiz\_cursors I**

Toggles the plotting of horizontal lines at the level corresponding to the current frequency selected by the cursor:

0: **Off**

1: **On**

### **xspectrum\_inv\_filt\_dur D**

Gives the size of the region over which to perform inverse filtering.

### **xspectrum\_inv\_filt\_integ D**

Gives the integration coefficient for inverse filtering.

### **xspectrum\_lpc\_order I**

Gives the order for LPC-based (maximum entropy) analysis types.

### **xspectrum\_max\_fft\_size I**

Maximum allowable FFT size for use by the **DFT** and **CEPST** Analysis type item. This has no effect on the **DFTR** item. Any positive integer greater than 512 is acceptable, but it may be possible to exceed the UNIX system swap space (or the life of your computer) if unreasonably large values are used.

### **xspectrum\_max\_lpc\_order I**

This is the maximum LPC (maximum entropy) order that **xspectrum** is allowed to use when computing reflection coefficients. The limit is enforced on the **xspectrum** con-

trol panel. The value must be less than or equal to 200.

### **xpectrum\_plot\_scale** I

Specifies the plotting scale as follows:

- 0: **log pwr (dB)**
- 1: **magnitude**
- 2: **power (sq mag)**

### **xpectrum\_preemp** D

Gives the preemphasis coefficient.

### **xpectrum\_range\_db** I

Specifies the extent, in dB, of the vertical axis when the plot scale is **log pwr (dB)** and automatic zooming in amplitude is not in effect.

### **xpectrum\_reference\_level** D

This is taken to be the 0 dB level when displaying spectra. The spectrum dB reticle and numeric cursor readout values are adjusted accordingly. It is only effective for the **log pwr (dB)** scale mode. This variable can be set either explicitly or via the menu option in the `xpectrum` display window.

### **xpectrum\_reticle** I

Turns the reticle **On** or **Off**, with:

- 0: **Off**
- 1: **On**

### **xpectrum\_window\_limits** I

Determines whether window limits are determined by the **size (sec)**: entry in the control panel centered on the cursor **Cursor +/- size/2** (if 0) or by the current `xwaves` left and right marker positions (**Marker**; if 1).

## **xspectrum Commands and Symbols**

---

### **xspectrum\_window\_size** D

Gives the size of the analysis window (when in **Cursor +/- size/2** mode, not **Marker** mode).

### **xspectrum\_window\_type** I

Gives the type of shaping window to be applied to the analysis data, with:

- 0: **rectangular**
- 1: **Hamming**
- 2: **cos<sup>4</sup>**
- 3: **Hanning**