Mouse button modes

The middle and left mouse-button functions in data display windows can be set to any currently defined operator. This can be either a built-in operator or a user-defined operator (add_op). The functions can be initialized in your .wave_pro file by defining the symbols left_op, middle_op, move_op, spec_left_op and spec_middle_op. Most of them can be modified interactively through the Button Modes menu operation. Of course you can at any time modify them with a set command.

There are a few special and display window-type-specific modes. These modes are described in the following sections.

Modes for all window types

up/down move Causes the cursor positions at the times of button press and button release to set the left and right marker locations. In typical use one presses the left button at one segment boundary location, sweeps the cursor (with button pressed) to the other boundary location and then releases the button. If the segment to be delimited is not displayed in its entirety on the screen, the scrollbar may be used to access the off-screen portions. In this case the button should be pressed and held to establish the left mark.
Mouse button modes

The pointer is then moved into the scrollbar region. The middle button is used to position the display window over the target end. The mouse pointer is moved back into the data region, and the button released at the position for the right mark.

**move closest** Causes either the left or right marker to be moved as long as the button is depressed. The marker nearest the cursor at time of button press will be moved. Note that this works for arbitrarily long files where the closest marker may or may not be currently visible.

**play between marks** D/A converts the signal delimited by the left and right markers.

**repeat previous** When the button is pressed, the most recent operation invoked from the menu (right button) will be invoked again.

---

**Modes for waveform windows only**

**modify signal** If the button is pressed in a non-spectrogram display window, a copy of the corresponding data will be modified. Unless the original signal’s name has the form `.ed.*`, the original file remains unchanged and the edited copy becomes a new signal, the name of which is based on the original’s as follows: if the original name had an extension, `.ed` is inserted before the extension, else `.ed` is appended to the name.

In multidimensional signals, the channel (vector element) to be modified is selected at the time of the initial button press as the channel with the displayed data value closest to the cursor. This selection is maintained until the button is released in the window. The edited trajectory is indicated by an alternately colored dotted line. The numerical data displays are updated to the new values.

Any files so modified are marked to be saved when their
Mouse button modes

display windows are destroyed or when xwaves exits. Upon button release the display may be refreshed to reflect the editing, depending upon the redraw_on_release setting for the display window. If rewrite_after_edit is also set, the modified signal will also be written immediately to disk, rather than later when the display window is destroyed. (See “xwaves Symbols” in waves+ Reference for information on the symbols).

modify signal only works correctly for signals completely contained in xwaves’s internal buffers. The full signal may be read by doing zoom out repeatedly to the full extent of the file, then doing bracket markers on the region to be modified.

blow up time When the button is pressed in the data display region, the waveform time scale will be multiplied by a factor of ten and the waveform will be shifted so as to keep the position of the cursor on the screen constant. If the button is released in the data region, the time scale will be divided by the same factor.

blow up; function Operates the same as blow up time, but also executes the currently-defined blowup_op. This operator may be set by you or established automatically by the most recently attached attachment. In the latter case it typically sends a mark time message to the attachment (like xmarks(1-ESPS)) when the button is released. This permits greater precision in applying boundary marks, etc.

Modes for spectrogram windows only

modify intensity On color or greyscale monitors, horizontal and vertical cursor motion will change the displayed dynamic range and the background threshold on depressing the mouse button, respectively. The white-to-black amplitude Range: and all-white Threshold: are printed when the
Mouse button modes

modify intensity mode is active. On color monitors, horizontal and vertical cursor motion will change the width and level of a red-intensified contour.

move contour The move contour mode causes horizontal and vertical cursor motion to vary the amplitude extent and location of a red-highlighted interval of amplitudes on the spectrogram. This permits rapid quantitative measurements of relative amplitudes by moving the contour “up and down” while observing the numerical printouts of Level: and Range: at the top of the spectrogram.

mark formants The mark formants mode permits hand marking or correction of formant (or other time/frequency) tracks as explained in section “Overlays” on page 39.

When using the modify intensity and move contour operations, the relationship of colors to spectral values is sometimes confusing. A special ESPS file is available to provide visual cues. See the discussion of $ESPS_BASE/lib/waves/files/color.fspec in section “Colormaps” on page 43.

On monochrome monitors, spectrograms are displayed with a Floyd-Steinberg digital halftone algorithm to simulate a greyscale. The halftone image is painted as it is computed. The dithered display can be customized by setting the globals xwaves variables image_clip and image_range; this provides some of the capability of the modify intensity function. After you change the values of these globals, their effect will be seen the next time a spectrogram is created or redrawn with a window size change.