Signal for Windows

Version 2.13

March 2003

First edition (1.00)	May 1997
Second edition (1.50)	May 1998
Third edition (1.60)	July 1998
Fourth edition (1.70)	Nov 1998
Fifth edition (1.80)	May 1999
Sixth edition (2.00)	July 2000
Seventh edition (2.05)	July 2001
Eighth edition (2.10)	June 2002
Ninth edition (2.11)	Jan 2003
Tenth edition (2.13)	Mar 2003

Published by:

Cambridge Electronic Design Limited
Science Park
Milton Road
Cambridge
CB4 0FE
UK

Telephone:	Cambridge (01223) 420186
Fax:	Cambridge (01223) 420488
Email:	info@ced.co.uk
Web:	http://www.ced.co.uk

Curve fitting procedures are based on routines in Numerical Recipes: The Art of Scientific Computing, published by Cambridge University Press, and are used by permission.

Table of contents

Signal for Windows

Introduction

The Signal software running under Windows together with a CED 1401, 1401*plus*, micro1401 or Power1401 interface gives a PC the power to capture and analyse multi-channel waveform and time marker data.

Signal is designed to let you manipulate your data using the familiar Windows idioms. You can arrange the windows to display the data within them to best advantage and cut and paste the results to other applications. Alternatively, you can obtain printer hard copy directly from the application. When you close a data file, Signal saves the screen format and analysis window setup associated with it. When you open a file, Signal restores the configuration, so it is easy to resume work where you stopped in a previous session.

You can analyse sections of data by reading off values at and between cursors, or by applying the built-in frame by frame automated analyses such as waveform averaging and power spectrum. More ambitious users can further automate both data capture and analysis with scripts. The script language is described in *The Signal script language* manual.

If you or your colleagues use the DOS SIGAVG software, you can transfer SIGAVG data files to Signal without effort. Both SIGAVG and Signal use CFS data files and Signal will directly read and use SIGAVG data. SIGAVG can also read data files generated by Signal, but it may not be able to access all the information in Signal files.

New features after version 1.8

Version 2 of Signal is completely compatible with earlier versions; it will read data files and sampling configurations created by Version 1.xx without problems. Version 2 will not however, run under Windows 3.1. The main changes from Version 1 to Version 2 are:

- Can now sample up to 80 waveform inputs at aggregate rates of up to 400kHz, 2.5MHz single channel.
- Digital filtering is now available.
- Error bars are available for waveform averages.
- Active cursors now have an expression evaluator allowing more sophisticated cursor positioning.
- It is now possible to switch between sampling configurations and output protocols with one mouse click.
- Curve fitting of multiple exponentials and Gaussians has been added for file and memory views.
- Data import from other formats.
- Independent channel sizing.
- Scrollable axes.
- XY views now have cursors.
- Results of curve fitting can be included in a trend plot.
- Amplitude histograms are now possible.
- Toolbars have been added for access to commonly used scripts and for book-marking and editing.
- Access to the CED web site from the help menu.
- Waveforms can now be pasted into an XY view.

- Enhanced script editor with syntax colouring, fast function find, split view, bookmarks, block select, block indent and outdent and many more features.
- Data recovery after a power failure.
- “Fast triggers”, a sweep mode with minimal inter-sweep interval.
- Inter-channel arithmetic via modify channels menu – “Subtract channel”, “Add channel” etc.
- Program spawning from a script with the new ProgRun() command.
- Support for the Power1401 gain option has been added.

Work continues at CED to provide new Signal features. It is worth checking the web site (http://www.ced.co.uk) on a regular basis for free updates.

Hardware required

The absolute minimum requirement to run the program is a 386 PC with 8 MB of memory running Windows 95. The more memory you have, and the faster the processor, the better Signal will run, a graphics accelerator will greatly improve drawing and scrolling speeds. If you wish to sample data, you will require a CED 1401, micro1401, Power1401 or 1401*plus* system; pulse output during sampling is not available with a standard 1401.

Installation

Your installation disk(s) are serialised to personalise them to you. Please do not allow others to install unlicensed copies of Signal.

From CD-ROM

Just put the CD-ROM in the drive and it will start the installation. You can also run the installation manually by opening the folder `Signal` on the CD-ROM, then open the `disk1` folder and run `setup.exe`.

From floppy disk

If you are installing from floppy disk you may wish to make backup copies of the installation disks before you begin and keep them in a safe place. We supply the disks write-protected to avoid accidental over-writing, but they can be physically damaged.

If we shipped you a CD-ROM and you must install from floppy disk you can create the disks by copying the contents of the folders `disk1`, `disk2` etc. in the `Signal` folder to separate floppy disks. The resulting disks should hold only the files, NOT the folders!

Place the Signal installation disk 1 in a suitable drive (assumed to be a: below). Click Start, then Run, then type:

```
a:setup
```

During installation (CD-ROM and floppy disk)

You must select a suitable drive and folder for Signal and personalise your copy with your name and organisation. You can run both version 1 and version 2 on the same system (but not simultaneously) as long as they are in different folders. If you have version 1 on your system, install version 2 to a different folder. The installation program copies the Signal program plus help and example files. It can also install 1401 support in Windows NT 4. If you use Windows 95, 98 or NT 2000 there are instructions for loading the 1401 drivers using the Windows Device Manager.

After Installation

If you are new to Signal, please work through the *Getting Started* tutorial in the next chapter. Where you go next depends on your requirements. The *Signal Training Course Manual* is more descriptive than the other manuals which are organised as reference material. However, it covers all versions of Signal and you will occasionally need to refer to the other manuals for version 2 specific details. The on-line Help in Signal has a lot of information; if in doubt use the `F1` key for Help.

Updating Signal

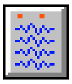
You can update your copy of Signal to the latest version 2 release free of charge from our Web site: http://www.ced.co.uk in the updates section. You can only update a correctly installed copy of Signal version 2. There are full instructions for downloading the update on the Web site.

Once you have downloaded the Signal update, you will find that the update program is very similar to the original installation, except that you must select an installation directory that contains a copy of Signal for Windows version 2.

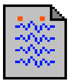
Removing Signal

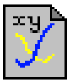
You can remove Signal from your system: open the system Control Panel, select Add/Remove Programs, select CED Signal for Windows version 2 and click Remove. This removes files installed with Signal; you will not lose files you created.

File icons

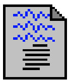


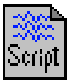
The various file types in Signal have icons so that you can easily recognise them when you minimise their windows. You can also use these icons when you place these files in a group in Program manager. The icons will automatically be used for the relevant files by programs such as Windows Explorer. All the icons have a set of waveforms to remind you of the application to which they belong. The icon to the left is the Signal application icon that you double-click to launch Signal from the Signal program group. There are several sizes of icon.





These icons are for Signal data files. The icon on the left represents Signal CFS data files or file views. The icon on the right represents an XY data file; a saved XY view. If you double-click on one of these in Windows Explorer or Program Manager it will launch the Signal application (if it is not already running) and open the data file.





These icons are for text-type Signal documents and files. The icon on the left represents a text file, which can hold any textual data. The other icon represents a Signal script file. Script files hold script programs that execute within Signal to automate analysis or to customise Signal behaviour in some way.

Using this manual

The first section of this manual introduces you to Signal by suggesting a few tasks you might undertake to familiarise yourself with the system. We have supplied an example data file for you to experiment with, so there is no need to have your own data available at this time. A second chapter introduces sampling new data, subsequent chapters describe using pulse outputs during sampling and multiple frame states.

The second section describes the individual menu commands. We suggest that you work through as much (or as little) of the familiarisation section as you feel you need, then dip into the menu details section as required for more detailed information.

Once you are familiar with the program, you may wish to investigate the script language so you can automate your data capture and analysis. This is described in the separate manual *The Signal script language*. The online help system duplicates all the information in the manuals and is often the fastest way to look up a topic.

Forthcoming attractions

This manual covers version 2.13 of the Signal software and contains some references to features not in this version, but which will be added to later versions of the software, in particular digital markers.

Direct access to the raw data

Some users may wish to write their own applications that manipulate Signal data files directly. A C library*: The CFS library* is available from CED, the library includes all functions necessary to read or write Signal data files from either DOS or Windows programs. This library is available, along with complete documentation in Word format, from the CED web site (http://www.ced.co.uk).

Licence information

CED software is protected by both United Kingdom Copyright Law and International Treaty provisions. Unless you have purchased multiple licences, you are licensed to run one copy of the software. Each copy of the software is identified by a serial number (located in the Help menu About Signal… dialog box). You may make archival copies of the software for the sole purpose of back up in case of damage to the original. You may install the software on more than one computer as long as there is **No Possibility** of it being used at one location while it is being used at another. If multiple simultaneous use is possible, you must purchase additional software licences.

Additional software licences

The original licensee of a CED software product can purchase additional licences to run multiple copies of the same software. CED supplies an additional manual set with each licence. CED does not supply additional software media. As these additional licences are at a substantially reduced price, there are limitations on their use:

1. The additional licences cannot be separated from the original software and are recorded at CED in the name of the original licensee.
2. All support for the software is expected to be through one nominated person, usually the original licensee.
3. The additional licensed copies are expected to be used on the same site and in the same building/laboratory and by people working within the same group.
4. When upgrades to the software become available that require payment, both the original licence and the additional licences must be upgraded together. If the upgrade price is date dependent, the date used is the date of purchase of the original licence. If some or all of the additional licences are no longer required, you can cancel the unwanted additional licences before the upgrade.
5. If you are the user of an additional licence and circumstances change such that you no longer meet the conditions for use of an additional licence, you may no longer use the software. In this case, with the agreement of the original licence, it may be possible for you to purchase a full licence at a price that takes into account any monies paid for the additional licence. Contact CED to discuss your circumstances.
6. If you hold the original licence and you move, all licences are presumed to move with you unless you notify us that the software should be registered to someone else.

Getting started

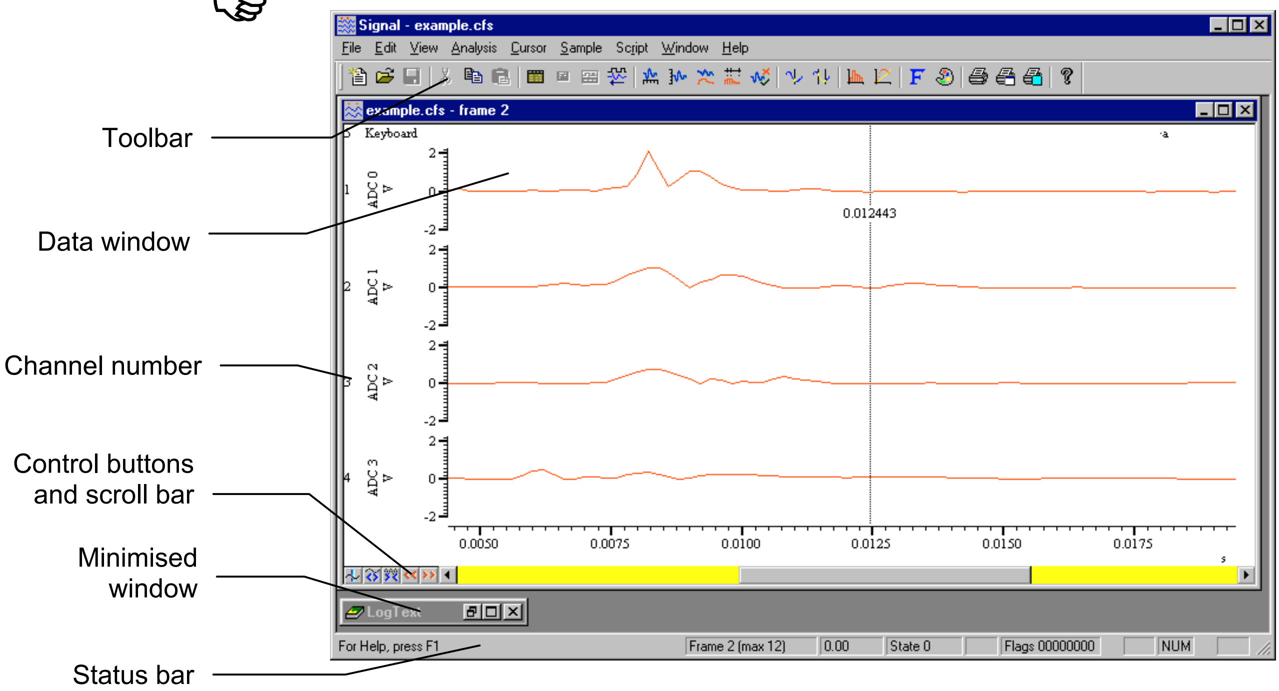
Introduction

In this section you will open a Signal data file, manipulate the contents and familiarise yourself with the basic display and analysis controls. Instructions that you must follow to keep in step with the text are in **bold** type with a pointing finger. Explanations are in normal type. In addition you may like to work your way through the Training manual as this has all the basic information needed to start using Signal.

Basic operations

The first task is to become familiar with the basic operations that are always needed to manipulate Signal files. We use a sample file called `example.cfs`. The Help menu Index command *Getting started* topic duplicates this chapter. Once you have started Signal you may prefer to follow the remainder of the chapter from the on-screen help.

From the Start button choose Programs, then Signal, then example.



You could also have started Signal by selecting the Signal icon and then opened `example.cfs` with the File menu Open command. Signal displays the file as it was last saved (using information in the file `example.sgr`, if it can be found). The picture shows the state as shipped. You are looking at the raw data in the file, we call this a *file view*. This view displays a single *frame* of the data; the axis at the bottom is in seconds.

There are five *data channels* displayed in the window. Channels 1 to 4 hold waveform information and channel 5 holds markers logged from the keyboard. Signal always arranges channels so that the markers are at the top, with waveform channels below.

Below the file view window you will see a minimised window with the title LogText. This is the log view; a text document that is always open in Signal. If you try to close the log view, it is only hidden and can be re-displayed using the Window menu Show command.

Selecting channels

Click on the channel number to select a particular channel. Signal highlights the channel number when it is selected. Hold down the Shift key and click on a channel to select all visible channels between it and the last selection. Hold down Ctrl to select discontinuous channels. Click on the blank rectangular area below the y axes and to the left of the x axis to de-select all channels. Many commands and operations can operate on the selected channels (for example y axis optimisation).

Toolbar and Status bar

The Toolbar is a shortcut to commonly used menu items, each button in the toolbar corresponds to an action from the menu. These buttons are illustrated in this manual along with the relevant menu items. To find out what a toolbar button does, leave the mouse pointer over the button for a few seconds.

The Status bar provides information about the current view. The information in the status bar consists of a number of panes plus an open area on the left. This leftmost portion is used to show prompts from menu items as you move the mouse pointer over the menus, the panes each show a particular item of information. If the space available is too small for all of the panes, panes disappear starting at the right hand side. From the left, the status bar panes are:

Frame — This pane shows the current frame number for the current view and the maximum frame number in the view. If the current view is not a data document view then this pane is blank. See below for a discussion on frames.

Start — This pane, adjacent to the frame number, shows the absolute start time for the frame. For files collected by Signal version 1.00, this will always be zero.

State — This pane shows the state code for the current frame in the current view as a decimal number. If the current view is not a data document view then this pane is blank.

Tag — If the current frame in the current view is tagged (described below), this pane displays the text TAGGED, otherwise it is blank. If the current view is not a data document view then this pane is always blank.

Flags — This pane shows the flags for the current frame in the current view as an 8 digit hexadecimal number (we use hexadecimal format so that the individual flag states can be determined), each digit shows the state of four flags with the highest on the left. If the current view is not a data document view then this pane is blank.

Caps — If the Caps lock on the keyboard is in effect, this pane displays the text CAPS, otherwise it is blank.

Num — If the Num lock on the keyboard is in effect, this pane displays the text NUM, otherwise it is blank.

Record — This pane displays the text REC if Signal is currently recording user actions into a script, and is blank otherwise.

You can hide and show the toolbar and status bar by using the View menu. You can also drag the toolbar and “stick” it to any of the 4 sides of the application window.

A discussion about frames

Frames are a central concept within the Signal software. A CFS file or Signal data document consists of a number of sections or frames, each frame corresponds to a sweep of data. A data document view displays data from a single frame at a time, this is the current frame for that view. You can have duplicate views of the same data document, each view can have a different current frame so you can examine separate parts of the file simultaneously.

Each frame in a Signal data document has the same number and type of channels, but may have varying frame start and end times. Each frame holds channel data from the various channels in the source data file. Usually, all of the waveform channels will have the same number of data points, while the number of markers can vary. In addition to this channel data there are a number of other data items attached to each frame:

Comment: a single line of text, of length up to 72 characters, that can be read or written using Signal.

State: a 32-bit number that can have any value from 0 to over 4 billion, it is intended for use in conditional analysis where each state value corresponds to a separate condition in the experiment, but may be used for any purpose.

Signal can sample using multiple frame states logged from external equipment or generated internally to control 1401 outputs, see the chapter *Sampling with multiple states* (page 43).

Start: a floating point number holding the absolute start time for the frame. This value is the time for the frame x axis zero relative to the start of sampling. For files collected by Signal version 1.00, this will always be zero.

Tag: a single yes/no switch allowing each frame can be tagged or not. Tagging is intended for any purpose when selected frames need to be marked for analysis or attention.

Flags: a set of 32 flags, each of which may be set or clear. These flags are available for any user-defined purpose, currently they are only accessible from the Signal script language.

Variables: a set of 16 floating point numbers that can be read or written using Signal scripts. They are intended for any purpose required.

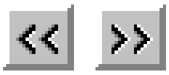
When you are working with Signal data, the current frame for the view is held in memory. This frame will be discarded when the view switches to a different frame. As the entire frame is held in memory, you may find that Signal's performance when handling large frames is improved by installing more memory in your PC. Any changes made to the frame data while it is held in memory must be saved before the new frame is loaded or the changes will be lost. You can write the changed data back into the file using the File menu Save command, or you can use the File:Update mode dialog to select what happens if the frame is changed while data is unsaved. Changes made to non-channel data such as the frame state or flags are always saved.

Specifying frames

Often in Signal you will need to specify the frame or frames to be used for an operation. You can select the current frame in the view or you can enter a single frame number directly. To specify more than one frame, you can enter a frame list such as 1..50,60,61,70..80, or you can select options such as All frames, Tagged frames, Untagged frames or Frames with state n (with a separate field for entering the value of n), giving a wide range of possibilities. These mechanisms are also available from within the Signal script language.

The bottom edge of the data window holds five buttons and a scroll bar. Try them.

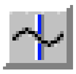
If you resize the window, the same data is redrawn re-scaled to fit the window. The scroll bar controls movement through the data within the current frame, while the buttons allow movement from frame to frame, changing the x axis width and adding a cursor.



Click on these buttons to move to the previous or next frame in the file. All CFS files contain one or more frames which hold similar data, you can use these buttons to move from one frame to another. If the currently displayed frame is the first available, then the previous frame button is greyed-out, similarly for the next frame button. These buttons correspond to the View menu Previous frame and Next frame commands (Page Up and Page Down keys), there is also a Goto frame command.

Click this button to halve the displayed x axis range (zoom in). The left hand edge of the display remains fixed. You can zoom in until the ratio between the total length of the frame and the width of one screen pixel reaches about 2 billion. In practice this means you can zoom in as far as you like. This button corresponds to the View menu Reduce View command (Ctrl + Cursor left).

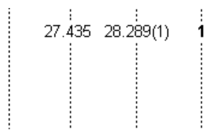
Click this button to double the displayed x axis range (zoom out). The left hand edge of the window does not change unless the start plus the new width exceeds the length of the frame, in which case the left edge moves back. If the new width would exceed the total length of the frame, the entire frame is displayed. This button corresponds to the View menu Enlarge View command (Ctrl + Cursor right).



Click this button to add a cursor to the display (up to 10 cursors can be present in a window). A cursor is a vertical dashed line used to mark positions. You can remove cursors by using the Cursor menu Delete command. You can add a cursor in two ways.

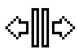
1. To add a new cursor in the centre of the window, click on the button.
2. The Cursor menu New cursor command, or its shortcut Ctrl+|, also adds a cursor in the centre of the window.

Click the cursor button so that at least one cursor is visible. Drag the cursor and observe how the mouse pointer changes. Use the Cursor menu Label mode command.



There are four labelling styles for the cursor: no label, position, position and cursor number, and number alone. You can select the most appropriate for your application using the Cursor menu Label mode command. To avoid confusion between the cursor number and the cursor position, Signal draws the number in **bold** type when it appears alone, and in brackets when shown with the position.

The mouse pointer changes when it is over a cursor into one of two possible shapes to indicate the two actions you can take with a cursor:



This shape indicates that you can drag the cursor from side to side. If you drag the cursor beyond the window edge, the window contents scroll to keep the cursor visible. The position vanishes when dragging unless the Ctrl key is down or you click on the label.

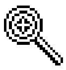
If you position the mouse pointer over the cursor label, the pointer changes to this shape to indicate that you can drag the label up and down the cursor. This can be very useful when you are preparing an image for publication and you need the cursor label to be clear of the data.

Controlling the display

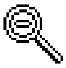
There are many ways to use Signal to adjust or customise the display or to control the data that is displayed.

Move the mouse pointer to the waveform channel, clear of any cursor. Click and drag a rectangle round a waveform feature, then release the button.

This action zooms the display so that the area within the rectangle expands to fill the entire view. If your rectangle covers more than one channel, only the time axis expands. If your rectangle fits in one channel and has zero width, the y (vertical) axis changes to display the selected range and the time axis remains unchanged.



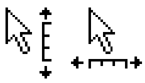
The mouse pointer changes to a magnifying glass when you hold the mouse button down in the data channel area to show that you are about to drag a rectangle or line to magnify the data.



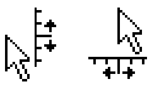
If you hold down the Ctrl key before you hold the mouse button down, the mouse pointer changes to the un-magnify symbol. If you drag a rectangle, the data in the view shrinks to cover an area the same size as the rectangle you have dragged, making this the inverse of the effect without the Ctrl key.

Whichever method used to scale the data, you can return to the previous display using the Edit menu Undo command or the keyboard short-cut Ctrl+Z. If you decide not to expand the display after starting to drag, return the mouse pointer to the original click position (making the rectangle have zero width and height). The rectangle will vanish and you can release the button without changing the display.

Move the mouse pointer over the x and y axes and experiment with clicking and dragging the axes. Try it with the `Ctrl` key held down.



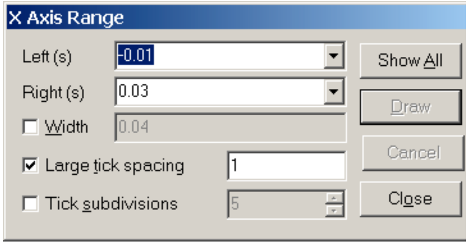
When the cursor is over the tick marks of an axis, you can drag the axis. This maintains the current axis scaling and the window moves to keep pace with the mouse pointer. You can do this with most x and y axes in Signal. This is particularly useful for y axes as they do not have a vertical scroll bar. The window does not update until you release the mouse button. If you hold down the `Ctrl` key, the window will update continuously.



When the cursor is over the axis numbers, a click and drag changes the axis scaling. The effect depends on the position of zero on the axis. If the zero point is visible, the scaling is done around the zero point; the zero point is fixed and you drag the point you clicked towards or away from zero. If the zero point is not visible, the fixed point is the middle of the axis and you drag the point you clicked towards and away from the middle of the axis

In a file view, memory view, or XY view, you can drag the y axis so as to invert the axis. You are not allowed to invert the x axis.

Now double-click on the time (x) axis of the display to bring up the X Axis Range dialog box. Experiment with the settings to vary the time axis.



The Left and Right fields set the window start and end times. The Width field shows the window width. Set the left and right positions, or check the Width box and set the left position and the window width.

You can type new positions or use the drop down lists next to each field that give you access to cursor positions. The Show All button expands the time axis to display all the data. The Draw button updates the display to show the time range set by the Left, Right and Width fields.

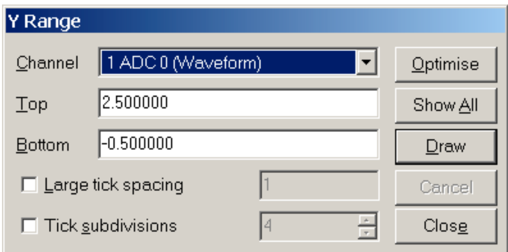
In addition to typing times, or selecting a time from the drop-down list, you can type in expressions using the maths symbols + (add), - (subtract), * (multiply) and / (divide). You can also use round brackets. For example, to display from 1 second before cursor 1 to one second past cursor 1 set Left to `Cursor(1)-1` and Right to `Cursor(1)+1`. The Draw button is disabled if you type an invalid expression, or if the Right value is less than or equal to the Left value or if the new range is the same as the current range.

The Large tick spacing and Tick subdivisions fields let you customise the axis. Values that would produce an illegible axis are ignored. Changes to these fields cause the axis to change immediately; you do not need to click Draw.

Specifying times

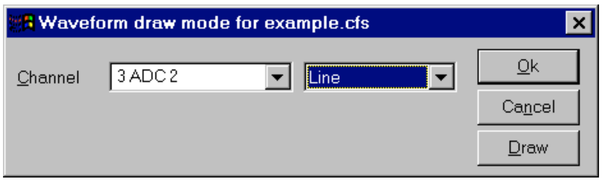
Often in Signal you will need to specify time points within the frame. For example, specifying the X axis limits, the start and end times for file export, or the search limits for an active cursor. Signal provides a standard control that allows you to enter a time directly or to select the frame limits (Mintime() and Maxtime()), the current display limits (XLow() and XHigh()) or the position of any cursors. You can also use an offset value along with the built-in values, for example "Mintime() + 0.75" or "Cursor(1) - 0.1". Note that any numerical values entered always use the currently selected time units.

Now double-click on the y axis of a waveform channel to open the Y Range dialog. Experiment with adjusting the y axis ranges on different channels.

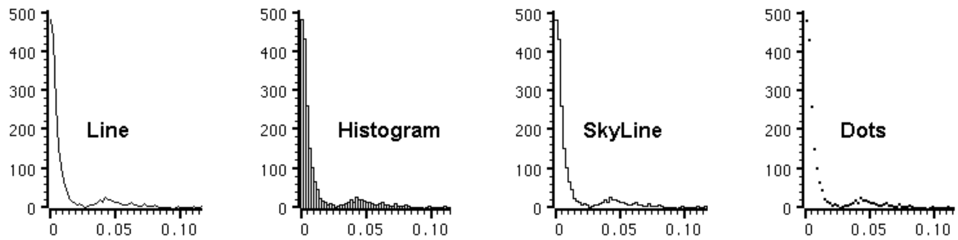


This dialog changes the y axis range of one or more channels. The Channel field is a drop-down list from which you can select any channel with a y axis, or all channels with y axes, or all selected channels, or all visible channels. You can also directly enter a list of channel numbers and ranges such as `1,4,6..10`. You can either Optimise the display, which makes sure that all the data in the window on the specified channel(s) fits in the y axis range, use Show All, which adjusts the display to the data limits if possible, or you can directly set the y axis limits as numbers. You can also control the axis tick spacing, as for the x axis.

Open the View menu Waveform Draw Mode dialog. Experiment with different drawing modes for the channels.

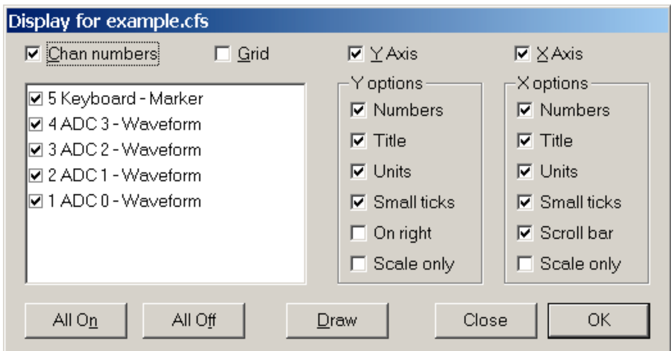


Signal data files hold two basic data channel types: waveform and marker. Waveform data channels hold values that are the amplitude of the waveform at equal time intervals, these are the primary Signal data. Marker data channels hold the times at which something happened, normally markers are used to mark points of interest. The display of any error information may also be manipulated here. There are several different ways to display waveform data (click the Draw button to cause an update without closing the dialog):



The most common draw mode for waveform data is Line mode, you can also select Histogram, Skyline or Dots in place of Line. In Dots mode, you can choose large or small dots (small dots can be very difficult to see on some displays). The pictures above show the result of all types of waveform display. See page 67 in the *View menu* chapter for a complete description of waveform and marker draw modes.

Open the View menu Customise display dialog. Experiment with the channels, axes and grid.



This dialog sets the channels to display in your window. Signal can handle up to 85 or more channels in a file, so this ability is quite important if you are to see any detail!

The list on the left of the dialog holds all the channels that can be displayed. You can also show and hide the axes, grid and scroll bar in the window from this dialog and control the appearance of the x and y axes. Check the boxes next to the items for display and click the Draw button to see the result. The Scale only option draws axes as scale bars.

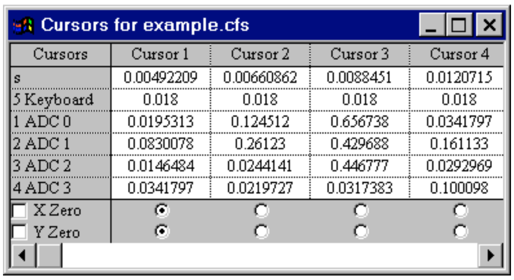
Open the View menu Frame display list dialog. Specify frames 1..4 as the frame list and click OK. Use the View menu Overdraw frame list command to turn overdrawing on and off. Experiment with selecting frames and with the colour cycling display mode.

The frame display list is a list of frames that will be shown in addition to the current frame if overdraw mode is enabled. You can turn overdraw mode on or off by using the View menu Overdraw frame list command. In colour cycling mode, each overdrawn frame is drawn in a different colour, otherwise all the display list frames are drawn in the colour specified by the Frame list traces item in the colour setup dialog. The current frame will be displayed in its usual colour if it is not in the frame list. All the standard mechanisms for selecting frames are available, see page 3 for details.

Cursor measurements

You can use the cursors to take measurements at or between cursor positions.

Make sure you have some cursors in the window, then open the Cursor menu Display Y Values window. Experiment with changing cursor positions and channel display types.


Cursors for example.cfs

Cursors	Cursor 1	Cursor 2	Cursor 3	Cursor 4
s	0.00492209	0.00660862	0.0088451	0.0120715
5 Keyboard	0.018	0.018	0.018	0.018
1 ADC 0	0.0195313	0.124512	0.656738	0.0341797
2 ADC 1	0.0830078	0.26123	0.429688	0.161133
3 ADC 2	0.0146484	0.0244141	0.446777	0.0292969
4 ADC 3	0.0341797	0.0219727	0.0317383	0.100098
X Zero				
Y Zero				

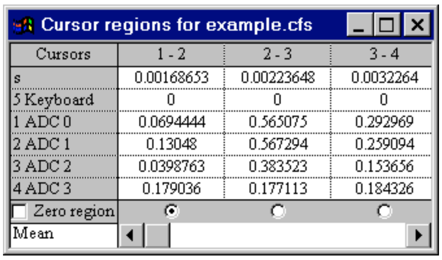
The columns show the cursor positions and the value at the cursor positions for waveform channels. Marker channels displayed as Rate also show the value at the cursor position, marker channels in other draw modes show the time of the next marker after the cursor.

If you want to measure the difference between cursor values, use the X zero and Y zero check-boxes. Use the radio buttons below each column to choose which cursor to make the reference. The values for the reference cursor are shown unchanged, but the values for the other cursors have the value at the reference cursor subtracted. You can use this feature to show how data values have changed from a reference point.
If you move the cursors, change frame or change the channel display mode, the values in the window will update to reflect the change of position. Likewise, if you show or hide data channels in the display, the cursor window display will change to match.

You can select fields in this window and copy them to the clipboard. Click on a field to select it or drag across the data area to make a rectangular selection of fields. Click at the top or left hand edge to select an entire column or row. Click in the top left hand box to select all the fields. Hold down the Ctrl key and click at the top or left hand edge for non-contiguous selection of rows or columns.

You can also print, copy to the clipboard, change the font or copy to the log window using the right mouse button menu.

Now open the Cursor menu Cursor Regions window. Experiment with changing cursor positions and measurement modes.



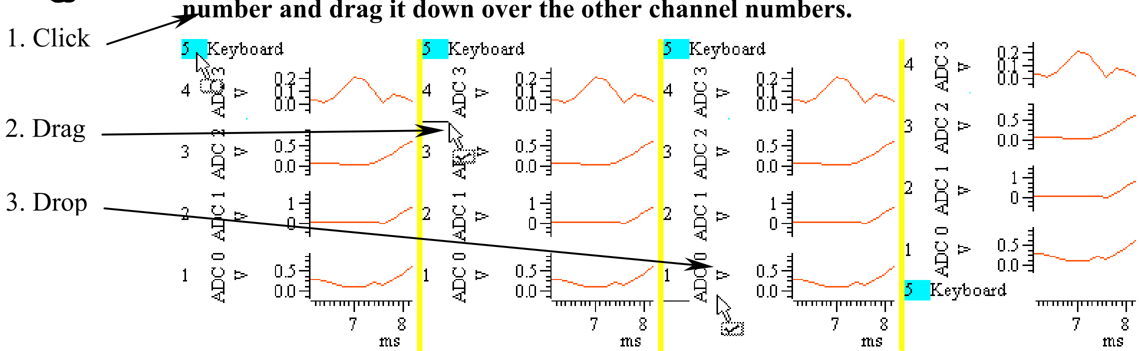
Cursors	1 - 2	2 - 3	3 - 4
s	0.00168653	0.00223648	0.0032264
5 Keyboard	0	0	0
1 ADC 0	0.0694444	0.565075	0.292969
2 ADC 1	0.13048	0.567294	0.259094
3 ADC 2	0.0398763	0.383523	0.153656
4 ADC 3	0.179036	0.177113	0.184326
Zero region			
Mean			

The regions window looks at the data values between cursors. There are many measurement modes including Curve area, Mean, Slope, Area, Sum, Modulus, Maximum, Minimum, Amplitude, SD and RMS. You can select the mode with the popup menu at the bottom left corner of the window, click on the rectangle showing Mean to see the menu. If you want to make measurements relative to one of the regions, check the Zero region box and choose a reference region with the radio buttons.

For a waveform channel or markers as rate, Curve area is the area between the waveform trace and the line joining the intersection points between the cursors and the trace, Mean is the mean level of the signal, Slope is the gradient of the least-squares best fit line to the data, Area is the area between the waveform trace and the y zero level, Sum is the sum of all data points and Modulus is the area, but with all amplitudes considered positive - the 'rectified area'.

For a marker channel in other (not Rate) draw modes, Sum is the number of markers in the region. Mean is the count of markers divided by the width of the region. Slope has no meaning for a marker channel, neither does Curve area, Area, Modulus or the others.

Use the View menu Standard Display command. Click on the Keyboard channel number and drag it down over the other channel numbers.

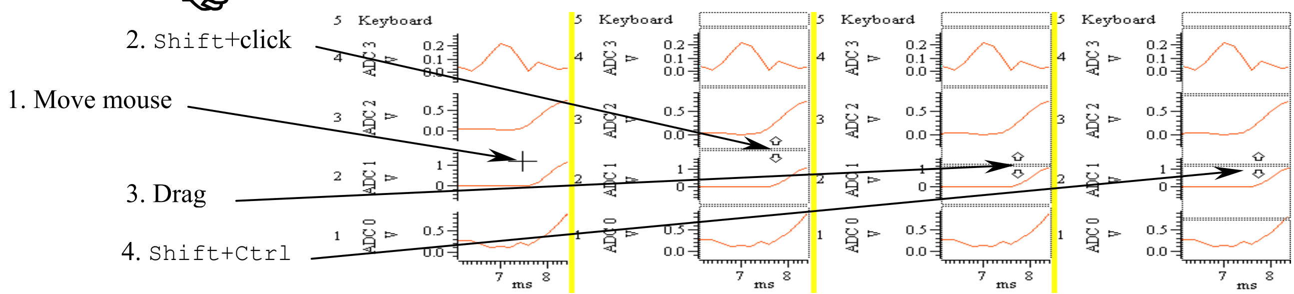


As the mouse pointer passes over each channel, a horizontal line appears above or below the channel. This horizontal line shows where the selected channel will be dropped. Drag until you have a horizontal line below channel 1 and release the mouse button. The Keyboard channel will now move to the bottom of the channel list. Type `Ctrl+Z` or use the Edit menu Undo to remove your change.

You can move more than one channel at a time. Signal moves all the channels that are selected when you start the drag operation. For example, hold down `Ctrl` and click on the channel 3 number. Keep `Ctrl` down and click and drag the channel 2 number. When you release, both channels will move. The mouse pointer shows a tick when you are in a position where dropping will work.

The usual Signal channel order is with low numbers at the bottom of the screen. If you prefer low numbers at the top of the screen, open the Edit menu Preferences and check Standard Display shows lowest numbered channel at the top, then use the View menu Standard Display command.

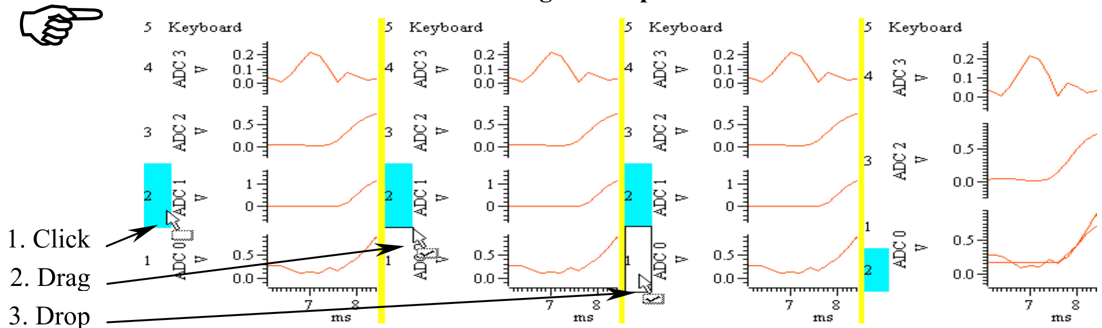
Hold down the `Shift` key and move the mouse over the data area for channel 2. Hold the `Shift` key down and click. Drag up and down and release the mouse.



When you click with `Shift` down, the mouse jumps to the nearest channel boundary and you can change the boundary position by dragging. With `Shift` down, you can move the edge up and down as far as the next channel edge. You can undo changes or use Standard Display to restore normal sizes.

If you add `Ctrl`, all channels with a Y axis are scaled. If there are no channels with a Y axis above or below the drag point, then all channels scale. You can force all channels to scale by lifting your finger off the `Shift` key (leaving `Ctrl` down) after you start to drag the boundary.

Click the "2" of channel 2 and drag it on top of Channel 1 and release.



The channels now share the same space with the channel numbers stacked up next to the y axis. The visible y axis is for the top channel number in the stack. To move a stacked channel to the top, double-click the channel number. Stacked channels keep their own y axes and scaling. To remove a channel, drag the channel number to a new position.

When you drag channels, and at least one of the selected channels has a Y axis, you can drop the channels with a Y axis on top of another channel with a Y axis. As you drag, a hollow rectangle appears around suitable dropping zones. You can also drop between channels when a horizontal line appears.

Merged channels are drawn such that the channel with the visible Y axis is drawn last. If you have a channel that fills in areas, such as a marker channel drawn as rate mode, put is at the bottom of the stack, as it will mask channels below it in the stack.

Memory views

So far, you have been looking at windows holding data read from a disk file. We call these *File views*. There is another type of data window, called a *Memory view*, which holds data created by the Signal program that is held in memory. When a memory view is saved to disk and then re-loaded, it has then become a file view; the two types of view are very similar. A simple way of creating a memory view is by analysing file view data. There are two steps in the analysis:

1. You set the type of analysis, the channels to analyse, the width (or number of bins) of the analysis result and any other parameters required. This creates a new, empty, memory view with the appropriate frame width and channels.
2. You define the frames from the file view that are to be analysed and Signal carries out the analysis and adds the result into the memory view.

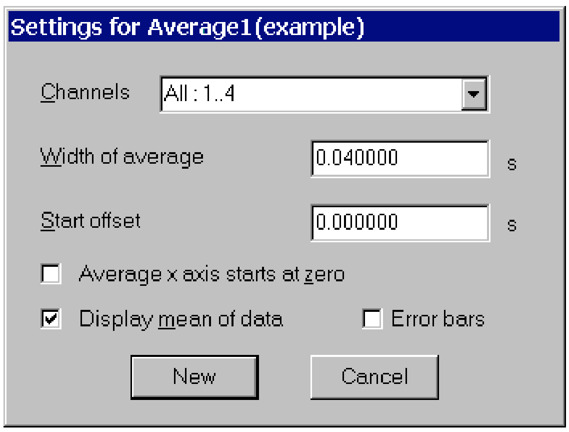
You may repeat step two as many times as required to accumulate results from different sets of frames of data.

You can use the Analysis menu to add additional frames to the memory view. Each frame can hold the result of analysis of different frames from the original file. One way of using this would be to separate averages for each frame state in the source file.

Processed memory views will be automatically re-processed if appropriate. For example, if a memory view holds the average of all tagged frames, and a frame in the source document is tagged or untagged, then the memory view data will be automatically regenerated using the new frames.

The new window behaves like a file view containing one or more frames of data. The simplest way to get a feel for this is to try it, so:

Make the original file view of the data the current window by clicking on it. You may find it easier if you close all the other windows first. Use the Analysis menu New Memory View command to select a Waveform average.



The Settings dialog prompts for information to define the new window. There are three fields that define the waveform average. The Channels field selects the channels to analyse. You can select any channel or list of channels that holds waveform data. The channel list in the pop-up menu only includes suitable channels for analysis.

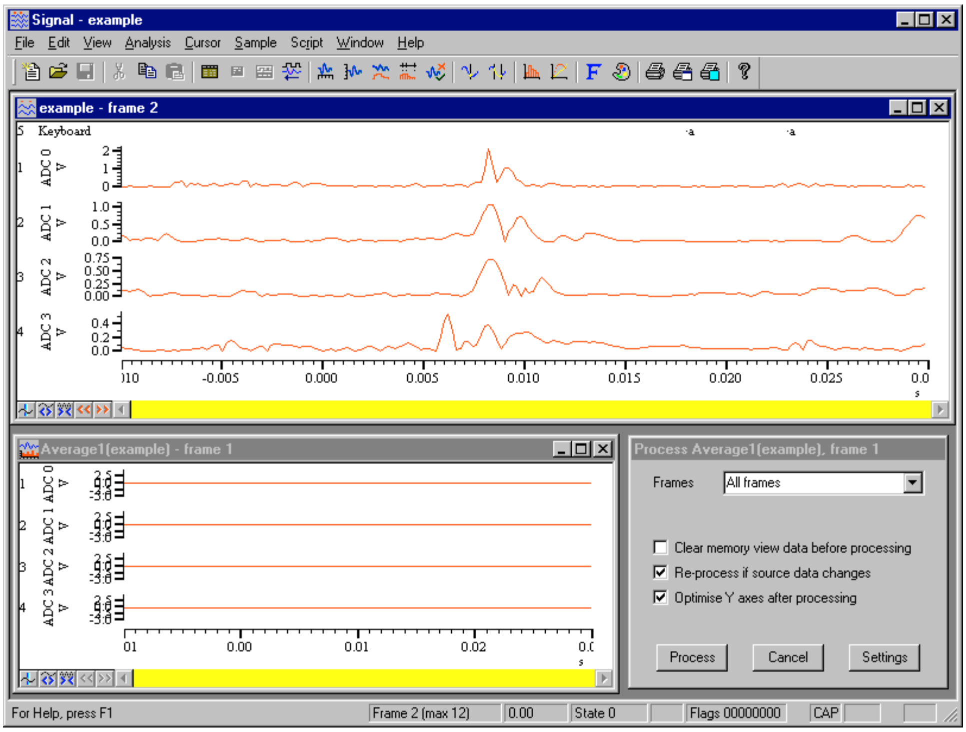
The Width of average field sets the width of the result, in units set by the source data x axis units. You can choose any width you like, limited only by the width of the source frame.

The Start offset field sets the start point within the frame of the data that is included in the average. This is specified as the offset from the start of each frame to the start of the data included, so an offset of zero will use data from the beginning of each frame. Again, this value is in source x axis units.

Below these fields are three checkboxes used to enable various options. The first checkbox is used to force the start time of the memory view data to zero, if this is clear then the memory view data x axis start will be copied from the first data added to the average. The second checkbox selects display of the data as a mean value, if this is clear then the sum of the data is displayed. The third causes error values to be calculated and displayed in the memory view.

The first thing to do is to select the channels to analyse. For this example we use All waveform channels, so select this option using the pop-up menu. Set the other fields as they are in the picture above; 0.04 seconds width and a start offset of 0.

Once you have set these values, click the New button to generate the new memory view. Now set the data frames to analyse.



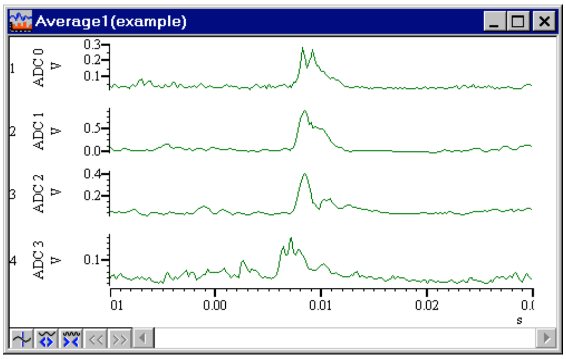
When you click the New button several things happen. Signal creates a new memory window ready to display the result of the analysis, the Settings dialog vanishes and the Process dialog appears. You must now set frames from the data document to analyse.

The three check boxes determine how to treat the result of the analysis. You can choose to clear the memory window contents before you analyse the data, otherwise each new average is added to the previous one. You can also choose to have Signal re-create the average if the source data changes, and you can choose to optimise the memory view display after each analysis so that the full range of the memory data is visible.

If you decide that you have not set the original parameters correctly, you can click on the Settings button to go back to the previous step and correct the values.

When you have set the frames to analyse, click the Process button.

The dialog vanishes and the memory window shows the analysed data.



You can recall the analysis dialog by selecting the Process command from the Analysis menu. Do this now and click the Process button again. The data in the memory window will not change (because this is an average), but the count of sweeps displayed by using the View menu Info command will double (as long as you have not checked the Clear memory view before process checkbox).

Experiment with this new window.

You will find that the new memory view behaves in a very similar manner to the original file view but has only one frame. All the display configuration mechanisms will operate on a memory view, you can create and position cursors and display the cursor regions and cursor values. Most other commands available with a file view can be used with a memory view.

This is a good time to experiment with manipulating the data in the memory view without worries about overwriting file data. Use the Analysis menu Modify channel command to try out some of the options, note that most of the options have keyboard commands assigned.

Saving and reloading memory views

You can save memory view data to disk, as a CFS file. When the CFS file is reloaded into Signal it appears in a file view.

Now select the File menu Save As command.

This displays a standard Windows Save As dialog to allow you to select a name for the file to hold the memory data. The memory view data will be saved as a CFS data file. Once you have entered a suitable name and saved the memory view, close it using the File menu Close command. You can open the file holding the memory view data by using the File menu Open command, it is now opened as a file view with a single frame.

XY views

In addition to file and memory views, Signal also uses XY views. These hold multiple data channels (up to 256) that share the same x and y axes. Each channel is a list of (x,y) co-ordinates. Each channel has its own point marking style, line style and colour. XY views have a wide range of uses, ranging from user-defined graphing to drawing pictures. XY views can be used from the script language, Signal can also create XY views holding data taken from measurements from data files.

☞ **Use the Script menu Run Script command and select the Load and run… command. Locate the `Scripts` folder (in the folder where you installed Signal), and open the file `clock.sgs`.**

Signal will load and run this script which generates an analogue clock in an XY view. You can move and resize the clock window. You can stop the script running (and regain control of Signal) by clicking on the OK button at the upper right hand side of the Signal window.

You could also use the analysis menu New XY view command to select Trend Plot analysis. This will give you a dialog for the trend plot settings allowing the trend plot measurements to be defined. Once the trend plot settings are set a normal process dialog is used to select the frames from which measurements are taken. As for memory view processing, trend plot generation can be saved as part of a sampling configuration.

You can manipulate the XY view using the Signal menus. Most of the Signal commands (for example, Show/Hide channels) act on XY views in the same way as for data views. You will find that the view menu contains new items; Options and XY Draw mode, for XY views and the analysis menu is extended to include Delete channel. The Change colours dialog is also different for XY views. You can read more about XY views in the Edit, View and Analysis menu chapters and in the script language manual.

Summary

If you have followed this chapter, you will now be familiar with the basic actions required to use Signal. The next chapter deals with the special actions required to configure the system for sampling your own data. The remainder of the manual covers the menu commands in the system, copying data to other applications and printing, utility programs and signal conditioner support.

The *Script menu* chapter describes the menu commands that control the script system. The script language itself is not covered in this manual; see the companion text *The Signal script language* for a full description.

Sampling data

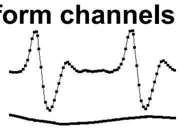
Introduction

If you have worked through the previous section you already have most of the skills needed to work with a new data document created by sampling; a *sampling document*. A sampling document is much the same as an old document, except that the sampling document grows by adding frames to the end. The sampling document also has an extra frame, frame zero, that contains transitory data retrieved as it is sampled. We have modified the Process dialog used during sampling to provide mechanisms for automatic processing and updates, otherwise you would have to use it constantly in order to follow the course of your experiment.

Types of channel

Before we discuss the sampling configuration dialog, we need to provide some background on the types of data channel that Signal can sample. Signal handles two types of channel: waveform and marker.

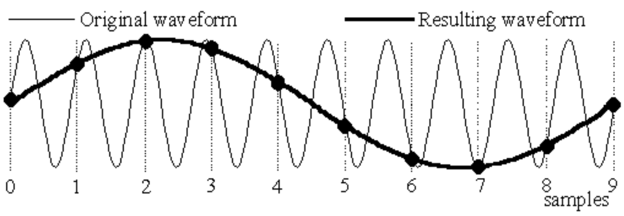
Waveform channels



The waveforms that Signal records and displays are continuously changing voltages. Signal stores waveforms as a list of numbers that represent the waveform amplitude at equally spaced time intervals. These numbers are 16-bit integers. They are scaled using calibration values to produce the floating point data values that Signal uses and displays. Signal can also use and create waveform channels where the underlying data are floating point values, these are indistinguishable from channels using integer data in nearly all circumstances. The process of converting a waveform into a number at a particular time is called *sampling*. The time between two samples is the *sample interval* and the reciprocal of this is the *sample rate*, which is the number of samples per second. A set of samples taken at regular intervals is referred to as a *sweep*.

Minimum sample rate

The sample rate for a waveform must be high enough to represent the data correctly. You must sample at a rate at least double, and preferably 2.5 to 5 times, the highest frequency contained in the data. If you do not sample fast enough, high frequency signals are aliased to lower frequencies, as illustrated above. The dots in the diagram represent samples; the lines show the original waveform. On the other hand, you want to sample at the lowest frequency possible, otherwise your disk will soon be full.



Use of filters

Many users pass waveform data through amplifiers or signal conditioners with filter options to limit the frequency range (see page **122**). Some transducers have a limited frequency response and require no filtering.

Input connections

Connect your waveform signals to the 1401 ADC input ports. Ports 0-7 (0-3 for an unexpanded micro1401 or Micro1401 mk II) are the labelled BNC connectors on the front of the 1401. For the 1401 and 1401*plus* ports 8-15 are on the 15 way Cannon connector. The input voltage range is normally ±5 volts (you can set ±10 volts, see the *1401 Owners manual*). Connections are:

Port	8	9	10	11	12	13	14	15	Ground
Pin number	1	2	3	4	5	6	7	8	9-15

For the Power1401 without an ADC expansion box the extra ADC ports are on the back panel labelled Analogue Expansion. The connections are:

Port	15	14	13	12	11	10	9	8	Ground
Pin number	35	34	33	32	31	30	29	28	1-19

Users of the micro1401 and Micro1401 mk II will find all the ADC ports on BNC connectors. If you have a micro1401 or Micro1401 mk II ADC expansion box installed, ports 4 to 15 are BNC connectors on the expansion box. Expansion boxes are also available for the Power1401. For port numbers above 15 you will require a 32-channel expansion card (for the standard 1401 or 1401*plus*) or extra expansion boxes for the micro1401, Micro1401 mk II or Power1401. If you sample a port above the number available, or a port that is not connected to a suitable signal source, the result is undefined.

TTL compatible signals

In several places in this manual we refer to TTL compatible signals. TTL stands for Transistor-Transistor Logic and is a method of passing logical (High/Low) information between devices using voltage levels. Levels above 3.0 volts are in the High state, levels below 0.8 volts are in the Low state. Levels in between 0.8 and 3.0 volts are undefined.

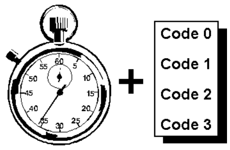
The TTL inputs and outputs on the 1401 are the digital inputs and outputs, the event inputs, the clock F external frequency inputs, the ADC external convert input, the clock output, the DAC Bri output and the micro1401, Micro1401 mk II or Power1401 trigger input. On the micro1401, Micro1401 mk II and Power1401, the event inputs are not actually TTL but can be treated as such.

Do not subject 1401 TTL inputs to voltages above 5.0 volts or less than 0.0 volts. CED hardware has special circuits on TTL compatible inputs to provide some protection, however determined abuse will damage them.

The 1401 TTL compatible inputs are pulled up by a resistor to 5 volts. They require a current of some 0.8 mA to pull them into the Low TTL state. Alternatively, you can connect them to ground to pull them low (useful for the Event inputs).

See the *Owners handbook* of your 1401 interface for full details of each input port.

Marker channels



Signal can sample two types of Marker data: keyboard markers and digital markers. In version 2.00 only keyboard markers are supported. Signal treats both marker types identically once the data has been captured; they differ only in their source.

A Marker is a 32 bit time value plus 4 bytes of marker data. The first of these 4 data bytes is the ASCII code of the keyboard character pressed by the user (for a keyboard marker) or an 8 bit digital code read by the 1401 (for a digital marker). The remaining three bytes are normally set to zero.

Keyboard markers

Keyboard markers time events to an accuracy of, at best, around 0.1 second, you should use digital markers if you require precise timing. The upper and lower case characters a-z and the numbers 0-9 are logged, but *only when the new document window or the sampling control panel is the current window*. The keyboard marker channel, if created, is the first channel after the waveform channels.

Digital markers

The digital markers are timed as accurately as the waveform data. They record 8 separate channels of on/off information, or one channel of 8 bit numbers, or any combination in between. Digital marker data is sampled when a low going TTL compatible pulse is detected as described below. The data is read from bits 0 to 7 of the 1401 digital input. Version 2.00 of Signal does not sample digital markers, this information is provided for future reference only.

Digital marker connections

The digital marker data is read from the 1401 digital inputs bits 0 to 7. These inputs are found on the 1401 Digital inputs connector; a 25-way ‘D-type’ plug located on the front of the 1401 and 1401*plus*, and on the rear of the micro1401, Micro1401 mk II and Power1401. In addition to the data lines, there are optional handshake (h/s) and strobe signals on the digital inputs connector. A TTL pulse is required on the digital inputs Data Available input to log a digital marker. The digital input bit and handshake/strobe connections are:

Digital input bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Gnd
Digital input pin	5	18	6	19	7	20	8	21	13

Signal	in/out	standard	*plus*	other
Data Available	in	pin 23	pin 23	pin 23
strobe	in	pin 24	N/A	pin 23
h/s	out	pin 24	pin 24	pin 24

To log a digital marker, apply a low going TTL pulse at least 1 µs wide to the Data Available input and strobe (if available) input in the table. When the 1401 detects the falling edge of the strobe input, it raises the h/s line to a TTL high state (within a few microseconds). The falling edge of strobe latches the input data on the standard 1401, Power1401, micro1401 and Micro1401 mk II. The h/s returns to a low level after the 1401 reads the input. If you do not provide strobe with the standard 1401 or if you have a 1401*plus* you must keep the digital input data signals stable until the h/s line returns to the low state (this will never be more than 50 µs after the event flag).

Marker codes

When Signal displays marker data from keyboard marker or digital marker channels, it shows the code of the first of the four markers as well as the marker time.

Marker codes have values from 0 to 255. This is the same range of numbers that the ASCII character set uses, and it is sometimes convenient to treat the codes as ASCII character codes (for instance when dealing with keyboard markers). At other times it is more convenient to deal with the codes as numbers.

Whenever Signal displays a marker code that has the same value as the ASCII code of a printable character, it displays the code as a character, otherwise it displays the marker code as a two digit hexadecimal number. Hexadecimal (base 16) numbers use the standard digits 0 to 9, but also use a to f (for decimal 10 to 15). Thus 00 to 09 hexadecimal is equivalent to 0 to 9 decimal. 0a to 0f is equivalent to 10 to 15 decimal. 10 to 1f hexadecimal is 16 to 31 decimal, 20 to 2f is 32 to 47 decimal and so on.

The printable characters (as far as Signal is concerned) span the hexadecimal range 20 to 7e (32 to 126 decimal) and are as shown in the table:

<table>
<tr><th>+</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>a</th><th>b</th><th>c</th><th>d</th><th>e</th><th>f</th></tr>
<tr><td>20</td><td></td><td>!</td><td>"</td><td>#</td><td>$</td><td>%</td><td>&</td><td>'</td><td>(</td><td>)</td><td>*</td><td>+</td><td>,</td><td>-</td><td>.</td><td>/</td></tr>
<tr><td>30</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>:</td><td>;</td><td><</td><td>=</td><td>></td><td>?</td></tr>
<tr><td>40</td><td>@</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td><td>H</td><td>I</td><td>J</td><td>K</td><td>L</td><td>M</td><td>N</td><td>O</td></tr>
<tr><td>50</td><td>P</td><td>Q</td><td>R</td><td>S</td><td>T</td><td>U</td><td>V</td><td>W</td><td>X</td><td>Y</td><td>Z</td><td>[</td><td>\</td><td>]</td><td>^</td><td>_</td></tr>
<tr><td>60</td><td>`</td><td>a</td><td>b</td><td>c</td><td>d</td><td>e</td><td>f</td><td>g</td><td>h</td><td>i</td><td>j</td><td>k</td><td>l</td><td>m</td><td>n</td><td>o</td></tr>
<tr><td>70</td><td>p</td><td>q</td><td>r</td><td>s</td><td>t</td><td>u</td><td>v</td><td>w</td><td>x</td><td>y</td><td>z</td><td>{</td><td>|</td><td>}</td><td>~</td><td></td></tr>
</table>

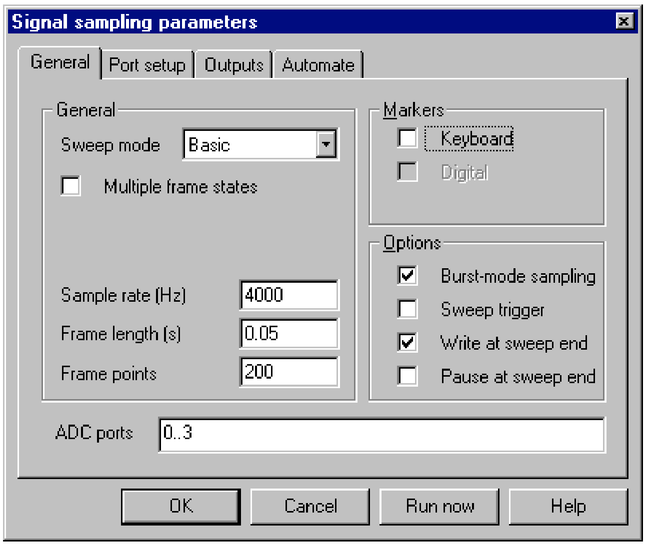
To find the hexadecimal code of a printable character, add the number above the character to the number to the left of the character. For example, the code for A is 41. To convert a code to a character, look up the first digit in the left column and the second in the top row. For example, `3f` codes to `?`, the intersection of the row for `30` and the column for `f`.

Sampling configuration

Before you start to sample data with Signal you must set the sampling configuration. This is done through the Sample menu Sampling Configuration dialog, which is also available by using a toolbar button. The sampling configuration dialog is a tabbed dialog; it contains a number of tabs for selecting separate sections of the sampling parameters. Click on a tab to display the corresponding section of the dialog. The sections always available are General, Port setup, Outputs and Automate. There are other sections, Peri-trigger and States which hold extra information not relevant to all sampling configurations. The Peri-trigger tab appears only when the sampling mode selector in the General section is set to Peri-trigger, the States tab only appears when the General section Multiple frame states item is checked.

General configuration

The General section holds a selector for the sweep mode, the multiple frame states checkbox, fields to define the waveform sampling rate and the frame width or points, checkboxes to control the creation of marker channels and various general options plus a list of ADC ports to sample.



Sweep mode

The Sweep mode selector defines how sweeps of data are taken and triggered and how sweeps relate to the outputs system. It has six selections: Basic, Peri-trigger, Outputs frame, Fixed interval, Fast triggers and Fast fixed int. In Basic mode the trigger, if any, for a sweep of data capture is a TTL pulse at the start of the sweep, and pulse outputs start and finish at the same time as a sampling sweep. In Peri-trigger mode the trigger point can be before the start of the sweep, at the start of the sweep or at any point within the sweep, and pulse outputs start at the trigger point and finish at the end of the sampling sweep. Peri-trigger mode allows a wide variety of triggers including threshold crossings on a sampled waveform channel. In Outputs frame mode it is the pulse outputs that are triggered rather than the sweep, pulses can occur before the sweep starts and after the sweep is over, and the sampling sweep is started by the outputs. Fixed interval mode is similar to Outputs frame, but the sweeps are internally timed so that they occur at the specified interval. External sweep triggers are not used in fixed interval mode. The details of using Outputs frame and Fixed interval mode are covered in the Pulse outputs during sampling chapter (see page 41). Fast triggers mode is like Basic mode except multiple frame states and incremental pulsing are not available. This keeps the inter-sweep interval to a minimum. Fast fixed int mode also has these limitations but uses a fixed interval between sweeps rather than requiring an external trigger.

Multiple frame states

This checkbox enables sampling with multiple frame states. With multiple frame states disabled, all sampling sweeps are the same, the same pulse outputs are generated and the new data frames are set to state zero. With multiple frame states enabled, each sampling sweep can be different from other sweeps in a number of ways and the data frame states are different to indicate what happened during sampling. This can be used to achieve a variety of effects and styles of data acquisition.

The use of multiple frame states is a complex topic which is covered in the Sampling with multiple states chapter of this manual (see page 43).

Sample rate The Sample rate field sets the sampling rate for all waveform channels, in Hz. The rate displayed will not always be the preferred rate that was entered, it shows the closest rate achievable given the 1401 clocks and the number of ADC ports to be sampled. The overall sampling rate in the 1401 is the Sample rate times the number of ADC ports.

With a Power1401, the maximum overall sampling rate is 2.5 MHz if a single ADC port is being sampled, or 400 kHz for multiple waveform channels. A 1401*plus* with a 16 bit ADC can also sample at 400 kHz. A Micro1401 mk II will sample at up to 500kHz. With a micro1401 or a 1401*plus* (with modern 12 bit ADC hardware), the maximum overall sampling rate is 333 kHz. With a standard 1401 or 1401*plus* with older ADC hardware the maximum rate is 82.5 kHz.

The sampling configuration dialog does not apply hardware-specific limits to the sample rates that you enter. If you use a sampling configuration with an overall sampling rate beyond that achievable a 1401 sampling error will occur and be reported by Signal. If Signal detects a Power1401 during program startup, it enables a higher timing resolution. You can force Signal to allow this higher timing resolution by checking the Assume Power1401 hardware box in the Edit menu preferences dialog.

Frame length and points The Frame length and Frame points fields set the length of the sampled frame. The frame length is always shown in the appropriate time units. Changes made to one of these fields automatically cause an appropriate change in the other. The Frame length field also update whenever the sampling rate changes.

The maximum frame length possible varies with the model of 1401 and the 1401 memory installed; each sampled point requires two bytes of memory. For a standard 1401 the maximum number of points (points per frame times number of channels) is about 28,000, for a micro1401 or unexpanded 1401*plus* the limit is about 480,000 while for an expanded 1401*plus* or Power1401 the limit depends upon the amount of extra memory installed but could be up to 8 million. For a Micro1401 mk II the limit is either about 480,000 or 1 million depending on the amount of memory the unit was built with. The sampling configuration dialog does not apply any limits to the frame length that you enter, when sampling starts the 1401 memory required is checked against the memory available.

ADC ports This field sets the ADC ports to sample. You can enter individual ADC ports separated by commas or spaces or a range of ports such as 0..7 or both (for example “0,7,1..6”). Port numbers between 0 and 31 are accepted. Each sampled ADC port creates a separate waveform channel in the resulting data document. The ADC ports are sampled in the order specified and the data document will have all waveform channels first, so the first ADC port provides data for channel 1, the second for channel 2, and so forth. Duplicate port numbers are allowed and will be sampled (see page 15 for a discussion of waveform channels).

Keyboard marker The Keyboard marker checkbox enables the keyboard marker channel and logging of keyboard markers. If the keyboard marker channel is enabled then it is the first channel in the data document after the waveform channels. Checkboxes to enable other marker channels are present but disabled in this version of Signal. (see page 16 for a discussion of marker channels).

Burst mode Check this box for Burst mode sampling, leave it clear for equal interval sampling. In equal interval sampling the waveform data points are sampled individually in turn. The interval between samples is 1/(Sample rate * number of ADC ports). In burst mode sampling all the ADC ports are sampled in a burst, as close together as possible, the interval between bursts is 1/Sample rate. Equal interval sampling has some advantages with the standard 1401 as it loads the 1401 system more evenly, while burst mode ensures that the interval between samples on adjacent ADC ports is kept to a minimum. With a 1401*plus*, Power1401, micro1401 or Micro1401 mk II there is no performance penalty with burst mode. Burst mode is generally recommended because it allows greater accuracy in matching the sampling rate used to that required.

If the first two ADC ports sampled are ports 0 and 7 (or 0 and 3 for a micro1401 or Micro1401 mk II), then the second sample and hold circuit optionally fitted to 1401s is enabled. If fitted this option causes the sampling on ports 0 and 7(3) to be exactly simultaneous. If the 1401 has the 1401-32 multiple sample and hold card fitted, then burst mode sampling will be exactly simultaneous on all channels. Second sample and hold is not currently available for the Power1401.

In Peri-triggered sweeps burst mode is always used for efficiency reasons so the state of this checkbox is ignored.

Sweep trigger This checkbox sets the initial state of the Sweep trigger checkbox in the sampling control panel enable and disable sweep triggers. With sweep triggers enabled, a sampling sweep will not occur until a trigger has been detected, the sampling configuration determines what a trigger is. With sweep triggers disabled, a sampling sweep starts immediately. For Outputs frame sweeps, the sweep trigger starts the outputs rather than the sampling sweep. For Basic or Outputs frame sweeps, the sweep trigger is a TTL pulse that is applied to the 1401 and 1401*plus* event 0 inputs, or to the micro1401, Micro1401 mk II or Power1401 Trigger input. For Peri-triggered sweeps the trigger can be any of a number of signals (see page 21).

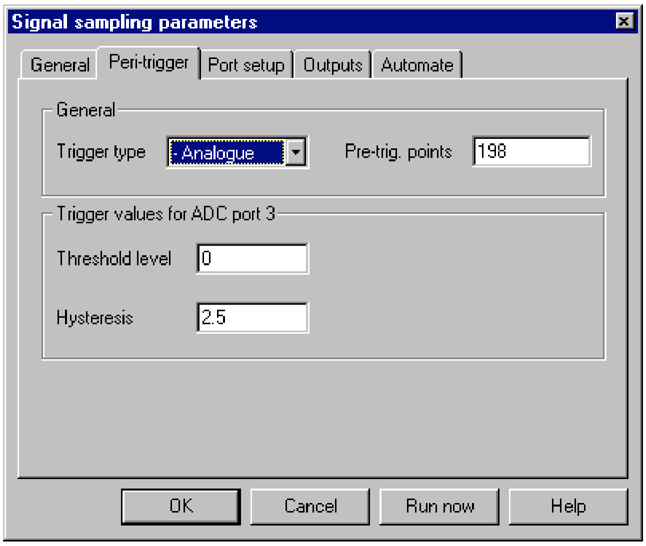
There is a small (~10 microseconds) delay between the time of the sweep trigger and start of sampling. This delay is affected by the outputs synchronisation controls in the Outputs configuration section. When using Basic sweep mode, it is possible to start sampling at exactly the time of the sweep trigger by providing the trigger pulse to both the 1401 E0 and E4 inputs. This mechanism is only available when the synchronised sampling option in the Outputs configuration is disabled. For the micro1401, Micro1401 mk II and Power1401, the trigger input is automatically routed to both E0 and E4 internally if appropriate, guaranteeing a precise start of sampling relative to the trigger.

Signal version 1.00 users please note: The trigger input is no longer event 4 in any sweep mode.

Write sweep to disk This checkbox sets the initial state of the Write to disk at sweep end checkbox in the sampling control panel. When this is set, sampled sweeps are automatically written to disk when the sweep finishes.

Pause at sweep end This checkbox sets the initial state of the Pause at sweep end checkbox in the sampling control panel. When this is set, Signal waits at the end of a sweep instead of immediately starting the next sampling sweep.

Peri-trigger configuration



The Peri-trigger section holds information that is specific to the Peri-trigger sampling mode. It is only available when Peri-triggered sweeps are selected in the General section.

At the top of the dialog is a selector for the type of trigger and a field for the pre-trigger points. Below this is a section holding details of the trigger parameters. The contents of this section changes with the type of trigger, the individual fields will be described along with the various types of trigger.

Trigger type

This can be set to one of +Analogue, -Analogue, =Analogue, Digital or Event. The three analogue types monitor the last ADC port in the sampled ADC ports list for a trigger. The trigger levels are shown with the sampled data as a pair of cursors which can be moved, without stopping the sampling, to alter the levels. The Digital trigger waits for a specified state on a bit in the 1401 digital inputs, while the Event trigger is a TTL pulse just as for the Basic sample mode triggers. Each form of trigger has different parameters.

+Analogue Trigger on a positive-going level transition. The parameters are Threshold level and Hysteresis, both in units set by the channel calibration. The trigger process first waits for the sampled data to go below (Threshold - Hysteresis) and then triggers when the sampled data value rises above Threshold. The hysteresis acts to prevent false triggering by noise as the sampled data passes downwards through the threshold level, triggering can only occur after the sampled data has clearly been below the threshold. If you find that you are having problems with false triggers due to noise, increase the Hysteresis value.

-Analogue Trigger on a negative-going level transition. This is identical to +Analogue, but in the opposite direction. The trigger process first waits for the sampled data to go above (Threshold + Hysteresis) and then triggers when the sampled data value falls below Threshold.

=Analogue Trigger on signal moving outside a pair of levels. The parameters are Upper threshold and Lower threshold. The trigger process first waits for the sampled data to go between the thresholds. It then monitors the sampled data and triggers when the sampled data value is above the upper level or below the lower level.

Digital Trigger on a digital input bit state. The parameters set the digital input bit, from 8 to 15 and select triggering on a high bit or on a low bit. The trigger occurs when the bit is in the correct state. There is no requirement for the bit to be in the other state first. The digital inputs are found on the 1401 digital inputs connector, the pins for the digital bits are:

Digital input bit	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	GND
Digital input pin	1	14	2	15	3	16	4	17	13

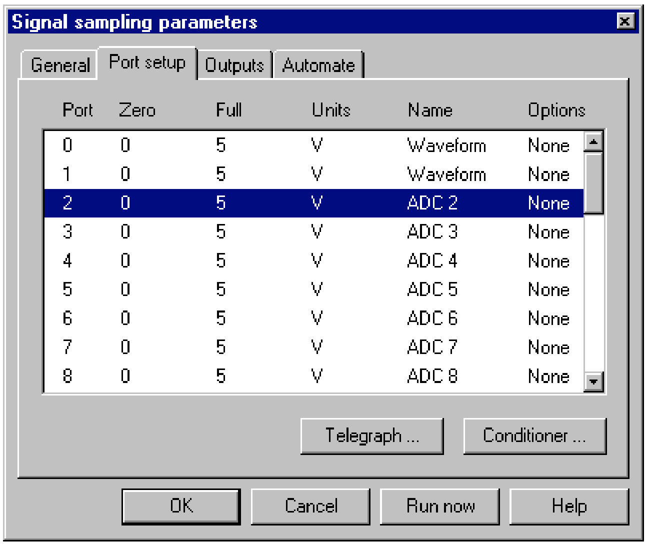
Event Trigger on a TTL pulse. There are no parameters; the trigger occurs when a TTL pulse is detected on the standard 1401 or 1401*plus* Event 0 input, or the Trigger input on a micro1401 or Power1401.

Pre-trig. time This parameter sets the number of points in the frame before the point at which the trigger occurred. This can have any value from -(1,000,000 * sample interval) to the length of the frame -(2 * sample interval). If the value is negative, this means that points sampled after the trigger occurs are discarded before the first point in the frame is kept. If the value is positive, then the specified time must have elapsed before the search for a trigger begins and the resulting frame contains points sampled before the trigger occurred.

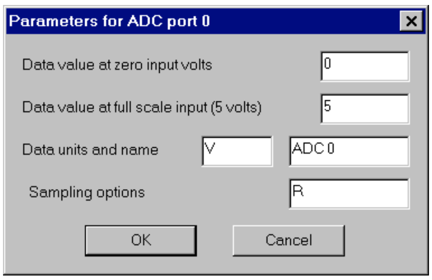
When a non-zero pre-trigger time is specified the resulting data x axis adjusts to start at -pre-trig. time. Thus a negative value gives an x axis starting at some positive value because the first point in the frame was sampled some time after the trigger. Similarly, a positive value gives an x axis starting at a negative value as some points sampled before the trigger are shown.

Ports configuration

The port setup section is used to set up the individual ADC ports. You can set the scaling and units for data sampled from a port, the name of a data channel taken from a port, and specify online processing options for data from a port. The main dialog displays the current settings for all of the available ADC ports. Double-click on the entry for a particular port to open the parameters dialog for that port. The entries for each port (both in the main dialog and in the parameters dialog) are:



Zero The value (in the specified units) corresponding to a zero volt reading from the ADC. This value, along with Full, is used to convert ADC data into the floating-point values used by Signal.



Full The value corresponding to a full scale +5 volt (+10 volts for a 10 volt 1401) reading from the ADC.

Units The units for calibrated data. This is a string from 1 to 6 characters long. If you set the first character of the units to be a space, then future versions of Signal will be able to automatically adjust the units by replacing the space with a character representing a factor of a 1000 such as µ or k.

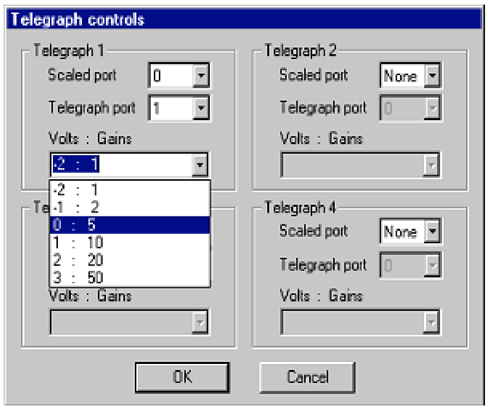
Name The port name. This is a string from 1 to 19 characters long, it sets the title of the waveform data channel sampled from this port.

Options This is a string of 0 to 8 characters that holds online processing options for data from the port. Characters corresponding to various processing options can be entered into this dialog. Currently, only one processing option is available; enter an 'R' character to cause online rectification of sampled data.

The Conditioner… button opens the signal conditioner setup dialog if a signal conditioner has been found (see page **122**).

The Telegraph… button opens a dialog that allows you to configure telegraph output support. Telegraph outputs are analogue outputs from an amplifier that signal the current amplifier settings, chiefly gains. By reading and interpreting the amplifier telegraph outputs, Signal can automatically adjust for manual changes in the amplifier gain settings.

Telegraph configuration



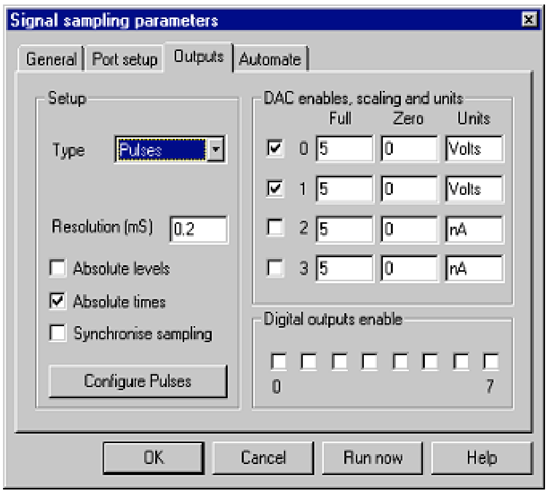
The telegraph configuration dialog can be used to configure up to four ports whose scaling depends upon telegraph outputs. For each port to be scaled, you can select a port from which the telegraph signal should be read and you can enter a list of signal values and the corresponding amplifier gains.

The Scaled port item sets the ADC port whose scaling is controlled by a telegraph signal. Set this to None to disable use of this telegraph. The Telegraph port item sets the ADC port from which the telegraph signal will be read.

The Volts : Gains control is used to enter a list of telegraph signal voltages and corresponding amplifier gains. Each entry holds a voltage value and a corresponding gain, separated by a colon. To add a new telegraph voltage and gain pair; just type the values in separated by a colon character, and press Tab to enter the new values into the dialog. To change the gain for a telegraph voltage, select that entry and edit the gain value, then press Tab as before. To delete a voltage/gain pair, select that entry and delete all the text, then press Tab.

The telegraph voltage values are not affected by the Full and Zero calibration values set for the telegraph ADC port, they are raw voltages as read from the 1401 inputs and assume that the ADC full scale is 5 volts. If you have a 10 volt ADC, you should divide the actual telegraph voltages output by your amplifier by 2. The calibration of the scaled port in the main ports configuration page should be set up using a gain of unity so that the telegraphed gains can be correctly applied. During sampling, the telegraph signals are read at the start and end of each sampling sweep and used to adjust the signal scaling. This will cause changes in the signal levels due to gain changes to disappear, the only effect of a gain change will be to the Y axis Show All limits for that channel.

Outputs configuration



Use this section of the sampling configuration dialog to set the outputs required during sampling, to set which DACs and digital outputs are available for use and to set the DAC units and scaling. The leftmost area is used to configure the outputs, it contains a selector for the type of outputs required plus items specific to the type of outputs. The right-hand areas enable and set up the output ports.

Outputs type This control selects the type of outputs to be used. It can be set to either None or Pulses. The controls in the area below the selector vary according to the selection.

Outputs type : None This disables outputs during sampling. When this is selected only a single control is shown:

Timer period (ms) This item sets the period of the internal timer used to measure the absolute frame start time and (in future versions of Signal) to time digital markers. A value of 1 to 10 ms is usually appropriate for these purposes, values from 0.1 microseconds to 250 ms can be entered, values are rounded to the nearest 0.1 microseconds.

Outputs type : Pulses This selects pulse outputs during sampling. The pulses can be controlled by the script language or interactively using a dialog. The details of configuring and using pulse outputs are covered in a separate chapter of this manual: Pulse outputs during sampling (see page 34). When pulse outputs are in use, a number of controls to configure the pulses are shown:

Resolution (ms) This sets the timing resolution of the output pulses in milliseconds or microseconds and also sets the period of the internal timer used to measure the absolute frame start time and (in future versions of Signal) to time digital markers. Values from 0.1 microseconds to 250 ms can be entered, values are rounded to the nearest 0.1 microseconds. The practical limit to the resolution depends upon the type of 1401 in use; for a 1401*plus* the recommended limit is 3 ms, for the micro1401 values down to 0.1 ms can be used, for the Micro1401 mk II 25 microseconds, while for the Power1401 you can go down as far as 10 microseconds.

Absolute levels This selects between absolute and relative pulse levels. With absolute pulse levels, the pulse amplitude sets the level directly, with relative levels the pulse amplitude is added to the level before the pulse to get the actual pulse level.

Absolute times This selects between absolute and relative pulse times. With absolute times, the pulse dialog allows you to enter the pulse start time directly, with relative times you use the delay since the start of the previous pulse. This control only affects the way in which the pulse dialog handles pulse start times, not the other times shown in the dialog, the underlying pulse data or the generation of pulses.

Synchronise sampling This selects between mechanisms for synchronising external triggers, pulse outputs and the actual sampling sweep. Normally, the sweep trigger starts the sampling sweep and the pulse output mechanism, which is free-running throughout sampling, is synchronised to the sweep. This arrangement gives the maximum accuracy of the sampling sweep start time relative to the sweep trigger (about 2 to 5 microseconds), but will give a 10 to 20 microsecond delay in pulse outputs relative to the sampling. If you select synchronised sampling, the sampling sweep start is delayed relative to the sweep trigger by 10 to 20 microseconds, but the pulse outputs are more precisely synchronised with the sweep, to an accuracy of 2 to 5 microseconds.

If Outputs frame, Fixed interval or Fast fixed int sweep mode is used, this item is ignored, as the sweep is triggered directly by the pulses system, giving the same effect as if synchronised sampling was selected.

When using Basic sweep mode, it is possible to start sampling at exactly the time of the sweep trigger by providing the trigger pulse to both the 1401 E0 and E4 inputs. This mode of operation is only available when synchronised sampling is disabled. On the micro1401, Micro1401 mk II and Power1401 in this circumstance, the trigger input is automatically routed to both E0 and E4 internally, guaranteeing a precise sampling start relative to the trigger.

Configure pulses Press this button to configure the output pulses using the pulse configuration dialog. Details of this are given in the chapter Pulse outputs during sampling (see page 35).

DAC enables, scaling and units This section contains four sets of controls, one for each DAC (users of micro1401s and Micro1401 mk IIs should ignore DACs 2 and 3). These control if a DAC is available for use and set the scaling and units with which DAC values are defined.

Enable These checkboxes enable the DACs for use. Set a checkbox to use a DAC, leave it clear otherwise. The fewer DACs are enabled for output the more space is available for the display of each DAC in the pulse dialog.

Zero The value (in calibrated units) corresponding to a zero value output from the DAC. This value, along with Full, is used to convert the floating-point values used by Signal into the integer quantities actually used by the DAC hardware. This conversion process occurs when generating pulse outputs, when waveform data is pasted into an arbitrary waveform pulse and when compiling pulse sequences.

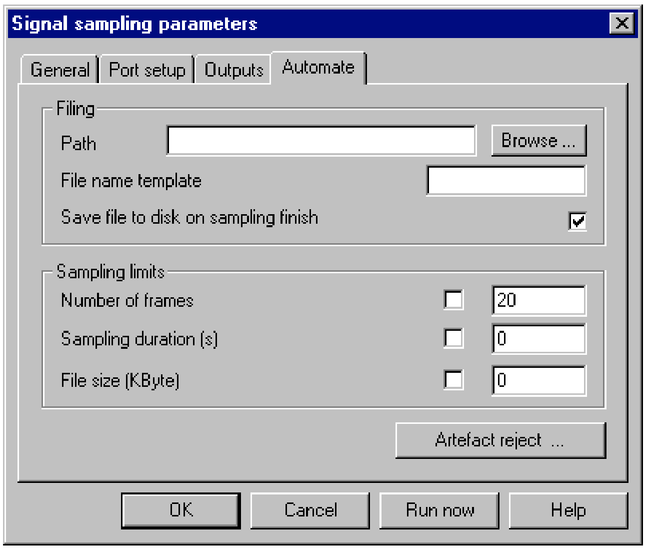
Full The value corresponding to a +5 volt output from the DAC; the maximum output possible.

Units The units with which the DAC output scaling is specified. This is a string from 1 to 6 characters long.

On the Power1401 DACs 2 and 3 are available on pins 36 and 37 respectively of the rear-panel analogue connector.

Digital outputs enable This section contains a set of checkboxes to enable and disable the individual digital outputs for use. Set the checkbox to use this digital output port, leave it clear otherwise. The fewer digital outputs are enabled for output, the more space is available for the display of each output in the pulse configuration dialog. See the Pulse outputs during sampling chapter for details of the digital outputs (see page 34).

Automation configuration



Use this section of the sampling configuration dialog to configure various automation features in Signal. There are two areas of the dialog: Filing, that controls automatic file name generation and automatic filing, and Sampling limits, that can be used to restrict the amount of data sampled or filed, plus a button used to access the artefact rejection dialog.

Path

This sets the directory where the automatic file naming looks to produce a unique file name and where new files are saved after sampling has finished. This is distinct from the directory for new files set in the Preferences dialog which sets the location for the temporary files used while sampling. If this field is blank, automatic file name generation and file saving use the current directory. You can enter a directory path directly, or use the Browse button to select a directory.

File name template

This sets the template for automatic file name generation. If this field is blank, automatic file name generation is disabled and normal document names (Data1, Data2, etc.) will be used for sampled data, these names are used as a default when saving data. If a template is provided, this is used to generate a sequence of unique file names for the sampled data. If the template is a simple string, unique file names are generated by appending a three digit code to the end of the template. The code used is determined by searching the current directory for existing names and using the first unused value. If the template ends with one or more digits, these set the length and initial value of the numeric code used. Thus a template of "testdat" would generate filenames "testdat000" to "testdat999", while "testdat10" would generate "testdat10" to "testdat99".

If a file name template is set, the generated name is used automatically when the data file is saved without the user being prompted to confirm the name. A different name can be specified using the File Save As command.

Save file to disk

If this checkbox is set, the new data file will automatically be saved to disk when sampling finishes. If automatic filename generation is in use, the generated filename is used, otherwise the usual prompts for a file name from the user are generated.

Sampling limits

This part of the dialog controls three limits; Number of Frames, Sampling duration and File size. Each option has a checkbox to enable the limit plus a field for entry of the limit value. If the checkbox for a particular limit is clear, or if the corresponding limit value is set to zero, then that limit is disabled. Note that all of these limits cause sampling to stop, not finish, sampling can still be continued after a limit is reached.

In addition to these user-defined limits, Signal has a built-in sampling limit based on the available free disk space. If the available free disk space drops below 0.5 Mbytes, sampling stops automatically. This limit cannot be disabled as it is important not to allow hard disks to get completely full, both because this can slow down file accesses considerably and also because of the trouble a full disk gives to an operating system such as Windows 95 that uses spare disk space for virtual memory management. The user-defined limits are:

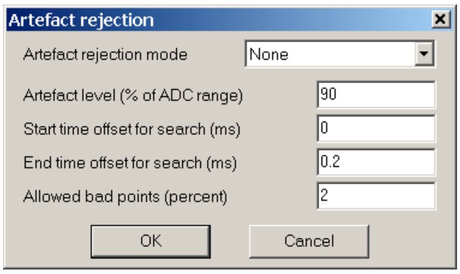
Number of frames If this limit is enabled, this will cause sampling to stop when the set number of frames have been written to the data file.

Sampling duration If this limit is enabled, this will cause sampling to stop when the set time has passed since sampling was started.

File size (Kbyte) If this limit is enabled, this will cause sampling to stop automatically when the file size reaches or exceeds the set size.

The Artefact reject … button opens a dialog that allows you to configure artefact rejection. This provides mechanisms for automatically rejecting or tagging sampled frames if they contain an artefact, normally caused by stimulation.

Artefact rejection



The artefact rejection dialog can be used to configure Signal to automatically examine newly sampled data and, if the data has reached the ADC limits, to reject or tag the new frame. Artefact rejection is particularly important when generating averaged evoked responses, particularly from the EEG, where artefacts often occur and where the mathematical rigor of the averaging process will be affected by the presence of signals at the ADC limits.

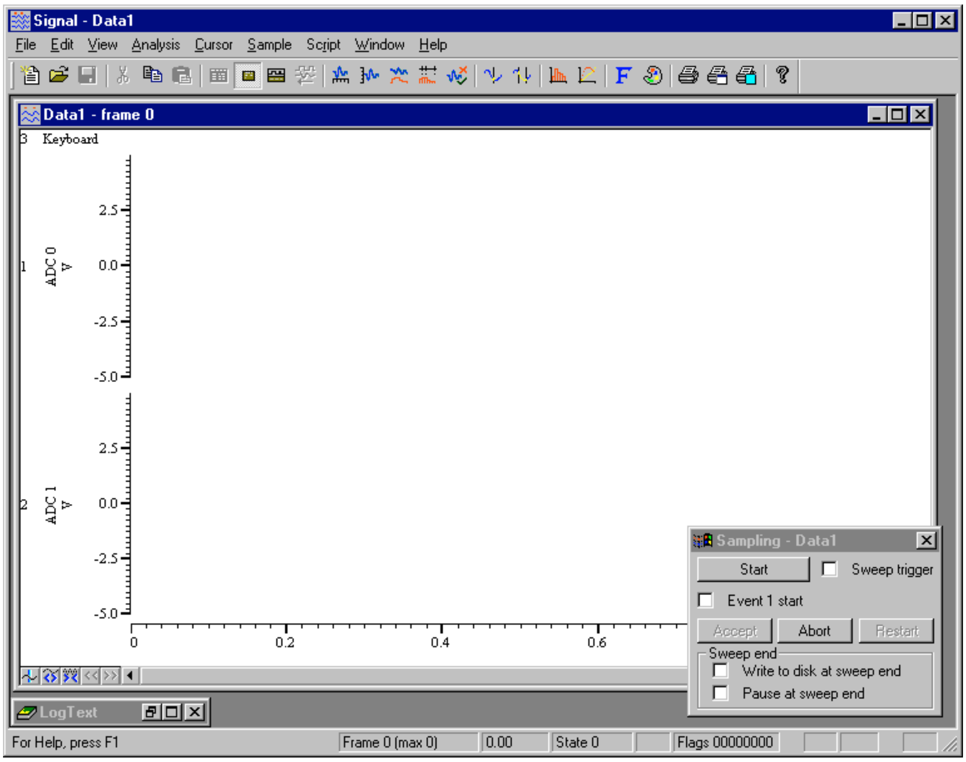
The Artefact rejection mode item controls what form of artefact rejection is carried out, it can be set to None for no artefact rejection, Tag frames to label frames with artefacts, or Reject frames, to discard frames with artefacts. The Artefact level is the percentage of the ADC range outside which will be regarded as an artefact. The next two items set the time range for the search for artefacts, these are specified as offsets from the start of the data rather than as absolute frame times. The final item, Allowed bad points (percent), sets the limit before the frame is rejected or tagged, allowing you to avoid rejecting frames with a trivial amount of bad data.

Unless the mode is None, each sampled frame of data is scanned for artefacts. All waveform channels are scanned over the time range specified and the number of points at the ADC limits are counted. If the number of bad points, expressed as a percentage of the total points scanned, exceeds the limit for allowed bad points then that frame is automatically rejected or tagged as appropriate.

Creating a new document

Once you have set up the sampling configuration you can create a new sampling data document. Either click the Run Now button in the sampling configuration dialog or the Run Now button on the Toolbar, or Select New from the File menu, then Data Document for the file type.

Typical display after creating a new file



The exact appearance of the new document view varies, depending on the configuration. The name for the new document will either be Data1, Data2 etc, or a name produced from the file name template if one is available. You can customise the view by adjusting the x axis, y axes, channels and other aspects of the view.

Frame zero

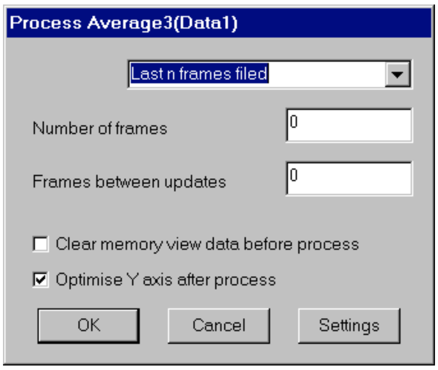
A sampling document is different from other types of data documents because it starts at frame zero, while all other types start at frame 1. Frame zero is a special frame that holds the transitory data for the sweep currently being sampled, whereas frames 1 onwards hold data that has been written to the new file. Sampling is a cyclical process of collecting a new sweep of data into frame 0, deciding (by a variety of mechanisms) if it is to be written to disk and writing it if necessary, clearing frame 0 and then starting off the next sweep. This process continues until enough frames have been written to disk or until sampling is stopped.

Data is shown in frame zero for as long as it is the most recent data; frame zero is cleared when pulse outputs for the next frame starts, or when sampling of the next frame is triggered. For Basic and Peri-trigger sweeps, this means that data is displayed right up until data from the next sweep starts to be drawn. For Outputs frame, Fixed interval or Fast fixed int sweeps, the frame zero data will be cleared when the pulse output for the next frame starts and the new data starts to be drawn when the sampling is triggered. There may be a period when no data is shown, which provides the user with a clear indication of when the next sweep has started off. This is valuable because, once the pulse output or sampling has started, the Accept/Reject button in the control panel cannot be used on the previous sweep of data any more.

The small floating window is the *Sampling control panel*, it contains buttons and other controls to interact with the sampling. Sampling will not start until you click the Start button in the Sampling control panel (the Sample menu duplicates the panel controls). If the Event 1 start box is checked, sampling waits for an external signal after the start button is pressed.

Online analysis

You can customise the display and set up any analyses required before starting the sampling. For example, to set a waveform average you can select that analysis exactly as you did in the *Getting started* chapter. There is a difference, however, when you click on New, the online processing dialog appears.



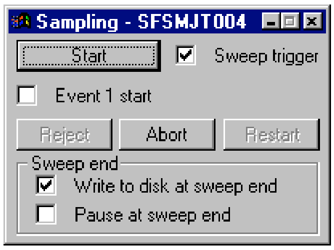
This dialog controls which frames to include in the processing and how to update the memory view holding the analysis result (see page 77 for a full description of this dialog).

The most common mode is All filed frames, with an update every 0 frames (every frame).

This dialog disappears once you select either OK or Cancel, you can recall it with the Process command in the Analysis menu.

Sampling control panel

The sampling control panel holds several buttons and checkboxes used to control and interact with data sampling. When the control panel appears it looks like the picture to the right. Click Start to begin sampling, or Abort to give up immediately. Once sampling has started the buttons change but most checkboxes are always present and can be changed at any time during sampling.



The Event 1 start checkbox allows you to trigger the entire sampling process externally. If this checkbox is set, clicking on Start will not start sampling directly, it enables the start of sampling on an event 1 pulse. This allows precise control of the time of the start of sampling; the time from which the frame absolute start times are measured. While Signal is waiting for an E1 start pulse, the Start button will flash 'Waiting for E1'.

The Sweep trigger checkbox is initially set from the Sweep trigger item in the sampling configuration. It enables or disables triggered sweeps, while Signal is waiting for a sweep trigger the checkbox text will show 'Waiting (E0)'.

The checkboxes in the Sweep mode section of the dialog control what happens at the end of a sweep of sampling and how data is written to disk. If the Write to disk at sweep end checkbox is set, then each frame zero will be written to the disk file automatically when the sweep finishes. If the checkbox is clear, then the frame is not written. You can override this behaviour for individual sweeps using the Accept/Reject button in the sampling control panel.

The Pause sampling at sweep end checkbox controls whether a new sweep is started automatically once the previous sweep has ended. If the checkbox is set then sampling will pause until the Continue button is pressed, or until the checkbox is cleared again, allowing the user to pause for a while or to inspect the data and accept or reject each frame. If the checkbox is clear then the next sweep starts immediately.

The Write to disk at sweep end and Pause sampling at sweep end checkboxes provide four sampling modes:

Write	**Pause**	**Effect**
No	No	Continuous sweeps for signal monitoring
Yes	No	Continuous sweeps written to disk
No	Yes	Interactive sweeps written to disk if accepted
Yes	Yes	Interactive sweeps removed from disk if rejected

The sampling control panel can be hidden or shown using the Sampling menu, the Signal toolbar or the pop-up menu provided by right-clicking the mouse on unused parts of the Signal window. It can also be hidden by clicking on the **x** button at the top right of the control panel, or minimised using the **–** button.

During sampling

Once sampling has started the sampling control panel changes to show buttons suitable for the sampling process. If you check the Event 1 start option and click Start, the window contents change and the word Waiting flashes until a suitable signal is applied to the E1 or Trigger input. This is distinct from the Sweep trigger. Use this method to synchronise the start of sampling with an external event. Sampling starts within 1 or 2 µs of the external event signal. The buttons available are:

Continue — This is labelled Start before data capture starts. It is only enabled when sampling is paused at the end of a sweep. When you click on the button, sampling of the next sweep is enabled. If the Sweep trigger box is checked, the 1401 system waits for the sweep trigger before starting to collect data.

Stop — This is displayed after data capture starts. If you click on this button, the data capture stops, in the same manner as if one of the Automate limits was reached. Once the data capture has been stopped, no more sweeps will be collected, but it is possible to resume sampling again.

Accept — Clicking on this button writes unwritten frame 0 data to disk. If frame zero is still being collected, it overrides the Write to disk at sweep end checkbox so that the frame is written to disk at sweep end, it does not affect subsequent sweeps. If the frame has been collected and sampling is paused this writes the frame to disk immediately. The button acts upon a sweep up until the point that the next sweep is triggered or pulse outputs for the next sweep begins.

Reject — If the Write to disk at sweep end checkbox is set, or sampling is paused at the end of the sweep and frame 0 has already been written to disk then the label on the Accept button changes to Reject. Clicking on the Reject button either overrides the Write to disk at sweep end checkbox to cause the currently sampling frame not to be written automatically, or removes the current frame zero data from the end of the new data file, as appropriate. As for Accept, this button acts upon a frame until the next frame is triggered or pulse outputs for the next frame starts.

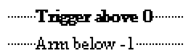
Abort — This button is used to abandon sampling and discard the new file. You can use this button before sampling starts, or while sampling is in progress. You are warned if this will loose saved data.

Restart — This button is available once sampling starts. It stops sampling, discards any saved data, then waits for you to start sampling again with the same document. You are warned if this will loose saved data.

Other interaction with sampling

If a keyboard marker channel is being sampled, you can insert markers into the sampled data by pressing keys on the keyboard. You can do this if the new data view is the current view, or if the sampling control panel is current. The current view or window has a highlighted title bar, you can make a view current by clicking on it. If the sampling control panel is current, you should be careful about pressing the space bar, which is equivalent to pressing whichever button is currently highlighted, or pressing Enter, which is equivalent to pressing the Start/Continue button.

If you are using Peri-triggered sampling in any of the analogue trigger modes, then frame zero of the sampling document will show a pair of horizontal cursors that indicate the positions of the two trigger thresholds. The cursors will be displayed on the last, highest numbered, waveform channel as this is the channel that is used for peri-triggered sampling. For +Analogue and -Analogue trigger modes, the cursors show the trigger level and the trigger level minus (or plus) the hysteresis, this latter level is shown as the Arm level. For =Analogue trigger mode the two separate trigger levels are shown. During sampling you can adjust the trigger levels used by moving the cursors, they cannot be moved to a different channel as the trigger channel is fixed.

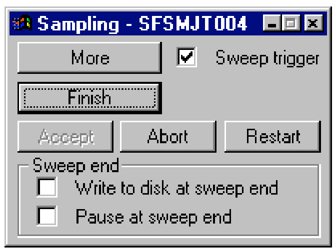


If you use the Display last frame command, or the equivalent shortcut Ctrl PgUp, Signal will switch to displaying the last frame that was saved to disk.

You can tag data frames as they are sampled by pressing Ctrl-T. Most standard display manipulation mechanisms work in exactly the same manner online, but frame overdraw mode works differently on frame zero; it never erases old data but just redraws it in grey and the frame display list is ignored. This provides a very nice 'storage oscilloscope' style display, but if any part of the view is redrawn the previous traces are lost. You can use the Edit menu Clear command to erase all the previous traces.

Stopping sampling

Sampling can be stopped by clicking on the Stop button or by the sampling limits being reached. When sampling is stopped, Signal is in between sampling and finishing sampling. The sampling control panel is still present, but the buttons have changed:



More — This is the button previously labelled as Start or Continue. Click on this button to resume sampling as if the Stop button had not been pressed. If sampling stopped due to the frame count reaching a limit then pressing More will reset the frame count and sampling will run until the count again reaches the limit, otherwise the limit will be disabled and sampling will continue indefinitely.

Finish — This is the button previously labelled as Stop. If you click on this button, the data capture is terminated and the sampling control panel disappears.

Finishing sampling

Click on the Finish button to finish sampling. If the sampling configuration has automatic saving to disk enabled, the new data file will be saved at this point, using either the automatically generated filename or a name entered by the user as appropriate. You will also be prompted for a comment for the new file, if this feature is enabled. Once the sampling has actually shut-down, the sampling control panel is removed. In the new data view, frame zero of the data document disappears and the view changes to show frame 1 if it was previously showing frame zero. If there are no saved frames, the data document and view are destroyed, giving the same effect as pressing Abort.

Saving new data

A sampling document which has stopped sampling is essentially the same as a document loaded from disk. However, unless you are using automatic saving, the data has not yet been saved in a permanent disk file, though it is stored on disk. To keep the data, you must save the new data using the File menu Save command. If you try to close the document view without saving the data Signal will check that you really want to do this.

Data documents differ from all other Signal documents as they are always stored on disk. Other document types are kept in memory until you save them. We keep data documents on disk because they can be very large. When you use the File menu New command, Signal creates a temporary file in the directory specified in the Edit menu Preferences dialog. If you do not specify a directory in the preferences, the location of the temporary file is system dependent. When you save a new data document after sampling, Signal moves it to the disk volume and directory you specify.

Saving configurations

If you had to set up the exact sampling, analysis and screen configuration that you wanted each time you sampled data, life would be very tedious. To avoid this, you can save and load sampling configurations from the File menu. The configuration includes:

The sampling parameters in full, including port information for all ports.

The position of all windows associated with the new file (including duplicated windows and the sampling control panel).

The displayed channels and display modes of the channels in the windows.

The outputs to be generated during sampling.

The multiple states information, including the protocols.

The processing parameters and update modes of all memory views.

Whenever sampling finishes without failing or being aborted by the user, the application saves the configuration used. The configuration is saved so that it is used the next time data is sampled, and also as the file `last.sgc`. When Signal starts, it searches for and loads the configuration file `default.sgc`. If this cannot be found, it uses `last.sgc`. These files are kept in the directory from which Signal was run.

If sampling fails or is aborted, the sampling configuration switches back to the configuration in use before you started sampling.

Remember that you can always recall the configuration that you used most recently, even if you forgot to save it.

Sequence of operations to set and save the configuration

This section describes a sequence of operations that you should follow to build a new sampling configuration from scratch. You will find that once you have built a few configurations, it is much simpler to load an existing configuration and change the sections that do not fit your requirements, rather than re-build entirely. The steps are:

1. Open the Sampling Configuration dialog using the Sampling menu or toolbar and set the sampling configuration, limits and port information.
2. Press Run Now from the configuration dialog or press OK and select the New command in the File menu and choose Data Document.
3. Arrange the new file view as you require and add or remove duplicate windows.
4. Use the Analysis menu New Memory view command to add memory views as required and set their update mode and position on screen.
5. You can use the File menu Save Configuration As command to save the configuration to disk at this point.
6. You can then allow sampling to proceed, adjusting any positions, display configurations and so forth as required.
7. Once sampling has finished without being aborted or failing, the sampling configuration is held in memory for use the next time you sample. You can use the File menu Save Configuration As command to save the configuration used to disk.

Once you have saved a configuration, you can re-use it by loading it (using the File menu Load Configuration command) before you use the File menu New command to create a sampling document.

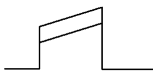
Pulse outputs during sampling

Introduction

In the *Sampling data* chapter you will have encountered pulses as an option within the outputs section of a Signal sampling configuration. Signal can generate outputs from the *1401plus*, Power1401 and micro1401 using the DACs and the digital outputs. This chapter describes the outputs, the pulses configuration dialog and its use during sampling.

Signal pulse output, like Signal data acquisition, is arranged as fixed-length frames. Depending upon the sampling sweep mode, the pulse output frame may be the same length as the sampling sweep, longer, or shorter. In all circumstances, the pulse outputs are fixed in time relative to the sampling sweep. In Basic mode, the pulse output frame starts at the same time as the sweep (triggered or untriggered) and is the same length. In Peri-triggered mode, the pulse output frame starts at the time of the trigger (which can be before or after the start of the sampled data, depending upon pre-trigger points), and again runs to the end of the sampling sweep. In Outputs frame and Fixed interval modes, the pulse output frame can be set to any length greater than or equal to the sampling sweep, and the sampling sweep starts at a defined point after the start of the pulse output frame.

DAC outputs

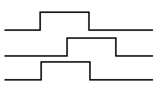


The 1401 DACs (Digital to Analogue Converters) produce varying voltage outputs in the range ±5 Volts, these can be optionally scaled to ±10 Volts if required. DACs can be used to generate pulses with arbitrary initial values and amplitudes (as long as they lie within the DAC output voltage range), they can also generate ramps, sine waves and arbitrary waveforms. Because these are analogue outputs, DAC pulses can have varying amplitudes.

In order to generate complex DAC outputs and particularly for arbitrary waveform output, the DACs must be repeatedly updated with new output values. The frequency with which this is done controls the accuracy with which the output pulses or waveforms are generated, but very high rates of DAC output may interfere with data acquisition. Signal pulse output is limited to a time resolution of 100 microseconds, which is not fast enough to cause any interference with data acquisition.

The 1401*plus* and Power1401 have four DACs, numbered 0 to 3, while the micro1401 has two (0 and 1). Both the 1401*plus* and micro1401 have the DAC outputs available on BNC connectors on the left-hand side of the 1401 front panel. The Power1401 has DACs 0 and 1 here but has DACs 2 and 3 on pins 36 and 37 respectively of the rear-panel analogue connector.

Digital outputs

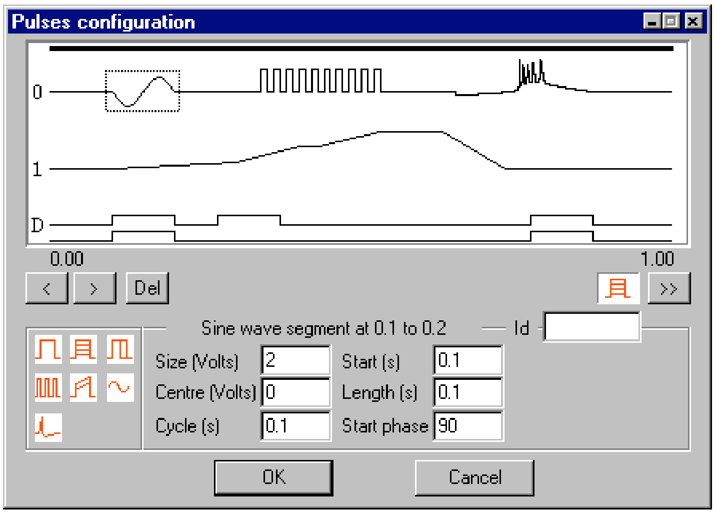


The 1401 digital outputs are TTL-compatible outputs that can be set high or low. When set high they generate a voltage guaranteed to be above 2.6 Volts and not more than 5 Volts, the usual lightly loaded level is about 4.5 Volts. When set low they generate a voltage guaranteed to be less than 0.6 Volts and not below zero volts.

Signal controls 8 1401 digital output lines, bits 8 to 15 of the 16-bit digital output port. These are the dedicated digital outputs in the 1401*plus* and the upper byte of the 16 digital outputs in the Power1401 and micro1401. Signal refers to these outputs as digital outputs bits 0 to 7 throughout for simplicity. The 1401 digital outputs are provided on a 25-way ‘D-type’ socket, this is on the right-hand side of the 1401*plus* front panel and on the rear of the Power1401 and micro1401. On the Power1401 and micro1401 digital outputs 0 and 1 (the labels match Signal usage here) are available on BNC sockets on the front panel, if the Spike2 digital I/O expansion top box is fitted this holds BNC sockets for digital output bits 2 to 7. The arrangement of the digital outputs on the 25-way socket is as follows:

Signal output bit number	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
Digital output bit	15	14	13	12	11	10	9	8	GND
Output socket pin	1	14	2	15	3	16	4	17	13

Pulses dialog



If you press the Configure Pulses button at the bottom of the outputs page in the sampling configuration, Signal displays the Pulses configuration dialog to allow you to view and edit the pulses.

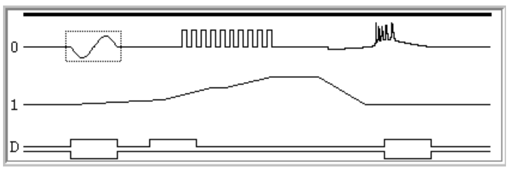
The dialog can also be used to control and adjust the pulses while Signal is sampling, in which case it is accessed using the sampling menu, the toolbar or by using the right mouse button popup menu.

The pulse configuration dialog can be used to define square pulses, pulse trains, ramps, sine waves and arbitrary waveforms that will be output during the sampling. Many of these pulses can automatically vary by incrementing their amplitude or duration by fixed amounts. This provides a straightforward way of generating a repeating set of pulses from one definition. If you are using Outputs frame or Fixed interval sweeps, this dialog is used to set and control the length of the pulse outputs frame, the start of the sampling sweep within the outputs frame and the fixed repeat interval.

Pulses dialog layout

The dialog is divided into four sections:

Display



This occupies the upper part of the dialog and shows the currently defined outputs as graphical traces. At the top of the display is a solid line which shows what portion of the pulse output frame period is covered by the sampling sweep. If you are using Outputs frame or Fixed interval sweeps, this line is thin where no sampling is taking place.

The outputs themselves are displayed starting with DAC zero at the top and finishing with the digital outputs at the bottom, with the lower-numbered digital outputs first. Only traces for enabled outputs are displayed. The output traces are labelled with the DAC number or a D for the digital outputs.

The display also includes a dotted rectangle drawn around one of the pulses, or around the initial level of an output. This indicates the currently selected pulse whose numerical parameters are displayed at the bottom of the dialog (see the Values section below). You can select a pulse by clicking on it with the mouse. Alternatively you can toggle the selection through the various pulses on an output using buttons in the Controls section. You can change the start time of a pulse by dragging it around the display area.

Experiment with clicking on pulses to get a feel for how pulse selection behaves. Note how the display of pulse parameters changes as you select different pulses.

Controls This is the central area of the dialog, which contains a number of controls used to interact with the dialog and displays providing information.

At the top of the controls area at the left-hand side is text such as "0.00" indicating the time for the start of the pulse output frame. The right-hand side holds similar text showing the time for the end of the pulse frame. When a pulse is being dragged about the display, the current pulse position is shown level with these numbers. All of these times are shown in the currently selected time units. Below these time indicators are a number of controls, from the left these are:

< > These buttons are used to toggle the selected pulse through the pulses on the current output trace in forwards or reverse time order.

Del This button is used to delete the currently selected pulse.

State 1 This pair of controls is only visible when multiple states controlling dynamic outputs are in use (see page 47). They are used to select the state that the dialog shows; you can display and edit pulses from only one state at a time.

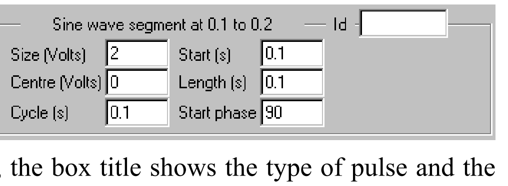
Click on this control to see the effect of pulse variations. While you hold down the mouse button on this control, the limits to pulse variations plus three intermediate values are displayed.

>> Click on this button to animate the display to show the effect of pulse variations. The display updates to show the effect of each variation in turn. Click again on the button to turn animation off.

Pulses This is the area at the bottom left of the dialog. It holds pulse icons which can be dragged into the display area to add a new pulse into the outputs. Each icon represents a different type of pulse:

- a square pulse without variations, digital or DAC.
- a square pulse with varying amplitude, DACs only.
- a square pulse with varying duration, digital or DAC.
- a train of square pulses without variations, digital or DAC.
- a ramp pulse with varying amplitude at start, finish or both, DACs only.
- a sine-wave without variations, DACs 0 and 1 only.
- an arbitrary waveform on multiple DACs, one only of these allowed.

Values This is the area at the bottom right of the dialog. It holds the parameters defining the currently selected pulse. The pulse can be changed by editing the parameter values. The parameters shown vary according to the type of pulse, the box title shows the type of pulse and the start and end times for the pulse, again using the currently selected time units. The details of the parameters are covered in the *Editing the pulse parameters* section below.



Dragging and dropping

A number of operations in the pulse configuration dialog are carried out by dragging and dropping. A major advantage of drag and drop is its graphical, visual nature. For this reason, it is increasingly commonly found as a way of carrying out operations in Windows software.

To drag and drop a screen object place the mouse pointer on top of the object, then press and hold down the left mouse button. The mouse pointer may change to indicate the start of a drag operation and an icon may be attached to the pointer to indicate that the object has been ‘grabbed’. Still holding the mouse button down, move the mouse pointer to drag the object to its required destination. The mouse pointer may change while you do this, a common effect is for the pointer to change to a no entry sign (a circle with a diagonal line through it) to indicate that you cannot drop the object at this point. When the object is at the required destination, release the mouse button to drop it in place.

Adding a new pulse

To add a new pulse into the outputs, select the pulse required from the pulses area of the dialog. Drag the new pulse from the pulses area into the display, the mouse pointer changes to a (a hand with a plus sign) to indicate that you are adding a new pulse. As you move the mouse pointer about the display, a vertical line indicates where the new pulse will go and the drop time is displayed in the control area. Once the pulse is correctly positioned, drop it into place by releasing the mouse button.

If you cannot get the display area to accept the new pulse (the mouse pointer is always No Entry), this could be because of the following reasons:

- You are trying to drag a DAC-only pulse into the digital outputs.
- You are trying to drag a sine wave item into DACs 2 or 3.
- You are trying to drag an arbitrary waveform item into a set of outputs that already contains an arbitrary waveform item, only one of these is allowed.

Moving a pulse

To move a pulse, select the pulse in the display area. Then drag the pulse into the new position required (you cannot drag the pulse to a different output). The drop position is shown as for adding a new pulse, the mouse pointer changes to (a hand) to indicate taking hold of something without addition or removal.

To avoid problems with precise positioning of the mouse, Signal does not recognise a drag-and-drop operation until the mouse has moved a certain amount away from the initial position. The mouse pointer shows this by only including the pulse icon and showing the drop position when the amount of movement is sufficient. You can give up on a move by returning the pointer close to the initial position. If you want to move a pulse a small amount, but find that Signal will not recognise a drag operation that short, move the mouse pointer a larger amount vertically to convince Signal that this is a drag and smaller horizontal movements will be accepted.

If you cannot find or click on the pulse to start dragging it (usually because it is too short or completely hidden by another pulse), see *Finding a pulse* below for how to select it. Once the pulse is selected, you can change the start time directly by editing it in the values area.

Removing a pulse

To remove a pulse, select it in the display area, then drag the pulse out of the display area completely. The mouse pointer changes to (a hand with a minus sign) to indicate a remove operation once you are outside the display area. Drop the pulse to complete the removal.

Alternatively, once the pulse has been selected, you can remove it by clicking the Del button in the controls area. If you cannot find or click on the pulse to start dragging it, see *Finding a pulse* below for how to select it.

Finding a pulse

It may be difficult to see a pulse or to click on it because the pulse is too short to see or because it is hidden by another pulse. Click on the appropriate output trace, then use the < > buttons to toggle through all of the pulses on that output. At the appropriate point, your hidden pulse will be selected and can then be edited or deleted directly.

Experiment with adding, moving and removing pulses to get a feel for how the dialog behaves. Note the different behaviour of overlapped pulses with absolute or relative pulse levels.

Editing the pulse parameters

In addition to changing the pulse start times by dragging and dropping, all of the pulse parameters can be edited by using the values area of the dialog. This area shows all the pulse parameters, it is different for each type of pulse.

Initial level

This specifies the state that the outputs are set to at the beginning of the pulse outputs frame, this item is always present and cannot be deleted or moved. The initial level has a single parameter; Level, that sets the level that the DAC is initially set to. The level entered is scaled before use as defined by the DAC settings in the output page.

The three other parameters at the bottom of the values area; Step change, Repeats and Steps, can be used to define a built-in variation in initial level. Built-in variations are described under *Pulses with variations*, below.

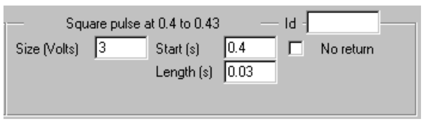
For digital outputs, the parameter changes to Initial bits and is an integer value that sets the initial state of all enabled digital outputs. This value is the sum of a set of codes, each code corresponding to a set bit in the outputs. The values for each bit are:

Bit	Code
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

So, to set digital output bits 2, 3 and 5, you would use the value 44 (4+8+32). There are no built-in variations for the digital outputs.

The Id field in the values area boundary displays the name for the currently selected pulse, which can be edited as desired. The main reason for giving a pulse a name is to make it easy to access the pulse from the script language.

Simple square pulse



This specifies a square pulse without any variations. This is the simplest type of pulse and is available for DACs and for the digital outputs. Many of the parameters used for other types of pulse are also used for this type, to save space these common parameters are described in detail once only.

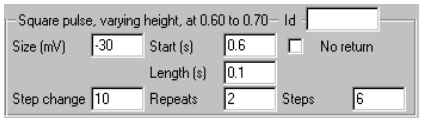
The Size parameter sets the pulse size, in calibrated units as defined in the outputs page. If absolute pulse levels are in use, the item changes to Level, and is the level that the pulse goes to, for relative levels the size is added to the level prior to the pulse to get the pulse level. Either positive or negative values can be used.

For digital outputs, the parameter changes to Pulse bits and is an integer defining which output bits will show the pulse. The value is made up by adding together the codes for the digital output bits, the codes are the same as shown for the initial level. For output bits with a low initial value a high-going pulse is produced and vice-versa. The use or otherwise of absolute pulse levels does not affect digital pulses.

The Start (s) parameter sets the start time for the pulse. If absolute times are not in use, this parameter changes to Delay (s) and sets the delay from the start of the previous pulse to the start of this pulse. The Length (s) parameter sets the length of the pulse. Both of these fields will use the current time units.

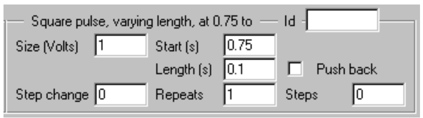
If the No return checkbox is checked, the pulse does not return back to the initial level when it ends but just stays at the pulse level, giving us a single step-change. In this circumstance the length parameter has no effect.

Varying amplitude pulse



This specifies a square pulse whose amplitude varies as it is used. The Size, Start and Length parameters are exactly the same as for the simple pulse. In addition there are three parameters controlling the built-in variation. This type of pulse is not available for digital outputs. The behaviour of the built-in variation is described under *Pulses with variations*, below.

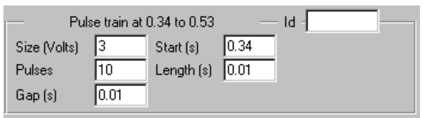
Varying duration pulse



This specifies a pulse whose amplitude is constant, but the pulse length changes as it is used. This is the only pulse with a built-in variation that is available for the digital outputs.

The Size, Start and Length parameters are exactly the same as for the simple pulse. The other parameters control the variation, with the exception of the Push back checkbox. If this is checked, increases in the pulse duration delay the start of following pulses by the same amount. Decreases in the pulse duration move the following pulses earlier in time. If the checkbox is clear, changes in the pulse duration do not affect the time of following pulses.

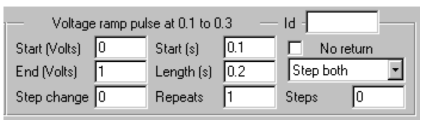
Square pulse train



This specifies a series of non-varying square pulses. This type of pulse is available for DACs and for the digital outputs.

The Size, Start and Length parameters are exactly the same as for the simple pulse. The extra parameters are Pulses, which sets the number of pulses in the train, and Gap(s) which sets the interval between the end of one pulse and the start of the next.

Ramp with varying amplitudes

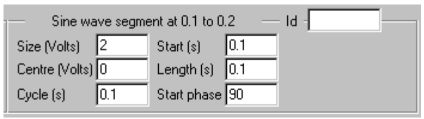


This item specifies a pulse with different start and end amplitudes so that the top of the pulse can be sloping. The variation in amplitude, if this is used, can be applied to either the pulse start or end, or both.

The Start and Length parameters are exactly the same as for the simple pulse, the size parameter is also called Start, but shows the DAC units to avoid confusion. The extra parameters are End, which sets the pulse level at the end of the pulse and a selector for the variation which can be set to Step both, Step start or Step end. This is an extremely versatile form of pulse; for example by setting the start size to zero you can produce a ramp running from one level to another.

Sine wave

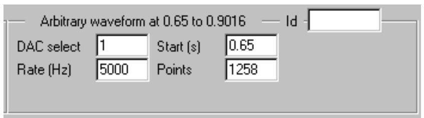
This specifies a cosine wave output of fixed duration, amplitude and frequency, it is only available for DACs 0 and 1.



The Size parameter sets the amplitude of the cosine wave (the distance from the mid-point to extreme). The other parameters are Centre, which sets the level about which the cosine oscillates, Cycle (s), which sets the duration of one complete cycle and Start phase, which sets the initial phase in degrees. The Centre value is an absolute or relative voltage level as required. Because the output is actually a cosine wave an initial phase of 90 degrees will start the output off at the centre level.

Arbitrary waveform

This specifies arrays of data to be output to one or more DACs at a specified rate. Output to each DAC starts simultaneously and consists of the same number of points, so the output also finishes synchronously. The arrays of data can be changed in two ways; by using the Signal script language functions and by copying and pasting Signal data using the Windows clipboard.



The DAC select parameter specifies which DACs are used. Its value is made up by summing codes for each DAC wanted, the code values are shown in the table. Thus for DACs 0 and 1, enter the value 3. The Rate (Hz) parameter sets the output rate for the data points for each DAC and the Points parameter sets the number of data points for each DAC.

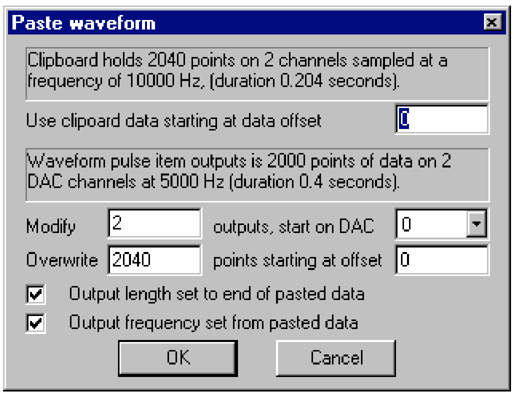
DAC	Code
0	1
1	2
2	4
3	8

The maximum output rates possible vary according to many factors such as the ADC rate and the pulse output timing resolution. Tests carried out at CED with two channels of ADC data sampled at 10 KHz show that waveform rates of 70KHz are possible using a 1401*plus*, and rates of up to 275KHz can be achieved with a micro1401.

Setting a waveform

The Signal data view copy operation places data onto the clipboard in a private format which can be used by the pulse dialog. To place suitable data on the clipboard, open a data file and adjust the display so that the required time range and channels are visible, then use the Edit menu Copy command to place the data onto the clipboard.

If you then open the pulse outputs configuration dialog, select or add a waveform output item and then press Ctrl+V (the Paste command shortcut), Signal recognises this as an attempt to paste data into the waveform buffer. It is difficult to ensure that you have copied exactly the right data onto the clipboard, so Signal provides a dialog to control the paste:

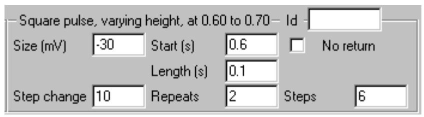


The upper part of the dialog describes the data on the clipboard and provides control of the first data point taken from the clipboard. An offset of zero takes data starting with the first point on the clipboard, larger offsets cause points at the start of the clipboard data to be skipped. The same offset is used for all DACs.

The lower part of the dialog describes the current waveform output parameters, and provides control over what waveform data is changed and how the waveform output is modified to match the clipboard data. The upper pair of controls, making up the line Modify x outputs, start on DAC n, set which channels of the waveform are modified. If changing more than one channel, channels in use starting with the DAC specified are changed. The lower pair, making up the line Overwrite x points, starting at offset y, sets the amount of data pasted and where in the waveform buffer it is put.

The Output length set to end of pasted data and Output frequency set from pasted data checkboxes act as their titles imply. If all data offsets are set to zero, the points to overwrite is set to the points on the clipboard and these two checkboxes are checked, the paste operation copies all the clipboard data and changes the waveform output parameters to match. Click OK to paste the data, or Cancel to do nothing.

Pulses with variations

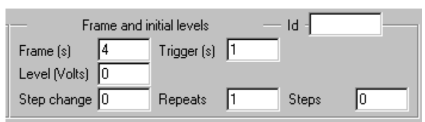


Some types of pulse can be set up so that they vary automatically. The pulse types that support this are initial level, square pulse with varying amplitude, square pulse with varying duration and the ramp pulse. All of these use the same three parameters to control the variation, only the varying aspect depends upon the type of pulse. The pattern of variation used is: the pulse is generated a number of times without variation, then a number of times with one 'step change' added, then two steps and so on. This repeats until the maximum number of changes has been reached, at which point the cycle restarts with the pulse with no step changes.

The Step change parameter sets the amount by which the varied aspect (the pulse amplitude for example) changes at a time. The Repeats parameter sets the number of times each step is repeated before moving on to the next step and the Steps parameter sets the maximum number of changes to be added. This arrangement gives a final value of Initial + (Steps * Step change). The total number of pulse forms generated is one more than Steps as the variation includes the basic pulse without any step changes.

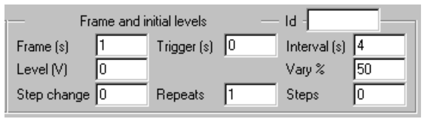
In the example shown above, seven pulses will be generated with amplitudes of -30, -20, -10, 0, 10, 20 and 30 mV. Each pulse will be generated twice in the order shown; the entire sequence will repeat after 14 pulses. If you wanted a sequence that ran 30, 20, 10, 0, -10, -20,and -30, you would set the basic pulse amplitude to 30 and the step change to -10.

Outputs frame and Fixed interval sweeps



With Basic, Peri-triggered, Fast triggers and Fast fixed int sweep modes, the length of the pulse output frame is set by the data frame. In Basic, Fast triggers and Fast fixed int modes, the pulse frame length is the same as the data frame length, in Peri-triggered mode the pulse frame length is the data frame length less any pre-trigger points. With Outputs frame sweep mode in use, the pulse frame length can be set independently of the data frame and the data acquisition sweep starts at a fixed time within the pulse output frame.

These times are set by extra parameters shown with the Initial level data for all outputs. The Frame (s) parameter sets the length of the pulse frame, this value can not be less than the sampling sweep length. The Trigger (s) parameter sets the time, relative to the start of the pulse outputs frame, of the start of the sampling sweep.



With Fixed interval mode in use, two extra parameters; Interval (s), and Vary (%) appear. With Fast fixed int mode the Interval (s) field is available but not the Vary (%). These set the timed interval between pulse output frames, which cannot be less than the pulse output frame length and the percentage variation in the interval, from 0 to 100. If the variation is non-zero, the frame interval used while sampling will vary randomly between Interval-Vary% and Interval+Vary%.

Controlling pulse outputs during sampling

While sampling is in progress, the pulse configuration dialog can be accessed by using the Sampling menu, by clicking on the toolbar button or by right-clicking on spare application areas to get a popup menu. All three mechanisms provide the pulse configuration dialog in the same form as used offline. The only difference is that the OK button has been renamed Apply and the Cancel button Close, and that pressing Apply doesn’t cause the dialog to disappear.

You can use the pulse configuration dialog freely during sampling to change the pulses that are output. All changes made will be copied into the sampling configuration to be used next time if the sampling completes successfully.

With Fixed interval or Fast fixed int mode in use, the use of external sweep triggers is disabled, as the interval timer does all of the triggering. If the delay between the end of one pulse frame and the start of the next is too short, the internal trigger may be missed. If this occurs a warning message is generated at the end of data acquisition.

Sampling with multiple states

Introduction

In the *Getting started* chapter you will have encountered the frame state; an integer value attached to each frame in a Signal data file that can indicate a condition or classification of the frame. You will also have encountered the Enable multiple states checkbox in the General section of the sampling configuration dialog. This chapter describes the uses and control of multiple states in data acquisition.

What can multiple states do?

Let us imagine an experiment involving measuring responses from a test subject, preparation or hardware to two forms of stimulation, for example you are recording the EEG evoked response to a flash of light in either the left eye or the right eye, or maybe vibrations caused by an impact from the right or left-hand side. You have built external equipment to generate these different stimuli and Signal is going to be used to record the result. What you want to do is to generate two averages, one for stimulus A only, and one for stimulus B. The clean way to do this would be to record the state (derived from outputs generated by the external equipment) with each frame indicating whether stimulus A or B was used. Then it would be easy to generate two averages with each average only processing frames with one state value. This is exactly what one form of Signal multiple states can do, by reading the 1401 digital inputs and using the value read to work out the state for each frame. This mode of operation is called *External digital* states in Signal.

Then perhaps you go on to make the experiment more complicated. First of all you want four separate stimuli now (maybe four different tones) and rather than using a fixed ABABAB order of stimuli as you did previously, you want to be able to randomise the order in some experiments. You also don't want to have to design and build external equipment complicated enough to generate randomised sequences, getting it to generate four separate stimuli is quite enough. You can use Signal multiple states for this too. Signal can generate the codes to control the external equipment, write the codes to the 1401 digital outputs, record which codes it is using with each frame as the frame state and optionally randomise the order of the codes. Your analysis is essentially unchanged apart from now producing four averages. This mode of operation is called *Static outputs* states.

Getting enthusiastic, you can dispense with the external equipment entirely. Rather than use Signal to generate codes that select tones from the external equipment, you could set up Signal to generate the tones directly using the sine wave output available from the pulse output system. This would allow you to generate, for example, sixteen different tones without having to extend your external equipment into some sort of electronic instrument. You can make things even more sophisticated by defining patterns of sequences of tones by using a protocol and have multiple protocols available to run through useful patterns when required. This mode of operation is called *Dynamic outputs* states, the protocol is also available with static outputs mode.

These three scenarios briefly describe the three ways that Signal multiple states can be used and touch on some of the ways that states can be sequenced during the experiment. There are many other things that can be done using multiple states, but they all share a common theme; different conditions in the experiment that need to be recorded so that the analysis can distinguish data recorded under the various conditions.

Enabling multiple states

The General page in the sampling configuration dialog contains a checkbox labelled Multiple frame states which is used to enable multiple states in data acquisition. With this checkbox clear, sampling does not use multiple states and all sampled data frames are set to state zero. With this checked, sampling will use multiple states and set the data frames to the appropriate state value.

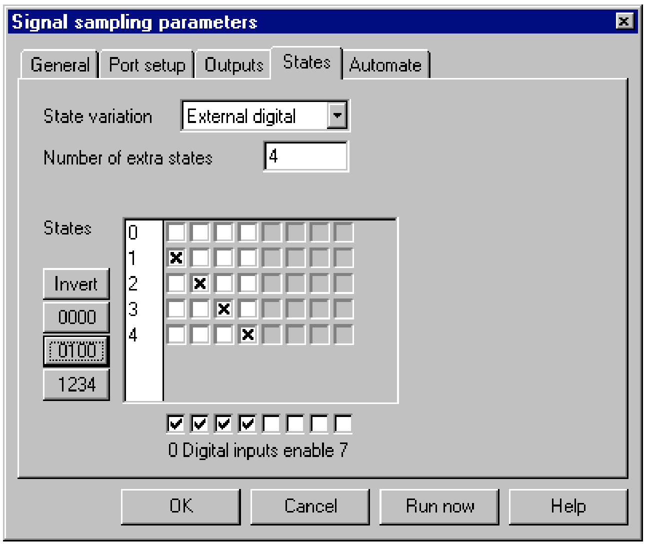
When multiple frame states are enabled, the sampling configuration dialog gains another page labelled States. This page holds controls used to set up multiple states.

Defining multiple states

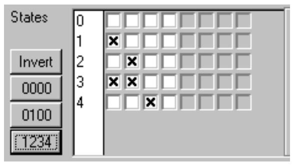
The State variation selector at the top left of the States page select the type of multiple states to use from External digital, Static outputs and Dynamic outputs. The Number of extra states item controls the number of states, it can be set to any value from 1 to 256. Note that this sets the number of extra states, in addition to the basic state zero. In the example with 4 extra states there are 5 states possible running from 0 to 4. In many circumstances Signal would use states 1 to 4 only, reserving state zero as a fallback or idle state. As each type of multiple states is selected, the controls shown in the rest of the dialog change drastically, so the three forms are described separately.

External digital states

External digital states are the simplest form of multiple states. External equipment is used to generate a digital code of up to 8 bits corresponding to the current state of the experiment. Signal reads this code from the 1401 digital inputs and uses it to determine the state for each sampled data frame. There is no software control over the state values and no difference in Signal's behaviour and outputs for the different states.



The rest of the states page defines the mapping from the digital input values to the frame state. The Digital inputs enable checkboxes at the bottom control which inputs are used for this, unchecked inputs are ignored.



The main area in the centre of the dialog defines, for the enabled digital input bits only, the bit patterns that correspond to the various states. The bits are shown in a row, with bit zero on the left. A blank location corresponds to a zero bit, a checked location gives a set bit. When Signal is sampling, it reads the digital inputs at the end of each sampling sweep. The bits read are then checked against the bits for each of the states, starting with state 1 and the frame state is set to the first one that matches. If no match is found then the frame state is set to zero. The bits for state zero are ignored.

Signal refers to these inputs as bits 0 to 7 throughout for simplicity, note that in the 1401 documentation these are referred to as bits 8 to 15 of a 16-bit digital input port. The 1401 digital inputs are provided on a 25-way 'D-type' plug, this is on the right-hand side of the 1401 and 1401*plus* front panel and on the rear of the Power1401 and micro1401. On the micro1401 digital inputs 0 and 1 are available on BNC sockets on the front panel labelled as event inputs 0 and 1, if the Spike2 digital I/O expansion top box is fitted this provides the other six inputs on BNC sockets labelled as events 2 to 7.

The arrangement of the digital inputs on the 25-way plug is as follows:

Signal bit number	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
Digital input bit	15	14	13	12	11	10	9	8	GND
Input plug pin	1	14	2	15	3	16	4	17	13

The buttons to the left of the digital bit controls set or modify the state bit patterns to various useful ways. These are intended to allow simple patterns to be set up quickly and to help to produce more complex ones:

Invert This inverts all of the bits for all states. Useful for converting all zeroes to all ones and a walking one into a walking zero.

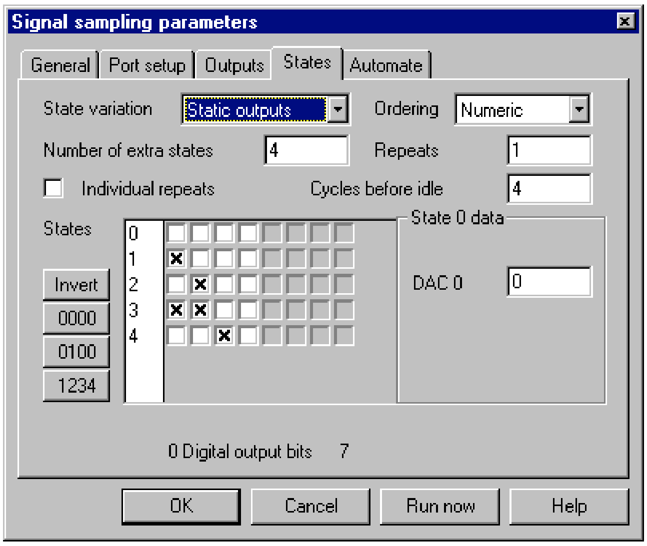
0000 This sets all of the bits for all states to zero. Useful for clearing the control before entering new patterns.

0100 This sets the bits to zero with a walking 1. This leaves state zero all clear, sets bit 0 for state 1, bit 1 for state 2 and so forth. The pattern is not adjusted to skip disabled digital input bits.

1234 This sets the bit values to count the states, using binary code. Thus state 1 has just bit 0 set, state 2 has bit 1 only and state 3 has both bits 0 and 1. Again, the pattern is not adjusted to skip disabled digital input bits.

When sampling using External digital states, Signal behaves much as it does with multiple states disabled. The only difference is that the state value for sampled data frames varies according to the digital inputs.

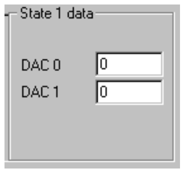
Static outputs states



Static output states are more complicated to set up than external digital states, as Signal controls the state of each sampled data frame. At the start of a sampling sweep, Signal writes values to the enabled DACs and digital outputs to indicate the current state of the experiment, these values could select a stimulus or stimulus level as required.

In addition to defining the values to be output, using static outputs states requires that you also set up how Signal will sequence the states. This requires a number of extra controls in the dialog.

The State variation and Number of extra states items behave just as they did for External digital states. The main box in the centre now defines the values to be written to the digital outputs for each sweep. Because these are now digital outputs, the enables for the bits are in the Outputs configuration section, but otherwise this and the adjacent buttons behave in the same way as for External digital mode. The actual digital outputs and connector pins used are the same as for pulse outputs (see page 35).



The box to the right of digital output bit settings holds controls for the DAC outputs for a selected single state. You can select the state for which values are shown by clicking on the digital outputs line for that state. To avoid confusion, this box is labelled with the state number. The DAC outputs are also enabled and disabled by the outputs section.

States sequencing

Because Signal is controlling the states in this mode (and in Dynamic outputs mode, below), we need ways of setting which state is used when. The simplest way to do this is to manually control the state, but usually something a bit more sophisticated is required. The Ordering selector to the top right of the states page selects the form of automatic state sequencing to be used:

Numeric — In this mode the states are used in numerical order, each state is used the number of times specified by the repeats. Firstly state 1 is used n times, then state 2 and so forth. Once the last state is done, the sequencing stops or restarts with state 1 again as appropriate.

Random — In this mode, one cycle of the sequencing uses each state the number of times specified by the repeats, but the order is randomised. When the sequencing starts another cycle, the order is re-randomised.

Protocol — In this mode, a separate protocol dialog is used to define sequences of states as required. Up to 50 separate sequences can be defined. The protocol data includes repeat information, so the controls for repeats are hidden and replaced by a Protocols… button.

Semi-random — If there are n states then the first n frames will contain one sweep of each state in a random order as will the next n etc. The net result is the same as using Random mode with repeats set to 1 though with Semi-random the cycles are longer.

State repeats

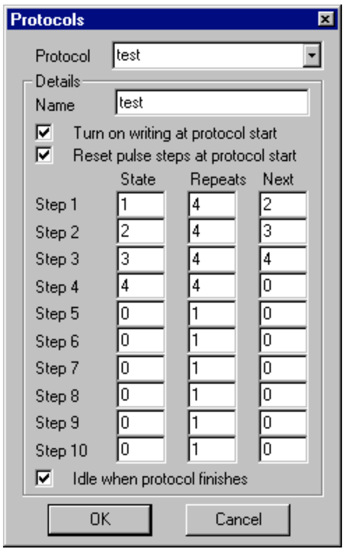
For Numeric and Random ordering the number of repeats must be set. Repeats can be individual, with a separate repeats count for each sweep, or all states can share the same repeat count. If the Individual repeats checkbox is clear, a single control sets the number of repeats for all of the states. If the box is checked, a control above the DAC output values in the state data sets the number of repeats for the state currently selected in the main digital outputs box.

Also for Numeric and Random modes, the Cycles before idle item may be used to force sampling to idle (writing to disk off, manual control of states, state 0) automatically when the ordering has run through the specified number of cycles of states*repeats frames. This allows us to produce either of 112233 or 123123 type patterns before automatically idling. Set this field to zero to disable this automatic idling.

Static output states can only be used with the outputs page type set to None, as otherwise the state values would be overwritten by other output values. When sampling using Static output states Signal behaves rather differently because controls for the states and state sequencing are now provided, see the section *Controlling multiple states online* below.

Protocols

A protocol consists of a list of steps with each step containing a state number, a repeat count and a next step. There can be up to ten steps in a protocol, the next step item can be used to produce a protocol which finishes or one which loops forever.



Protocols are defined using the protocols dialog which is obtained from the states page by pressing the Protocol… button. The dialog contains a selector at the top which is used to select a protocol for editing or to create a new protocol. The area below holds details of the currently selected protocol.

To create a new protocol, type a new name into the protocol selector finishing off by pressing Tab or by clicking on the name field below. The details of the new protocol will be shown.

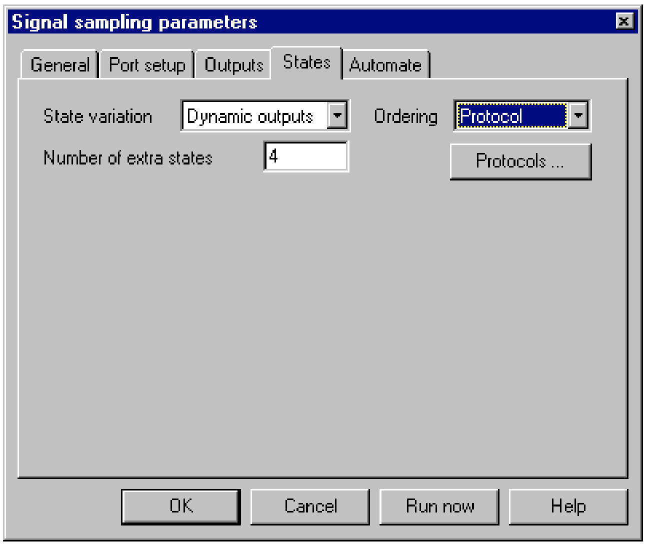
The protocol name, which is used to select the protocol online, is shown at the top of the protocol details. The name can be changed by editing it, to delete the protocol set the name to blank.

The checkboxes at the top of the details control operations that may be wanted at the start of running the protocol; turning on writing to disk and resetting any built-in pulse variations in the pulses output. The checkbox at the bottom enables automatic switching to idling when the protocol finishes.

The rest of the protocol details consists of the ten steps, each step specifies a state, a repeat and a next step. When protocol execution begins it starts with step 1 current. The state specified in the current step is used the number of times specified, then the protocol switches to the specified next step. This process continues until a next step of zero is encountered, which finishes the protocol execution. In the example shown above, states 1 to 4 are each repeated four times, and the protocol then finishes. If the next step for step 4 was set to 1, the protocol would loop back to step 1 so that it cycles forever.

Dynamic outputs states

Dynamic output states are very similar to static outputs, the difference being that each state selects a different set of pulse outputs (or a different sequencer script in later versions of Signal). This makes the states page much simpler in appearance.



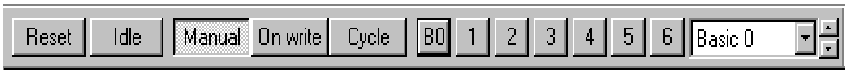
When dynamic outputs states are in use the pulse configuration dialog is modified to include a state selector to select the pulse outputs for a given state.

The number of extra states item is just the same as for the other modes. The controls for setting digital output bits and DAC values are not present, but the main box with a list of states may be present , along with the individual repeats control, so that individual repeat counts can be viewed and edited.

All of the controls for the sequencing of states, the repeats for sequencing and for the protocols are exactly the same as for static outputs. Because dynamic output states are controlled by Signal, controls for the state and state sequencing are provided while sampling.

Controlling multiple states online

When sampling using static or dynamic output states, Signal provides the states control bar to allow you to control the sampling state and states sequencing. The control bar will normally be docked at the top of the Signal application, but can be moved to the bottom or left floating.



The control bar contains a number of buttons and controls:

Reset — Press this button to reset the state sequencing and pulse built-in variations.

Idle — Press this button to cause sampling to idle. It switches to manual control of the states, switches to state zero and turns off writing to disk.

Manual — Press this button to get manual control of the states and disable automatic state sequencing. With manual control, the frame state is controlled interactively using the states control bar.

On write — Press this button to start automatic state sequencing with the states changing if previous data was written. The state sequencer will have control of the frame states and the sequencer will move on to the next state after each sampling sweep only if the sampled sweep was saved to disk. Rejected sweeps will simply repeat until they are accepted. This allows artefact rejection and interactive sweep accept/reject without missing states from the sequence.

Cycle — Press this button to start automatic state sequencing with the states free-running. The state sequencer will have control of the frame states and the sequencer will always move on to the next state after each sampling sweep.

B0, 1 .. — Press these buttons to select a state in manual mode. B0 stands for Basic 0, the basic state. Buttons marked 1 to 9 select the appropriate state, buttons for unused states will be hidden.

State — To the right of the individual state buttons is a state selector and scroller that may be used in manual mode to select any state from those available.

Protocol — If protocol sequencing is in use, the rightmost part of the states control bar holds a selector and scroller used to select a protocol from those available.

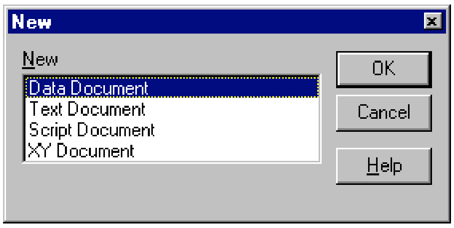
File menu

The File menu is used for operations that are mainly associated with files (opening, closing and creating) and with printing.

The first section of the menu is used for operations that create a new window. The second section closes windows and saves the contents of a window, and also includes commands for saving data under a new name and exporting data to a new data file or a different data format. It can also be used to revert a text-based view to its last saved state. If you have Mail enabled on your system it will include a Send Mail command. The third section holds a control for how changed data is written back to the data file. The fourth section of the menu loads and saves the configuration files that control data sampling. The fifth section controls printing. Both the printer setup and the commands to print a window are found here. The sixth section holds a list of the most recently used files. The final section is the standard way out of any Windows application, the Exit command.

New Ctrl+N

This command creates a new Signal file. This can be a sampling document, an XY file, or a new text-based file. You can activate this command with the Ctrl+N shortcut key or from the menu or toolbar.



The type of the new file is set by the New File dialog, which is automatically provided by this command. You select the type of document to create from the selection provided. You can create four types of document: Data documents which make CFS data files with the standard extension `.cfs`, XY documents which make XY data files with the extension `.sxy`, Script documents which make script files with the extension `.sgs` and Text documents which make text files with the extension `.txt`. Select a document type, click OK and Signal will create a window of the specified type.

Data Document A sampling document window opens plus additional windows as set by the sampling configuration (see page 28 in the *Sampling data* chapter for details). Sampling documents are not initially stored in memory, like most new files, but are kept on disk. Until they are saved after sampling they are temporary files in the directory set in the Signal preferences.

Script Document A script editor window opens in which a new script can be written, run and debugged. A script document is a specialised form of text document.

Text Document Text documents can be used to take notes, build reports and to cut and paste text between other windows and applications. The Log view is a specialised type of text document which is always present.

XY Document XY windows are used to draw user-defined graphs with a wide variety of line and point styles. Although these views can be created interactively, their major use is from the script language. They are also created when generating a Trend Plot.

Other file types used by Signal

In addition to the document types listed above, Signal also uses a number of other types of file:

Resource files

Signal creates resource files with the extension `.sgr`. Each resource file is associated with a data file of the same name but with the extension `.cfs.` The resource files hold configuration information about the data file display so that Signal can restore the display on loading. These files are not essential to Signal and if you delete them the associated data file is not damaged in any way, a new default display configuration will be used.

Configuration files

Signal stores sampling configuration information in files with the extension `.sgc`. These store all of the information needed to carry out sampling: the sampling parameters, the position of the sampling control panel and any other windows, the position and display configuration of the data document window (including any duplicate windows) plus any online processing required, the online processing parameters and the position and display configuration of the memory views showing the results of online processing.

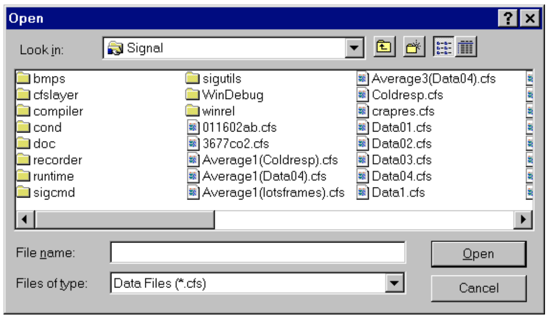
Application preferences

Signal stores its preferences in a file called `cfsview.sgp`. This file holds the position of the application window on the screen, the colour palette and 'use colour' switch and the main preferences information from the preferences dialog. Signal initialises itself with data from this file whenever it is started and saves the current state of the software into this file when it exits.

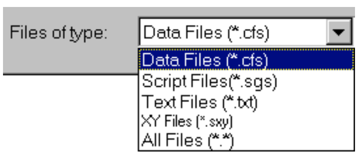
Filterbank files

These files hold descriptions of digital filters and have the extension `.cfb`. They are used by the Analysis menu Digital filters command.

Open Ctrl+O



This command opens a file into a Signal document of any type. You can activate this command with the Ctrl+O shortcut key or from the menu or toolbar. It shows the standard file open dialog for you to select a file. You can open four types of file with this command: a Signal data file with the standard extension `.cfs`, an XY data file with the extension `.sxy`, a text file with the extension `.txt` or a script file with the extension `.sgs`. The type of the file is selected with the Files of type field, if this is set to All files `(*.*)` Signal will try to open the file selected as a CFS file whatever its extension..



When you select a CFS data file, Signal also looks for a file of the same name, but with the file extension `.sgr`. If this is found, the new window displays the file in the same state and screen position as it was put away. Several windows may open if the file was closed with the Close All command. See the Close and Close All commands, below, for more details.

If a read-only CFS data file is opened, you will be warned that the file may not be modified.

If a text file is opened, a simple text edit window is opened. This facility can be used as a notepad or as a repository for data copied from other parts of Signal. If a script or XY file is opened, a script or XY window is created for it.

Import data

Signal can translate data files from other formats into Signal data files. The import data command leads to a standard file open dialog in which you select the file to convert. You then set the file name for the result; Signal will suggest the same name with the extension changed to `.cfs`. The details of the conversion depend on the file type.

Supported formats include the SON files used by CED programs such as Spike2; Spike2 for Macintosh files as well as data from several third party vendors. Signal searches the `import` folder in the Signal installation folder for CED File Converter DLLs. If you need to translate a file format that is not covered, please contact us and describe your requirements. The script language command to convert files is `FileConvert$()`.

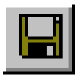
Close

This command is used to close the active window. It is equivalent to double-clicking the control menu icon at the top left of the window (in windows that have one) or pressing the right-hand top corner **x** button (in Windows 95 and NT 4.0 etc). If you use this command on a new memory view, a newly sampled data document or a text-based view with text that has not been saved, a dialog will ask if you wish to save the text before closing the window.

Close All

This command closes the current window and all windows associated with the file. In addition to saving the state of the data file in a `.sgr` resource file (see page 50), Signal offers to save the state and contents of any memory view windows that belong to the data file. Next time you open the data file, all the data view windows and their contents will be restored. If you open saved memory view data then that will be opened as a file view and restored to its previous state as well.

Save Ctrl+S Save As



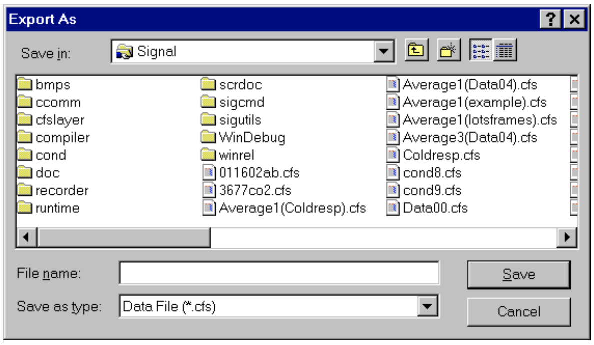
Save will save the current file under its current name, unless it is unnamed, in which case you are prompted for a name before it is saved. Save As is used to save the file with a different name, leaving the original file intact. The Save command is not normally available for a data document unless the data has been changed since the last save.

Data files are kept on disk, not in memory, as they can be very large. Changes made to a data file are permanent as they are made on disk. When you save a newly sampled data file, you are giving the data file on disk a name (replacing a temporary name). If you save it to a different drive from that set in the preferences, Signal copies the file to the new drive, then deletes the original. If the file is large this operation can take several seconds to complete.

When you are working with a file view, the current frame for the view is held in memory, this frame will be discarded when the view switches to a different frame. Any changes made to the frame data while it is held in memory must be saved before the new frame is loaded or the changes will be lost. You can write the changed data back into the file using the Save command. The Preferences dialog allows you to select what happens if the frame is changed while data is unsaved, the default action is to query the user. Changes made to non-channel frame data (such as the frame state or flags) are always saved. Memory views have all frames held in memory and do not save any changes until the entire document is saved.

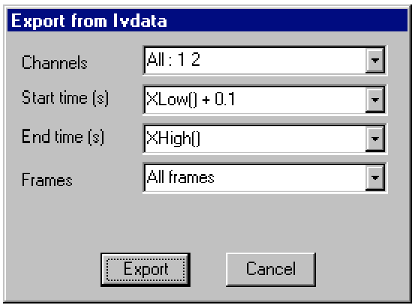
Text, script and XY files are held in memory. Changes made to these files are not permanent until the file is saved to disk.

Export As Ctrl+E



This menu item, available only when the current window is a data view, is used to save some or all of the view data to a new file in one of a number of formats. The dialog prompts you to choose a file name for the output, and lets you select the output format. You can choose between: Data file (*.cfs) to export as a new Signal data file, Text file (*.txt), Metafile (*.wmf) for a scaleable image and Bitmap file (*.bmp) for a copy of the screen rectangle containing the window. Select one of the formats and set a file name, then use the Save button.

Data file This option opens a dialog in which you can create a new data file from a time range and selected channels and frames of the current file.



Use the dialog to select the channels, time range within the frames, and the frames to be exported. You can choose all the channels, individual channels, selected channels or enter a list of channel numbers directly. You can export All frames, the current frame, all tagged or untagged frames, frames with a given state, or you can enter a list of frames.

Once you have selected the channels, frames and a time range, click the Export button to write the data to the new data file.

Text file This is the same as the Edit menu Copy As Text command, but with the output sent to a text file and not the clipboard. See the Copy As Text command for details of the dialogs.

Bitmap file This copies the screen area containing the window to a file. Make sure that the window is completely on the screen and that it is not covered by any other window. You should only use this option when the image you require is an exact copy of the screen. If you need to scale the image, or want to edit it, a Windows Metafile copy is usually better.

Metafile This copies the window as a Windows Metafile. This file format can be scaled without losing any resolution and is usually the preferred format for moving Signal images to drawing programs or into reports.

Revert To Saved

You can use this command with a text file or a script file. The file changes back to the state it was in at the time of the last save to disk.

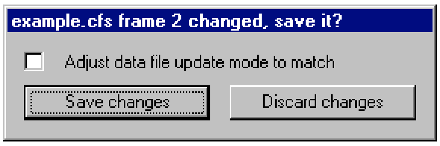
Send Mail

If your system has support for Mail installed (for example Outlook Express), you can send any document from Signal to another linked computer. This command vanishes if you do not have mail support.

Text-based documents, XY views and memory views can be sent, even if they have not been saved to disk (Signal writes them to a temporary file if they have not been saved). Signal data files cannot be sent as Signal keeps them open so they cannot be copied.

Data update mode

This command controls how and if changed file view data is written back to the CFS data file. Signal holds the data for the current frame in memory where it can be accessed and modified by script commands or by the channel data manipulation commands. When the file is closed or the view switches to another frame, the data update mode determines what happens. Any changes to the frame data can be written back to the disk, or the changed data can be discarded, or the user can be asked to choose what is to be done. The Edit menu Preferences dialog sets the default data update mode for all files, use this command to change the mode for a particular file.



If Query the user mode is selected, the Save changes dialog will appear as required. This allows changed frames to be individually discarded or saved, check the Adjust data file update mode to match checkbox if you don't want to see the dialog again.

Load and Save Configuration As

These commands manage Signal configuration files. Configuration files hold the sampling parameters, the window arrangement required during sampling, the output setup and the types of on-line analysis required. Signal always has one configuration loaded, this is the configuration used for sampling and the sampling configuration dialogs.

The Load Configuration and Save Configuration As commands transfer the Signal configuration between disk file and the application. They both open an appropriate file dialog to select a file for loading or saving to.

The Save Default Configuration command saves the current configuration as the default configuration that will be automatically loaded whenever the Signal program is started.

Default configuration files
default.sgc
last.sgc

If the configuration file `default.sgc` exists in the Signal program directory, it is loaded when Signal starts. You can save the current configuration in this file by using the Save Default Configuration command. There is also the standard file `last.sgc` that holds the last configuration that was successfully used for sampling. If `default.sgc` cannot be found, and `last.sgc` exists, `last.sgc` is loaded. Signal saves the current configuration in `last.sgc` each time sampling finishes successfully.

Print Setup

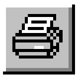
This enters the Windows print setup dialog for the printer. It varies between versions of Windows and the options depend on the printer. See the Windows documentation for more information for your system.

The important options that are always present include the paper orientation (portrait or landscape), the paper source (if your printer has a choice), the printer margins and the choice of printer. Some of these options will be available through buttons, others are visible in the dialog.

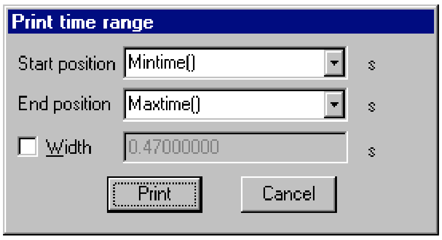
Print preview

This command allows you to view the current document as it would be printed. You can preview file and memory views and text based windows. You can zoom in and out, step through pages of multi-page documents and print using a toolbar at the top of the screen. Use the Close button to leave this mode.

Print visible, Print Ctrl+P Print selection

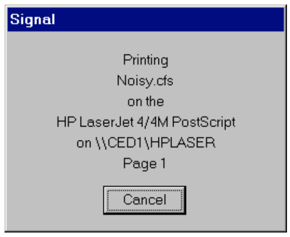


These commands print data views, XY views, cursor windows and text-based windows. The scroll bar at the bottom of the window is not printed. Print selection prints the selected area of a cursor or text window. Print visible prints the visible data in the current window. Print or its Ctrl+P shortcut prints a specified region of a data view at the scaling in the window (one page of paper will hold the same x axis range as the current display, the output will span as many printer pages as are required to show the data selected). You must set the print range in a dialog, either by typing the start and end times directly, or by selecting them from the pop-up menus.



To print an entire frame, move the view to the frame required, set the visible width of the view to the x axis range per page required in the print, then select Print. Select Mintime() for the start position and Maxtime() for the end position. Print doesn't print multiple frames, this will be provided in later versions of Signal.

All the print commands open the standard print dialog for your printer. You can set the print quality you require (the better the quality, the longer the print takes) and you can also go to the Setup page for the printer. Once you have set the desired values, click the Print button to continue or the Cancel button to abandon the print operation.



During the print operation (which can take some time, particularly if you have selected a lot of data) a dialog box appears. If your output spans several pages, the dialog box indicates the number of pages, and the current page so you can gauge progress. If you decide that you didn't want to print, click the Cancel button.

Print screen

This command will print all the file, memory, XY and text based views on the screen to one printer page. The screen is scaled to fill the printer page and the views are scaled to occupy the same proportional positions on the printed page as they do on the screen. The command leads to a dialog in which you can set a page title, choose to print view titles and choose to draw a box round each view. Landscape print orientation often looks better than portrait as the output aspect ratio is closer to that of the screen.

Exit

This command will close all open files and exit from Signal. If there are any data documents or text-based files open containing changes that have not been saved, you will be prompted to save them before the application terminates.

Edit menu

This menu holds the standard Edit functions that all Windows programs provide. The majority of the menu is associated with commands that move data to and from the clipboard.

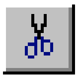
You can also use these commands to search for strings in a text window or to search for the currently selected text.

When the current window is a text-based view the Edit menu operates in the standard Windows manner, allowing you to cut and paste text between text windows in Signal and other applications. When the current window holds a Signal data document, the behaviour is modified.

Undo Ctrl+Z

In a text-based view this is used to undo edit operations. In Signal data views you can undo display scaling operations (for example when you drag a rectangle over a channel to zoom in or out or use the X or Y axis dialogs) and a multiple-level undo list is maintained.

Cut Ctrl+X



You can cut selected portions of editable text to the clipboard from any position in Signal where the text pointer is visible. This includes any text or numeric fields in dialogs and all text-based views. You cannot use this command in Signal data document windows. Once text has been cut to the clipboard it remains there until the next Cut or Copy operation.

Copy Ctrl+C

You can copy selected portions of editable text to the clipboard. If you use this command from a data document window, the contents of the window, less the scroll bar, are copied to the clipboard as text, a bitmap (in two forms) and also as a metafile. It is also copied as binary data which can be pasted into either an XY view or the pulses dialog. See also Copy As Text.

Text output

The visible data is copied to the clipboard as text. The text format used is the same as was last used in Export As with text or the Copy as Text command. See the Copy As Text command for details of the text output format.

Bitmap output

The screen display shown is copied to the clipboard as device-specific and device-independent bitmaps. If, when pasting data into another application you use the Paste Special command, you will be able to select between the two forms. Make sure that the window is completely on the screen and that it is not covered by any other window before copying. You should only use this option when the image you require is an exact copy of the screen, if you need to scale or edit the image Metafile format is usually better.

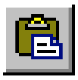
Metafile output

The screen display is copied to the clipboard as a Windows Metafile. You can read an exported image into a drawing program as either a bitmap or as a metafile. Metafiles are often the preferred choice as you can treat the image as lines and text for further changes and adjustments. Use Paste Special and select Picture in the target application to select the metafile image.

CED binary format

The visible waveform data is copied to the clipboard as binary (numbers) using a private CED format. This binary data can be pasted into the arbitrary waveform buffer used by the pulses dialog (see page 40), into another data view or into an XY view. Because this is a private clipboard format, the data is not usable by other applications.

Paste Ctrl+V



You can paste text on the clipboard into any text-based document, or any text or numeric field in a dialog by using the Ctrl+V key combination. Clipboard data using the private CED binary format can be pasted into compatible data views, an XY view or into the arbitrary waveforms output by the pulse dialog. Data is pasted into views within the displayed limits, if the clipboard holds more points or channels that the visible data, only the visible data is modified. This is not true for XY views. If the binary data holds fewer points or channels than the displayed data, only part of the visible data is modified. Pasting from the clipboard does not affect the data stored on the clipboard.

Delete Del

This command is used to delete the current text selection, or if there is no selection, it deletes the character to the right of the text caret. Do not confuse this with Clear, which in a text field is the equivalent of Select All followed by Delete.

Clear

When you are working with editable text, this command will delete it all. If you are in a memory data view, this command will set all the bins in all channels to zero and redraw the window contents. If you are looking at frame zero in sampled data, Clear erases any overdrawn traces. Clear removes everything, Delete removes the current selection.

Copy As Text... (Data view)

This command, available for data document windows only, copies data to the clipboard in text format only. The command leads to dialogs in which you specify the text output format and the data to be copied. Text representations of Signal data can be very large and awkward to manipulate using the clipboard, as an alternative you can write the text output to a file with the File menu Export As command.

Text output format

The first dialog sets the format of the text output. You can enable and disable a header section for the entire output, you can also set the number of decimal places to be used when writing waveform data and time values, the field width and the separators to be used between fields and to delimit text items. For each type of channel you can individually enable the channel data, time, and heading output.

Output fields, separators and string delimiters

All information is written in fields. Each field is either numeric or text. Between each field there is a separator, which can be set to be a Tab character, a blank or a comma using the Separator selector. Most Windows programs that accept tabulated numbers will accept space, a Tab or a comma. The examples below use tab as a separator.

A numeric field holds numbers only, either floating point numbers with a decimal point, or integer numbers. A text field is a sequence of characters that may include spaces. Text fields may hold numbers, but numeric fields cannot hold text. You can mark the start and end of a text field with special characters (usually ") so that a program reading the field can include blanks (spaces) and punctuation within a field without confusion. All keywords are treated as text fields. You can set a one or two character delimiter in the String delimiter field, or leave it blank for none. The examples use " as a delimiter.

Header

The first part of the output is an optional header that displays information about the data file. This header is output if the Add header to text block checkbox is set. The header consists of the file name followed by the frame number. This header is normally disabled as it may interfere with reading exported Signal data into spreadsheets.

```
"Noisy.cfs"    "Frame 4"
```

Waveform data

After the header, if enabled, comes the waveform data. All the waveform data is dumped together, with the first column holding the time values, if enabled, and then one column per waveform channel. Waveform headings provide a single line holding a title for each column of text. Following this are multiple lines with the waveform data times and values:

```
"s"        "ADC 0"    "ADC 1"    "ADC 2"    "ADC 3"
0.23675    1.01074    4.61670    0.46875    0.23438
0.23700    1.37451    4.61182    0.44678    0.27832
.
.
0.33350    -1.69922   -0.18799   0.43945    0.21729
```

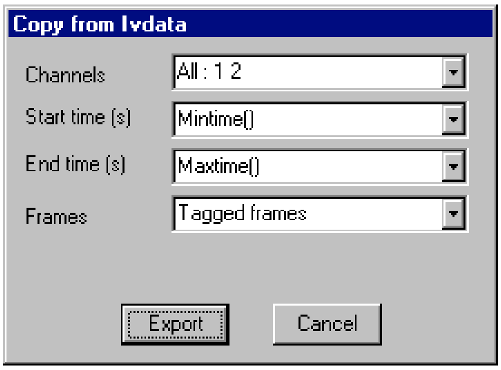
Marker data

Marker data consists of a series of times and 4 bytes of marker information. Unlike waveform data, data from each marker channel is dumped separately. The data dump starts with the channel headings, if these are enabled. The data follows in two columns, a time followed by a character.

```
"s"        "Keyboard"
20.20608   "a"
21.24544   "n"
```

Copy data selection

Once you have defined the output format, you have to select the data to copy. This is done using the same dialog that is used to select the data to be exported to a CFS or text file. You can specify the channels to use, the time range for the data within each frame and the frames. When you are satisfied with the selection, click on Cancel to quit or Export to start the copy process.



Copy As Text (XY view)

This menu item is available in XY views. It copies the XY points for all visible channels to the clipboard. The first line of output for a channel holds “`Channel : cc : nn`” where `cc` is the channel number and `nn` is the number of data points. The data points are output, one per line as the X value followed by the Y value, separated by a tab character.

Find Ctrl+F
Find Again Ctrl+Shift+F

These commands can be used to move the text cursor to the next occurrence of a string when you are working with a text-based view. The search is normally insensitive to the case of characters, if you want to Match Case in a search, check the box.

The search starts at the cursor position and stops at the end of the document unless the Wrap around search box is checked in which case the search will continue from the start of the document after reaching the end. Searches are done line by line (you cannot search for text that spans more than one line). You can also search for *regular expressions*. If you check this box, you can include pattern matching characters in your string. The simplest pattern matching characters are:

- `^` Start-of-line marker. If you include this character it must be at the start of the search text. The following search text will only be matched if it is found at the start of a line.
- `$` End-of-line marker. If you include this character it must come at the end of the search text. The preceding text will only be matched if it is found at the end of a line.
- `.` Matches any character.

To treat these special characters as normal characters with regular expressions enabled, put a backslash before them. A search for “`^\^.\.`” would find all lines with a “^” as the first character, anything as a second character and a period as a third character.

To search for a character list, enclose it in square brackets, for example “`[aeiou]`” will find any vowel. For a character range use a hyphen to link the start and end of the range. For example, to find any alphanumeric character use “`[a-zA-Z0-9]`”. If the search is not case sensitive you can omit one of “`A-Z`” or “`a-z`”. To include the “`-`” character in a search, place it first or last in the list. To search for any character that is not in a list by placing a “`^`” as the first character. To find a non-vowel character use “`[^aeiou]`”.

There are three more special search characters that control how many times to find a particular character. These characters follow the character to search for:

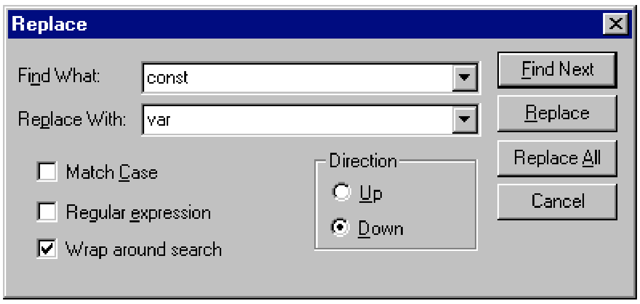
- `*` Match 0 or more of the previous character. So “`51*2`” matches `52`, `512`, `5112`, `51112` and so on. You can also match 0 or more or a character pattern, for example: “`h.*l`” matches `hl`, `hel`, `hail`, `horribl` and “`B[aeiou]*r`” matches `Br`, `Bear`, `Beer`, `Beeeaaaooor` and so on.
- `+` Match 1 or more. The same as “*”, but there must be at least one matching character.
- `?` Match 0 or 1. The same as “*”, but matches one character at most.

The Find Again command repeats the last search with the same options.

Select All Ctrl+A

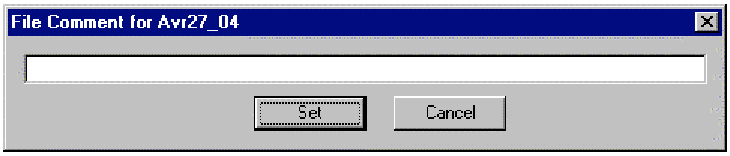
This command is available in text-based views and selects all the text, usually in preparation for a copy to the Clipboard command.

Replace



This command is used to replace text with different text when you are working with a text-based view. Press Find Next to move the text cursor to the next occurrence of the search string, Replace to replace the search string found with the new text and Replace All to find and replace all occurrences of the search text. The search is normally insensitive to the case of characters, if you want to Match Case in a search, check the box. The search starts at the cursor position and stops at the end of the document unless the Wrap around search box is checked in which case the search will continue from the start of the document after reaching the end.

File comment



This command is available with file and memory views, it allows you to see and edit the file comment. The file comment is a single line of text attached to the data file, it can be entered at the end of sampling, or by using this command.

Frame comment

This command is available with file and memory views, it allows you to see and edit the frame comment. The frame comment is a single line of text attached to each frame in the data file which is available for any purpose.

Preferences

The Edit menu Preferences command opens a dialog in which you set items relating to the screen appearance, metafile output, hardware connections and where to store data during sampling. Most of the preferences are stored in the Windows registry and are user specific. If you have several different logons set for your computer, each logon will have its own preferences.

Display

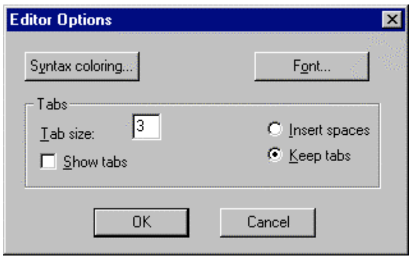
The Display tab contains preferences relating to the way data is shown on the screen

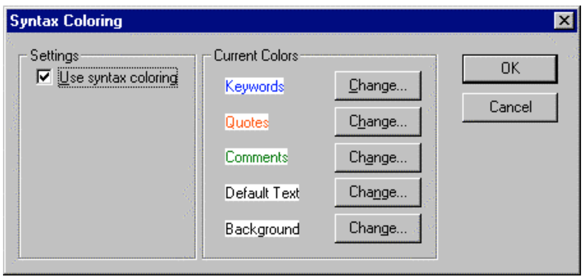
Default font for data views

Signal data views record their font so that they appear the same when re-opened, but they always have a default font when first created . This font is shown in the rectangular area within the default font box, and can be set using the Set Font button, which provides a standard font selector. Once data views are open, their fonts can be individually set using the View menu Font command.

Editor settings

These buttons lead to dialogs for the standard settings for script and text files and to control syntax colouring. The Font… button opens a font dialog that sets the font used when you open a file. You are restricted to fixed pitch fonts in the editor. The Tab size field sets how many spaces to move for each press of the Tab key. You can also choose to keep the tabs as tab characters, or have tabs converted to spaces.





The Syntax colouring… button opens a new dialog in which you can enable and disable syntax colouring for script windows. Enabling it for other windows has no effect. You can also change the colours used for syntax colouring and preview the effect.

Time units The Show time as selector controls the time units used within Signal, you can select Seconds, Milliseconds or Microseconds. The selected units will be used in the sampling configuration, for all data files with a time-based X axis including cursors and cursor windows and for all appropriate dialogs including the various process settings dialogs. The main area of Signal not affected by this setting is the script language, which will always see and use values in seconds, though script programs can read the current settings and adjust their behaviour as required. This setting does not affect any data stored by Signal, just the way time is displayed to the user, so you can switch settings without causing problems with data collected using another setting. Some time values are saved by Signal as strings, particularly the parameters for memory view and XY view processing online and active cursors, and these may be misinterpreted after the time units are changed.

Line widths The two Line width items set the line width in points for drawing data and axes respectively. These are relatively unimportant for displays on the screen, where most reasonable settings will only select between lines one or two pixels wide and many different settings will produce the same display. The line width controls are particularly valuable for printing, where the lines drawn can get unsatisfactorily fine. At CED we find that values of 0.75 pt for data and 1.0 pt for axes look pleasant. The line widths also control various other drawing operations; for example the axis width controls borders drawn around views and the data width sets the basic size of drawn dots and XY view lines and symbols.

Channel order You can change the order of data and memory view channels by clicking and dragging channel numbers. The *Standard display shows the lowest numbered channels at the top* checkbox sets the order when you use the Standard display command or open a new window. If you do not check the box, lower numbered channels are at the bottom.

Data The Data tab controls the way data is stored and exported.

Data export format The data export format items control decisions on how waveform data is written to CFS data files. The Save waveform data as field sets the preferred format for waveform data on disk. Waveform data in CFS files can be stored either as floating point numbers of as scaled integers. Scaled integers are the format of data sampled by the 1401, occupy less space on disk and are the only format that can be read by the DOS SIGAVG software, but are less accurate as data are converted to 16-bit integer values and can overflow. Floating point numbers are more bulky on disk, but are more accurate and cannot overflow. In most circumstances, the format used for waveform data is set by the destination file, or it is forced to scaled integer when sampling data using the 1401, but when creating a new CFS data file by other mechanisms (saving a memory view or exporting to a CFS data file), the format used is controlled by this field. Select Real for maximum data accuracy, Integer to produce smaller data files or for SIGAVG compatibility.

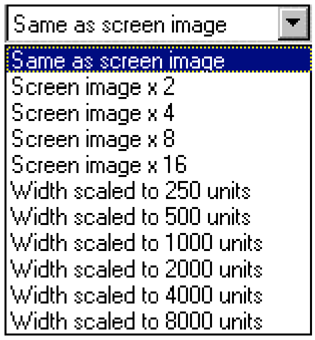
The Keep calibrated zero at zero volts checkbox controls how scaling factors for integer data are calculated. Signal tries to use the same integer scaling factors as the previous values for the frame (or the previous frame where appropriate), but may need to recalculate the factors if the calibrated data becomes too large for the existing scaling or too small to represent accurately. If this checkbox is set, Signal will always calculate scaling factors that keep zero in the integer data corresponding to zero in the scaled values, which can be convenient.

Signal saves the current preferences to disk on exiting and reloads them when it starts up.

Default data update

The Save changed data selector sets the default action for file data that has been changed in memory. For example, the script language could have been used to differentiate the data in a channel. You can choose to write changes back always, to only write changes after querying the user or to always discard changed data. This option sets the default initial state for all data files opened, use the File menu Data update mode command to control data write-back for a single file.

Metafile scaling



Signal saves file, memory or XY views as pictures in either bitmap (screen image) or metafile format. A metafile describes a picture in terms of lines and text based on a grid of points. Metafiles have the advantage that they can be scaled without losing resolution.

You can choose the density of the grid. The higher the density, the more detail in the picture. The problem for time and result views with a lot of data points is that the higher the grid density, the more lines need to be drawn, and many drawing programs have limits on the number of lines they can cope with.

You can set a grid based on the screen resolution, or a grid such that the width of the image is a fixed number of points. If you are not sure what setting to use, start with *Same as screen image* and adjust it as seems appropriate for your use.

Use enhanced Metafile format

Signal supports two metafile formats: Windows Metafile (`WMF`) and Enhanced Metafile (`EMF`). `WMF` is a relic of 16-bit Windows and has limitations, but is widely supported. `EMF` is the standard for 32-bit programs and has many more features. However, some graphics packages do not support this fully (this was written in late 1998, but is still true in 2002).

Prompting to save views

Several users pointed out that it was very irritating to be asked if you want to save data that is derived from other data, and that can easily be derived again. This is especially true when you are developing a new script application. If you check the *Do not prompt me to save unsaved result and XY views* box, Signal will close and throw away result and XY views without requiring a confirmation. As this is potentially destructive, we suggest that you don't use this option when you care about the result or XY data.

Use lines in place of rectangles…

This only affects metafile output. Some graphics programs cannot cope with axes drawn as rectangles; check this box to draw axes as lines. We use rectangles to make sure that axes drawn with pens of other than hairline thickness join up correctly.

Sampling

The Sampling tab contains preferences for the sampling of data.

Directory for new data

When Signal samples data, the new data is stored in a temporary file while it is being collected and only stored in a final CFS file when the file is saved. The Directory for new data field sets the directory in which Signal puts this temporary file. If this field is blank, Signal will use the current directory (which may not be where you expect), so it is a good idea to set one. If you want to save over a network, for example, or store new data files on media that is slow to write, you can use this field to ensure that the temporary file directory (which is where all the real-time writing occurs) is on a fast hard disk.

When you save a new data file, Signal prompts you for the file name. What happens next depends on where you choose to save the file. If the file is on the same volume (disk drive) as the temporary file, Signal just renames the file. If the volume is not the same, Signal copies the file, then deletes the original.

Prompt for file comment

The Prompt for file comment after sampling item is used to encourage entry of a file comment when a new data document has been created by automatically popping-up the file comment entry dialog.

Assume Power1401 hardware

In order to correctly show the sampling rates attainable and limits to the pulse output resolution, Signal needs to know if Power1401 hardware is in use. Signal tries to detect the 1401 type automatically when it starts up, this checkbox sets whether Signal will assume the presence of a Power1401 if it cannot detect the 1401 type directly.

Defer on-line optimise

If a channel display is optimised in the y-direction and the Defer on-line y-axis optimise to sweep end box is checked, Signal will wait until the sweep finishes before scanning the data collected in order to perform the optimise. If this box is unchecked then the optimisation will be done only with the data collected so far in that sweep. This may mean no data has been collected and axes will be set to default limits which may well not be ideal.

Conditioner settings

Programmable signal conditioners (see the *Programmable signal conditioners* chapter) are controlled through communication (serial) ports. Check the *Dump errors…* box to write diagnostic messages to `CEDCOND.LOG` in the current folder.

View menu

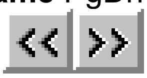
This menu, divided into six regions, is used to control the appearance of the Signal data and XY views. The first region holds commands to show and hide the application toolbar, edit bar and status bar. The second region holds commands used to move the display from one frame to another within the data. These commands are not available for XY views, which only have one frame. The third region holds controls for frame overdraw mode and the frame list used for overdrawing, again not available for XY views. The fourth region holds commands for zooming in and out in both the x and y directions, these commands are also available as short-cut keys, and through mouse actions. The fifth region controls the channels and other items that are displayed on the screen, and the waveform and marker drawing mode. The sixth and final region controls the fonts used when drawing the windows and the use of colour in the data and XY windows.

Toolbar
Edit bar
Status bar

These three commands enable and disable the display of the application toolbar, the edit toolbar and the status bar. The toolbar is the array of buttons normally displayed below the application menu bar. The edit bar is like the tool bar and contains buttons for functions available in text views. The status bar is always at the bottom of the application window and displays information about the current (highlighted) view. These items display a checkmark if the corresponding bar is displayed, click on the item to toggle the display state.

These bars can also be shown and hidden by clicking the right mouse button on an unused area of the Signal application (unused parts of the toolbar area are suitable and always visible). A small popup menu to show and hide the toolbars is provided. This menu also allows other types of toolbar to be shown and hidden. The script and sample bars are described in the script and sample menu chapters respectively. The debug bar is described in the script manual.

Next frame PgUp
Previous frame PgDn



These commands change the current frame to the next or previous frames in the data document. The commands mimic the behaviour of the buttons at the bottom left of the data document window. If the current frame is the first or last frame in the document then the corresponding menu item and button are greyed out. The Ctrl+PgUp and Ctrl+PgDn shortcuts change to the last or first frame in the document, when sampling Ctrl+PgUp switches to showing the last frame filed until the frame is manually changed.

Goto frame Ctrl+G

This command provides a simple dialog used to enter the number of a frame to move to.

Show buffer Ctrl+B

This command toggles between showing the frame buffer and the current frame. When the buffer is shown, the menu item is shown checked, and the view title is modified to show that the buffer is visible. The frame buffer is a separate frame of data ‘behind’ each open CFS file, which is maintained by Signal. See the Analysis menu chapter for more details of the frame buffer.

Overdraw frame list
Ctrl+D

This command switches the display between normal mode, when only the current frame is shown, and overdraw mode, when the current frame is shown along with all frames in the frame display list. When overdraw mode is enabled, the menu item is shown checked.

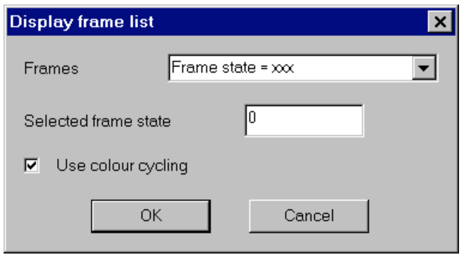
The frame display list specifies frames that are displayed in addition to the current frame. Traces from frames drawn from the list use different colours from the current frame trace. They can either be all drawn in a separate colour, set using the View menu Change Colours command or each frame in the list can have a different colour.

Add frame to list

This command adds the currently displayed frame to the frame display list. If overdraw mode is enabled, all frames in the frame display list are displayed along with the current frame. If the current frame is already in the display list, this command changes to Remove frame from list.

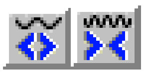
Frame display list…

This command provides a dialog which you can use to directly manipulate the frame display list and also control the manner in which the frames are displayed. The upper part of the dialog allows you to define which frames are in the display list. You can either enter a list of frame numbers directly, or you can use the drop-down list to select a category of frames.



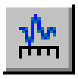
The Use colour cycling checkbox selects drawing of data from each frame in a different colour; Signal contains a fixed table of 18 colours that it uses for this mode of operations. If the checkbox is clear then all the display list frames are drawn using the same colour, this colour can be set by using the colour setup dialog.

Enlarge view
Reduce view

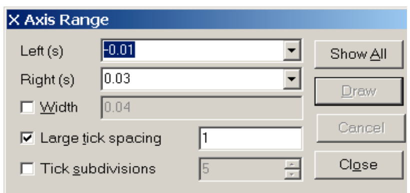


These two commands duplicate the two buttons at the lower left of data windows. The commands are duplicated by keyboard shortcuts Ctrl+cursor right and Ctrl+cursor left. The enlarge command zooms out by doubling the data region spanned by the x axis and the reduce command zooms in by halving the data region. The left hand edge of the screen is fixed unless the expand operation would display data beyond the end of the frame, in which case the displayed area is moved backwards. If the result of an expand would display more data than exists in the frame, all of the frame is displayed.

X Axis range Ctrl+X



This menu command and short-cut key duplicate the action of double clicking on the x axis in a data document or XY view. The dialog lets you select the region of the view to display in the window. You can either type in new positions, or select them from the drop-down lists.



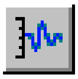
The Left and Right fields set the window start and end. You can type in new positions or use the drop down lists next to each field. The drop down list contains the initial field value, cursor positions, the minimum and maximum allowed values and the left and right edges of the window (`XLow()` and `XHigh()`). The Width field sets the window width if the Width box is checked. Click the Draw button to apply changes in these fields to the window. Show All expands the x axis to display all the data and closes the dialog.

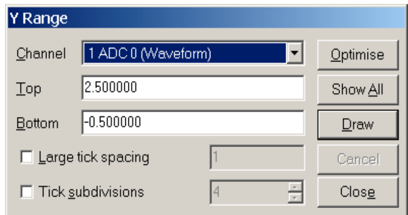
In normal use, you will let Signal organise the x axis style. However, when preparing data for publication you may wish to set the spacing between the major tick marks and the number of tick subdivisions. If you prefer a scale bar to an axis, you can select this in the Show/Hide channel dialog.

You can control the Large tick spacing (this also sets the scale bar size) and the number of Tick subdivisions by ticking the boxes. Your settings are ignored if they would produce an illegible axis. Changes made to these fields take effect immediately; there is no need to use the Draw button.

Cancel undoes any changes made with the dialog and closes it. The Close button closes the dialog.

Y Axis range Ctrl+Y





This dialog sets the range and style of the y axis for channels in time, result or XY views. The Channel field chooses one, all or selected channels (see the *Getting started* chapter for information about selecting channels). When you change the Channel field, the dialog contents update to display the settings for the selected channel. If multiple channels are selected, the settings are for the first channel in the list.

Click Optimise to draw the visible data at the highest magnification without clipping at the top and bottom. Click Show All to set the y axis to display the maximum possible range for waveform channels and from 0 to 10 for marker channels drawn with a frequency axis. Both these buttons close the dialog. You can optimise without opening the dialog with the keyboard short-cut `End` and by right-clicking on a channel and selecting the optimise option from the context menu.

The Draw button applies the y axis range set by the Top and Bottom fields to the channels set by the Channel field.

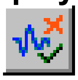
In normal use, you will let Signal organise the y axis style. However, when preparing data for publication you may wish to set the spacing between the major tick marks and the number of tick subdivisions. If you prefer a scale bar to an axis, you can select this in the Show/Hide channel dialog.

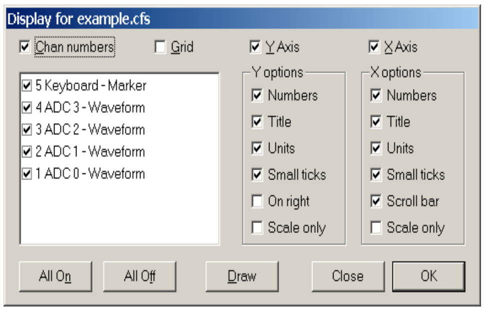
You can control the Large tick spacing (this also sets the scale bar size) and the number of Tick subdivisions by ticking the boxes. These check boxes will be grey if you choose more than one channel and these channels have different tick settings. You can set any tick spacing you like. However, your settings are ignored if they would produce an illegible axis. Changes made to these fields take effect immediately; there is no need to use the Draw button. Cancel undoes any changes and closes the dialog. The Close button closes the dialog.

Standard Display

This menu command sets the current data, memory or XY view to a standard state. In file and memory views it turns on all channels in a standard display mode and size and ordered as set in the Edit menu Preferences, with x and y axes on and grids off. In an XY view, all channels are made visible, the point display mode is set to dot at the standard size, the points are joined and the x and y axis range is set to span the range of the data.

Customise display





This command opens up a dialog that controls the channels that are displayed in a data or XY view and the display of x and y axes, grids and the horizontal scroll bar.

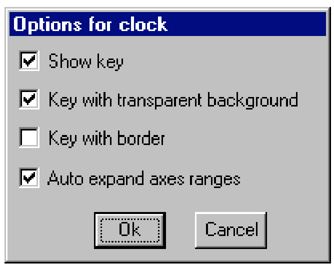
Check the Chan numbers box for channel numbers in time and result windows. The All On and All Off buttons select all or none of the channels, to save time when there is a long list. The Draw button updates the data window. You also have control over the x and y axis appearance. You can hide or display the grid, numbers on the axes, the big and small ticks and the axis title and axis units. You can also choose to show the y axis on the right of the data, rather than on the left.

For publication purposes, it is sometimes preferable to display axes as a scale bar. If you check the Scale only box, a scale bar replaces the selected axis. You can remove the end caps from the scale bar (leaving a line) by clearing the Small ticks check box. The size of the tick bar can set by the Large tick spacing option in the Y Axis Range or X Axis Range dialogs, or you can let Signal choose a suitable size for you.

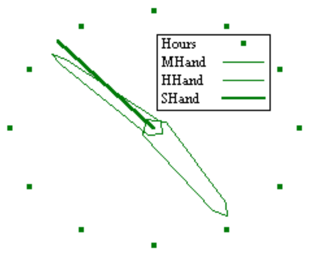
Info Ctrl+I

This command displays information about the current data document. Currently, it just displays the number of sweeps that have been added into a waveform average or the number of data blocks in a power spectrum. The information displayed by this command will be enhanced in later versions.

Options

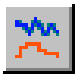


This command is for XY windows only and opens the XY options dialog. The main purpose of this dialog is to control the XY window “key”. The key is a small region that can be dragged around within the XY window that identifies the various data channels within the window. For each visible channel it display the channel name and draws the line and point style for the channel. This dialog also has a checkbox that controls the automatic expansion of the axes when new data is added.

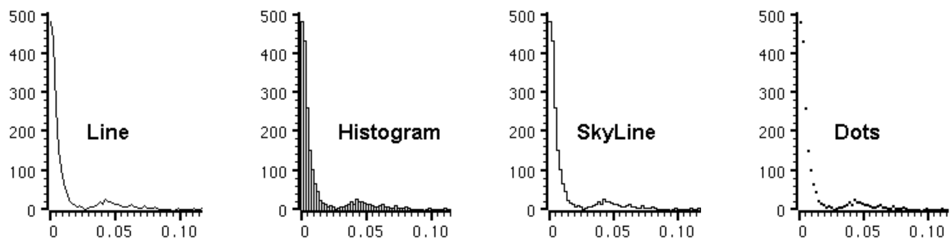


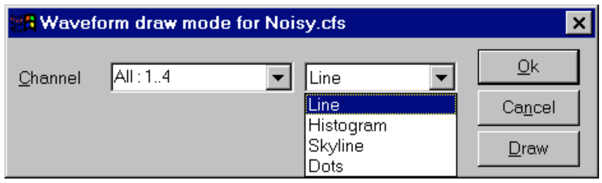
This example (made by the `clock.sgs` script in the `Scripts` folder) shows the key. You can choose to make the key background transparent or opaque and choose to draw a border around the key. If you move the mouse pointer over the key, the pointer changes to show that you can drag the key around the picture. If you double-click the key, the Options dialog is opened.

Waveform draw mode



This menu item is used to set the display mode for waveform data channels. You can set the mode for a single channel, all suitable channels or any subset of the suitable channels. There are four drawing modes defined for waveform channels: Line, Histogram, SkyLine and Dots:





When this command is used the waveform draw mode dialog is provided. This provides a selector for the channels to be affected, plus another selector for the draw mode to use. Press the Draw button to change the draw mode for the selected channels without removing the dialog, click OK to change the draw mode and kill the dialog.

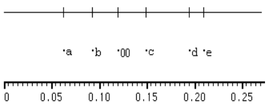
If your view has associated error information, for example a waveform average with error bars enabled, you have an extra control. You can choose from None, 1 SEM, 2 SEM or SD. It should be emphasised that error bars only have meaning if the data points that contribute to the average have a normal distribution about the mean. Given this, then 1 SEM shows ±1 standard error of the mean, 2 SEM is ±2 standard errors of the mean and SD is ± 1 standard deviation.

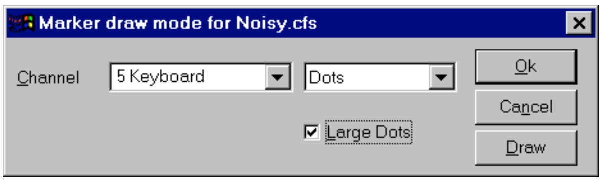
If each point of your data can be modelled as a constant "real" value to which is added normally distributed noise with zero mean, then you would expect the measured mean value to lie within 1 standard error of the mean (SEM) 68% of the time, or within 2 SEM 95% of the time. The standard deviation represents the width of the normal distribution of the underlying data at each data point.

Marker draw mode

This menu item is used to set the display mode for marker data channels. You can set the mode for a single channel, all suitable channels, or any subset of the suitable channels. When this command is used the marker draw mode dialog is provided. This dialog changes depending on the display mode selected, there are three modes available; Dots, Lines and Rate.

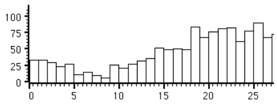
Dots and Lines mode

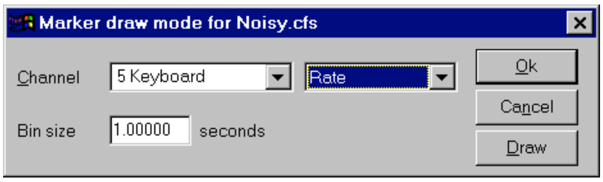




The simplest method is to draw the marker channel as dots. You can choose large or small dots (small dots can be very difficult to see on some displays). You can also select Lines in place of Dots. The picture above left shows the result of both types of display. If you select lines mode the display of marker values is suppressed.

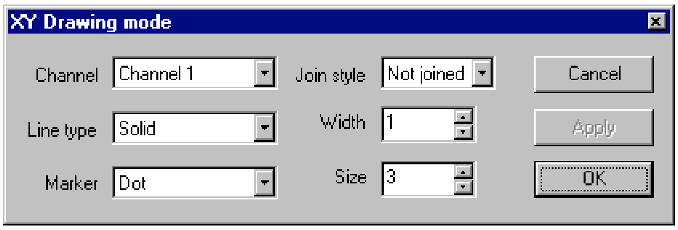
Rate histogram mode





The rate display mode counts how many markers fall in each time period and displays the result as a histogram. The result is not divided by the width of the bin, so strictly speaking it is a count histogram, not a rate. This form of display is especially useful when the marker rate before an operation is to be compared with the rate afterwards.

XY Draw Mode

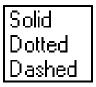


This command is used in XY windows to set the drawing style of the XY data channels. Click on OK to make changes and close the dialog, click on Apply to make changes without closing the dialog. The Cancel button closes the window and ignores any changes made since the last Apply. The Channel field sets the channel to edit. If you change the channel, the dialog remembers any changes you have made so there is no need to use the Apply button before changing channel unless you want to see the change immediately. The other fields are:

Join style

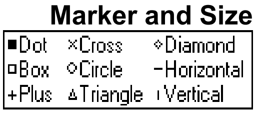
You can set three join styles for a channel. If a channel is Not joined, no lines are drawn between data points. If the channel is Joined, each point is linked to the next by a straight line. If the channel is Looped, the points are joined and the final point is linked back to the first point. The lines joining the points are defined by the Line type and Width fields.

Line type and Width



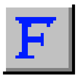
These two fields set the type of line used to join data points. The Width field determines how wide the line is, in units of half the data line width set in the Signal Preferences dialog. If you set a Width of other than 1, the Line type field is ignored and the line is drawn as a solid line.

Marker and Size

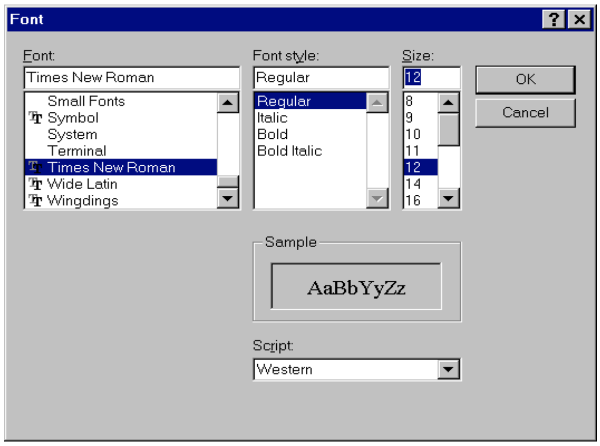


The Size field sets how far, in points, the markers extend around their screen position. A size of 0 makes the markers invisible. There is a wide range of marker styles to choose from, including boxes, circles, triangles and vertical and horizontal bars.

Font



Use this command to select the font that is used for each window in Signal. The dialog provided for font selection is generated by Windows, and will vary in appearance according to the version of Windows you are using.

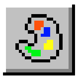


The font size changes the space allocated to data channels in a data view. Smaller fonts give more space to the channels, however fonts need to be large enough to read easily. You can set different fonts in each data or text window.

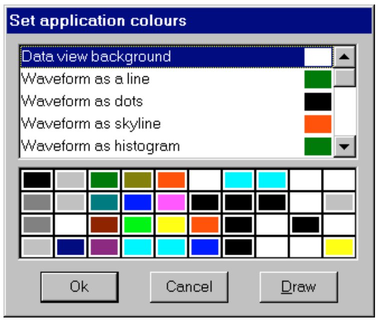
Use Colour and Use Black And White

If you have a colour monitor, you can choose to display your data files in colour. The Use Colour menu item switches from black and white data displays to colour. If you change to colour, the menu item changes to Use To Black And White. You may prefer to work in monochrome if you have to print the end result in black and white.

Change Colours



Use this command to choose the colours used for data document windows. The command displays a dialog with two sections: an upper section in which you can select an item to colour and a colour palette below. The palette is a selection of colours that are available for use within Signal; items are assigned a palette location to use and this palette location then sets the colour of the item.



To change a colour, select an item in the upper part of the menu, then click on a location in the palette. Click on Draw to check the result of your action.

You can change a palette colour by double clicking on it to open a dialog in which you select a replacement colour. The first seven colours in the palette are fixed colours set to various shades of grey and cannot be changed in this manner.

You can replace the palette colour with any standard Windows colour, or you can click the Define Custom Colours... button to select an arbitrary colour. Click OK or Cancel to exit.

The colour selection applies to all data files. It is automatically stored with the Signal preferences information in the file `cfsview.sgp`. When you restore a Signal data file, the colours will be those that are currently active, not those in use when you last saved it.

Keyboard display control

Windows software is usually orientated towards control by means of the mouse and menus, but it is often convenient to use the keyboard instead. For interactive adjustments of the data or display, keyboard control can also be much faster. With this in mind, Signal includes keyboard shortcuts to handle most display manipulation requirements:

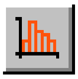
Scroll data down / up	Cursor down / cursor up
Decrease Y range / increase range	Ctrl cursor down / Ctrl cursor up
Optimise Y range	End
Show all Y range	Home
Y axis dialog	Ctrl Y
Scroll left / right	Cursor left / right
Decrease X range / increase range	Ctrl cursor left / Ctrl cursor right
Show all X range	Ctrl Home
X axis range dialog	Ctrl X
Next frame / previous frame	PgUp / PgDn
First frame / last frame	Ctrl PgDn / Ctrl PgUp
Zoom / un-zoom channel	Double-click
Hide selected channels	Del
Customise display	Ctrl Del

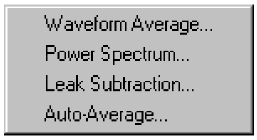
Some of these shortcuts are documented with the appropriate menu command, others do not have an equivalent command. All appropriate shortcuts are listed here for convenience. There are more shortcuts provided for data manipulation; see the Analysis menu chapter.

The Analysis menu creates memory or XY views that hold analysed data from data channels from other data documents and provides access to other analysis functions. The menu is divided into six regions.

The first region holds commands that operate on existing memory or XY views. The second region is used to create new memory or XY views holding data created by built-in Signal analysis mechanisms. It also contains the curve fitting option. The third region provides mechanisms to append new data frames to documents and to delete appended frames. The fourth region provides commands that use the frame buffer, this region also includes the multiple frames dialog which carries out operations on many frames. The fifth region provides channel data modification mechanisms while the sixth region provides control of frame tagging.

New Memory View

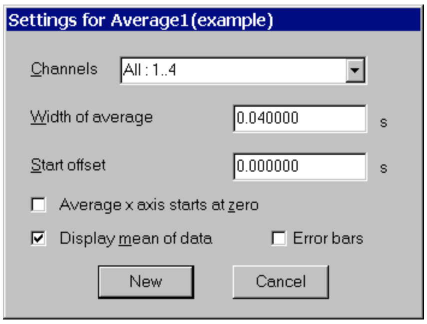




This command is only available when a file view is selected. It provides a pop-up menu from which you can select an analysis type, currently Waveform Average, Power Spectrum, Leak Subtraction and Auto-Average analyses are available. Selecting an analysis leads into a Settings dialog where you define the analysis parameters and other information to construct a memory view. This dialog is also available later on to change the analysis and memory view parameters.

A memory view holds arrays of data that can be treated as frames of CFS data, though they do not become a CFS file until they are saved. Once you have set the required values in the Settings dialog box, a new memory view is created with all data values set to zero. A new Process dialog then appears to prompt you to select frames of the source data to be analysed. The results of analysing different sets of frames can be summed by repeatedly using the Process dialog to select different frames to analyse.

Waveform Average



This analysis averages a waveform across multiple frames. The Settings dialog holds fields for the channels, the width of the average and the data start offset. The Channels field is used to specify the channels to average. Only waveform channels can be averaged. The Width of average field sets the x range in each sweep of the average and therefore the width of the new memory view. The Offset field sets the start time for the data as an offset from the start of the frame, so an offset of zero selects data from the start of the frame, regardless of the frame start time. The data from each frame to be analysed starts at Frame start + Offset and runs up to Frame start + Offset + Width. If this block of data goes past the end of the data in the frame then the sweep is not added into the average and an error message generated.

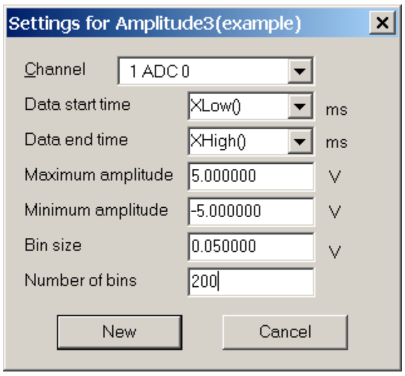
The Average x axis starts at zero checkbox, if set, forces the new memory view x axis to start at zero. If this is clear then the x axis starts at the start time of the first section of data added into the average (i.e. Frame start + Offset for the first frame analysed).

The Display mean of data checkbox selects between displaying the mean data value or the sum of all sweeps added into the average.

If you check the Error bars box, extra information is saved with the result so that you can display the standard deviation and standard error of the mean of the resulting data. The Waveform Draw Mode command controls the display of the error information.

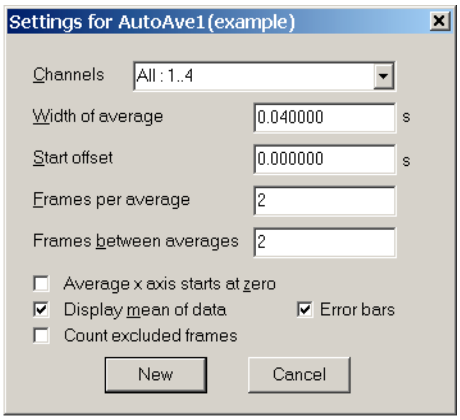
The New button (or Change if this is used from the Process Settings command) closes the dialog, creates the new memory view and opens the Process dialog, described below.

Amplitude Histogram



This analysis creates a memory view is a histogram of the number of data points found at a particular amplitude range of a channel in the file. A time range is given in which the data will be analysed. The Maximum and Minimum amplitudes will determine the start and end points of the x-axis in the memory view. These fields of the dialog are linked to the Bin size and Number of bins. If the amplitude range is changed the Bin size will change to fill the new range and keep the Number of bins constant. If the Bin size is changed then the Number of bins will change to keep the amplitude range constant. Finally if the Number of bins changes then the Bin size will change again keeping the amplitude range constant.

Auto-Average



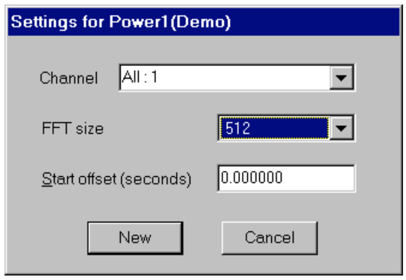
This analysis averages waveforms as for the standard waveform average processing, but automatically produces multiple averages. Each average frame is generated from a fixed number of frames from the source data, the first frame used for each average is offset from the previous average's start by a set number of frames. The Settings dialog holds fields for the channels, the width of the average, the data start offset and the frames and frame start offset per average. The Channels, Width of average and Offset fields are all the same as for the standard waveform average processing described above, as are the Average x axis starts at zero and Display mean of data checkboxes. Count excluded frames is described below.

The Frames per average item sets the number of source frames that are used to make up the first frame in the average. The Frames between averages item sets the number of frames in the source between the start of one average and the start of the next. Thus, if we are using 2 frames to make up each average, setting Frames between averages to 2 will use frames 1 and 2 for the first average, 3 and 4 for the second average and so on. Setting Frames between averages to 4 means that frames 1 and 2 go to average 1, frames 5 and 6 are used for average 2 while frames 3 and 4 are unused. If you set Frames between averages to less than Frames per average, then the data for each will overlap with some frames being used for more than one average.

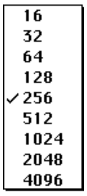
If Count excluded frames is not set and you request for instance to process only un-tagged frames (see below), then Signal will continue to scan the data file for enough untagged frames to form each average. If it is set then tagged frames are not included in the average but do count as if they had been. Thus some of the averages would be formed from less frames. Averages formed from no frames are set to zero.

The New button (or Change if from the Process Settings command) closes the dialog, creates the new memory view and opens the standard Process dialog, described below.

Power Spectrum



This analysis creates a memory view that holds the power spectrum of a section, or sections of data. If multiple sections are processed the result is an averaged power spectrum. The result of the analysis is scaled to RMS power, so it can be converted to energy by multiplying by the time over which the transform was done. There are three fields to set in the dialog: the waveform channels to analyse, the number of points in the Fast Fourier Transform (FFT) used to convert the waveform data into a power spectrum and the start offset within the frame for the data. The channels and offset fields behave the same as their equivalents in the waveform average dialog.



The FFT is a mathematical device that transforms data between a waveform and an equivalent representation as a set of cosine waves, each with an amplitude and relative phase angle. The version of the FFT that we use limits the size of the blocks to be transformed to a power of 2 points in the range 16 to 4096. You set the FFT block size from a pop-up menu. The way the maths works out, the resulting data ends up with half as many bins as the FFT block size. As for waveform averaging, if the block of data starting at the offset specified runs past the end of the frame the sweep is discarded and no analysis is done.

The data in the memory view spans a frequency range from 0 to half the sampling rate of the source waveform channel. The width of each bin is given by the waveform channel sampling rate divided by the FFT block size. Thus the resolution in frequency improves as you increase the block size. However, the resolution in time decreases as you increase the block size as the larger the block, the longer it lasts.

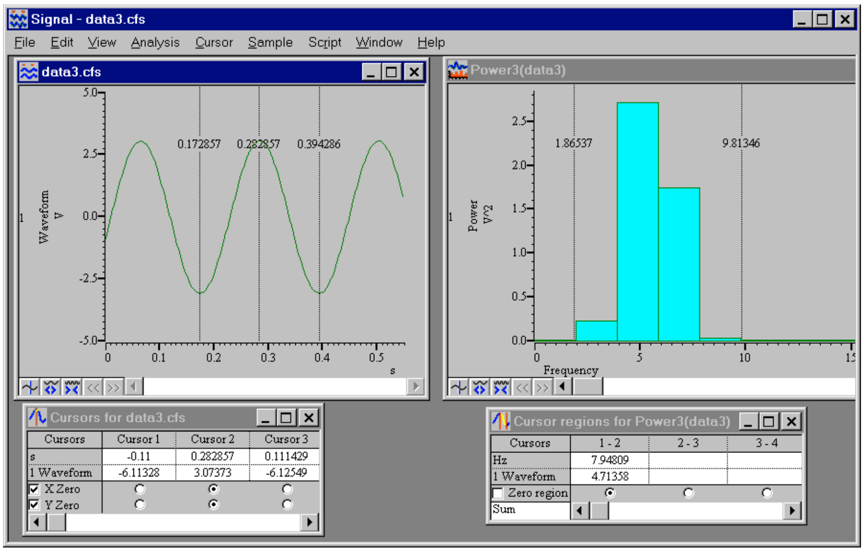
Windowing of data

Another feature of the FFT is that the input waveform is assumed to repeat cyclically. This means that the maths treats the block of data as though it was taken from an input consisting only of that block, repeated over and over again. In most waveforms this is far from the case; if the block were spliced end to end there would be sharp discontinuities between the end of one block and the start of the next. Unless something is done to prevent it, these sharp discontinuities cause high frequency components in the result.

The standard solution to this problem is to taper each data block to zero at the start and end, so that the start and end join smoothly. This is known as *windowing* and the mathematical function used to smooth the data is called the *window function*. The use of a window function causes smearing of the data, and also loss of power in the result. A discussion of the relative merits of different window functions is beyond the scope of this manual. We use a raised cosine window and compensate for the loss of power it causes.

Power spectrum of a sine wave

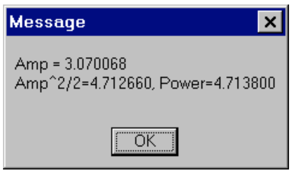
If you sample a pure sine wave of amplitude 1 volt and take the power spectrum, you will not get all the power in a single bin. You will find data spread over three bins, and the sum of the three bins will be 0.5 volt^2. The factor of 2 in the power is because we give the result as RMS (root mean square) power. This is illustrated by the example below where we have sampled a sine wave with amplitude 3.06 volts (peak to peak amplitude = 6.12). We have formed the power spectrum of the signal using a 256 point transform and zoomed in around the bins where the result lies.



If the sampled data was a perfect sine wave we would predict a RMS power of 4.6818 volts^2 from this waveform ($3.06^2/2$). The cursor analysis of the power shows a total power of 4.7135 volts^2. This is about 0.7% above the predicted result for perfect data.

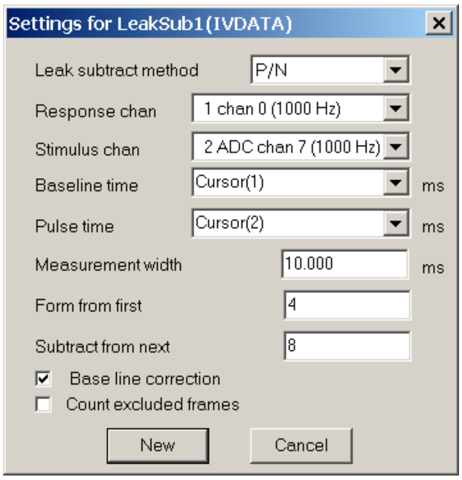
The predicted result is slightly low because the waveform samples used for the cursor measurements are unlikely to lie at the exact peak and trough of any particular cycle. Using the script language `Minmax()` function on a waveform channel to find the maximum and minimum values over a wide time range gives a slightly larger amplitude, and a much closer agreement:

You can use the Evaluate command in the Script menu if you want to try this. It gives a slightly larger value for the amplitude and now the power calculated from the amplitude and the measured power differ by 0.025%. For an explanation of the text in the Evaluate window see *The Signal script language* manual.



The duration of one cycle of the waveform (the time between cursor 1 and cursor 3) is approximately 0.2214 seconds, a frequency of 4.52 Hz. Again, this is in agreement with the displayed power spectrum.

Leak Subtraction



This analysis creates a multi-frame memory view by carrying out a leak subtraction analysis on the source data. Leak subtraction is a specialised analysis used by Voltage and Patch clamp researchers. The basic technique is to use a small stimulus, one that does not cause the cell membrane ion channels to turn on, and measure the current flow through the membrane impedance (made up from resistive and capacitive components). This 'leak' measurement is then scaled to give the expected non-ionic conductance during a larger pulse and subtracted from the recorded traces to leave only the ion-channel effects. Normally an average leak trace is assembled from a number of small pulses, to minimise the effects of noise, but this makes no substantial difference to the technique. Leak subtraction makes special use of two channels; the stimulus channel which is used to measure the amplitude of the pulse so it can be scaled, but is not modified by the analysis, and the response channel which is the only channel modified by the leak subtraction process. All other channels are ignored and copied unchanged.

The Settings dialog holds fields for the leak subtraction mode and the channels for the stimulus and the response to be corrected. The baseline time is a time within the sweep where there is no stimulus and the pulse time is a time where there is a stimulus. These are used to measure the stimulus amplitude, the level measurements are averaged over the requested width. The last two edit fields specify the frames to use to calculate the theoretical scaled trace and which frames to subtract the leak from. These fields are interpreted in different ways depending on the leak subtraction method described below. If base line correction is on, the corrected response will also have a DC offset removed so that at the baseline time the response is unchanged.

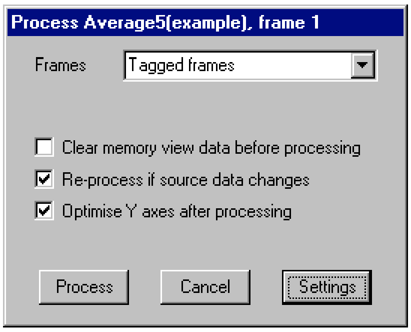
The leak subtraction method can be set to Basic, P/N, or States. These three modes are very similar, the only real difference is how the leak trace is assembled. In Basic mode, the leak data is assembled from a fixed contiguous set of frames, these frames are set in the dialog by entering a first and last frame. All the frames processed use this set of leak data, frames contributing to the leak data are automatically skipped. In P/N mode, the frames processed provide the leak traces, the first n frames processed are used to make the leak, then the next m frames are processed using the leak, this cycle continues until all frames are done. The values for n and m are entered in the settings dialog. In States mode, the leak data is assembled from all frames with a given state, the state number is entered in the settings dialog. All the frames processed use this set of leak data, frames contributing to the leak data are automatically skipped.

As for Auto-Average if, for instance, we only process un-tagged frames and there are some tagged frames in the file, the process will continue to search for un-tagged frames until the required number have been found to complete the formation or subtraction of the leak. If Count excluded frames is turned on, however, then even frames which are excluded from the process will count in the total number of frames used.

The New button (or Change if this is used from the Process Settings command) closes the dialog, creates the new memory view and opens the Process dialog, described below. Leak subtraction processing uses the same process dialog, but because leak subtraction creates a set of frames in the memory view the behaviour is subtly different; the Clear bins checkbox, if set, clears out the entire memory view and if not set doesn't accumulate data into the existing frames but rather appends more frames to the memory view. For similar reasons the Analysis menu Append Frame command does not create a second set of process parameters, all frames use the same process parameters.

Process…

This command is available when a memory view created using the New Memory View command or a similar XY view is the current view. When you use it a dialog prompts you to select the frames of the source data document to process. The Process dialog is also provided automatically when you use the New or Change button from the Process Settings dialog to create or rebuild a memory view.



The simplest way to use this dialog is to type in a frame list directly. You can also select the current frame, all frames, tagged or untagged frames or frames with a given state code. If you choose the state code option, the dialog displays a field into which you can enter the state code to use.

The frame list values are evaluated when the Process button is used, the frames are processed and the results added into the memory view data. The dialog window will remain on screen until removed with Cancel. This means that you can set the frames to Tagged then adjust the frame tagging in the source data document and click the Process button to analyse the selected frames.

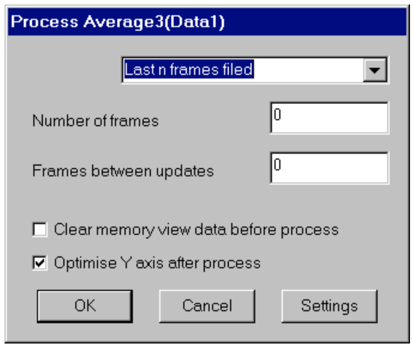
If you check the Clear memory view before process checkbox, the memory view data will be cleared before the results of the processing are added. The Reprocess if source data changes checkbox enables automatic re-processing. Automatic re-processing is optimised to try to prevent unnecessary work, but can still slow Signal significantly on occasion, particularly with large files. If you check the Optimise Y axis after process checkbox, the memory view y axes will be re-scaled after the new results are added into the memory view to best display the data.

If the Settings button is pressed, the process dialog is replaced by the settings dialog.

Breaking out of Process

Processing operations can take quite a time, especially in large data documents. You can stop a processing operation early with the Esc key.

Process command with a new file



The Process command behaves slightly differently when the current window is a memory view derived from a sampling document. The command activates a modified version of the Process dialog. This dialog is also activated automatically when you create a new memory view from a sampling document or when you press the Change button in the Process settings dialog for a similar memory view.

This form of the Process dialog gives you control over when and how the memory view is updated during sampling. The main frame selection control contains extra items that are suitable for processing sampled data: Sampled frames and Last n frames filed. The contents of the dialog changes depending upon which frame option is selected. In addition there is a new field: Frames between updates.

Sampled frames	all frames that are sampled will be processed. This option is not available if you are using Fast triggers or Fast fixed int sampling modes.
All filed frames	all frames saved to disk are processed.
Last n frames filed	process the most recent frames saved to disk, the dialog displays a field in which you can enter the number of frames required. The Clear memory view checkbox is ignored as the memory view is always cleared before processing.

The Frames between updates field sets how often the processing of filed frames occurs. Set this to zero to process as often as possible. If you are processing Sampled frames, then this field is ignored and the memory view is updated for each frame.

Process settings…

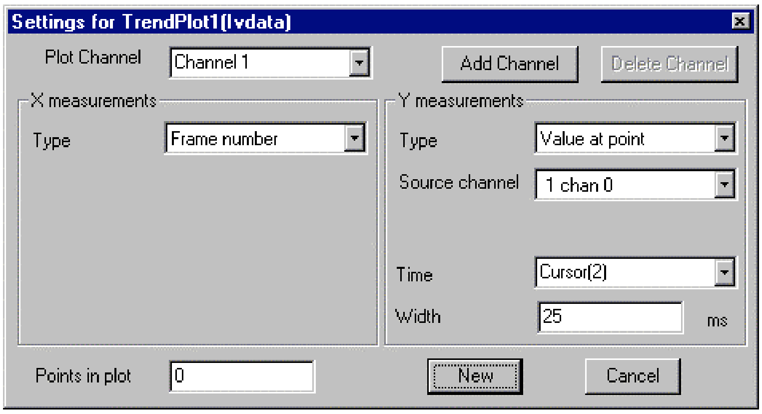
This menu command opens the analysis settings dialog for the current memory or XY view. This is the same dialog as the one used to define and create the new view except that the New button is now a Change button. The Change button accepts the changed settings and rebuilds and clears the memory view.

New XY View

This command, analogous to New Memory View, is available when a file or memory view is selected. It provides a pop-up menu from which you can select an analysis type, currently Trend plot analysis is the only one available.

Trend plot

Trend plots consist of sets of measurements taken from a source data document and plotted into an XY view. Two measurements are taken from each frame; one measurement generates the X part of the XY data, the other generates the Y data. Selecting Trend plot analysis leads into a Settings dialog where you define the analysis parameters and other information to construct the XY view. This dialog is also available later on to change the analysis and view parameters. Once you have set the required values in the Settings dialog the new XY view is created and the standard Process dialogs can be used to control the analysis.



An XY view holds lists of XY points, one list per channel. Trend plot analysis can create data points using a wide variety of types of measurement for both the X and Y values, up to thirty-two channels of XY data can be created. When the command is used the Trend plot settings dialog is provided to allow you to enter the trend plot settings.

The dialog consists of four regions. At the top is a control to select a channel and buttons to add and delete channels, you cannot delete the last channel. In the middle are two similar regions setting the parameters for the X and Y measurements, and at the bottom are the standard buttons plus a control for the points per channel.

The Plot channel selector and the Add and Delete channel buttons all act much as expected, note though that the channel selector can be edited to change the name of the currently selected channel.

The two measurements sections are the same. Both hold a selector for the channel to take measurements from, another for the type of measurement, items for the one or two time values needed for the measurements and a measurement width item. The types of measurement available are:

Value at point	the value read from the channel at the time specified using the specified width.
Value difference	the difference between the value read from the channel at the time specified and the value at the reference time, both using the specified width.
Time at point	the time specified. Note that this can be a cursor position and, if that cursor mode is set up to move to a feature, this gives a measurement of the time of a feature.
Time difference	the difference between the time specified and the reference time. Note that either of these can be a cursor position which can vary.
Frame number	the frame number. This is often used as the X measurement to give a plot of 'measurement against frame'.
Absolute frame time	the absolute start time of the frame. This can be used as an alternative to the frame number to give 'measurement against time'.
Frame state value	the frame state value.
Fit coefficients	the resulting coefficients from a fit (see below). The coefficient index should be specified.
User entered value	a value entered by the user. A dialog will be provided for the value to be entered.

Cursor regions — any of the measurements available for the cursor regions window can be used (see page 91).

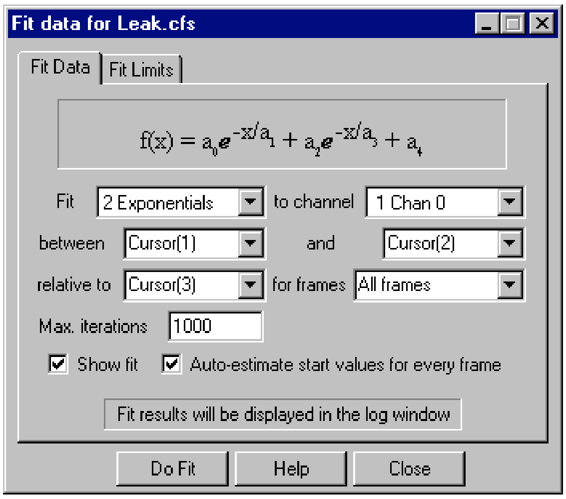
The channel selector and time entry fields are as standard for Signal, and should be easy to use. Don't forget that, in addition to entering a time value directly or selecting an item such as "Cursor(2)" or "XLow()", you can apply an offset to these selected value allowing you to enter "Cursor(2) - 0.1" or "XLow() + 1".

The Points in plot field at the bottom allows you to specify the number of points this channel can contain before old points are deleted to make way for new. Set this field to zero to allow all points to accumulate.

The New button (or Change if this is used from the Process Settings command) closes the dialog, creates the new XY view and opens the Process dialog, described above. Processing for trend plots is very similar to standard memory view processing; the frames specified are used to generate measurements which are added to the view data. If the Clear XY view data before processing checkbox is checked, all of the data points in the XY view will be deleted first. In addition to the Y axis optimisation control, there is an extra checkbox for view X axis optimisation after processing.

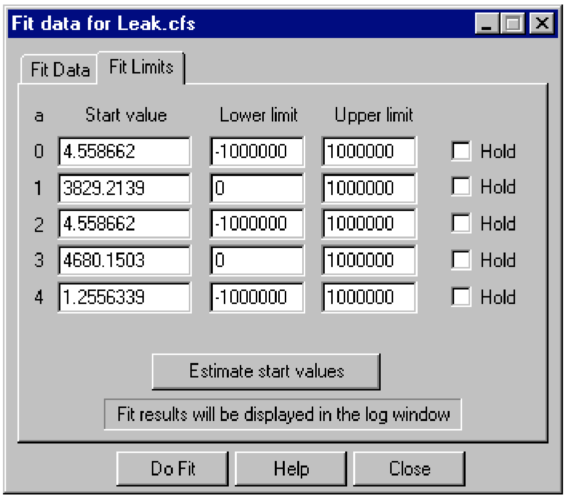
Fit data

This command allows theoretical functions to be fitted to waveform data in a file or memory view. Fitting is done using an iterative process described in detail in the script manual.

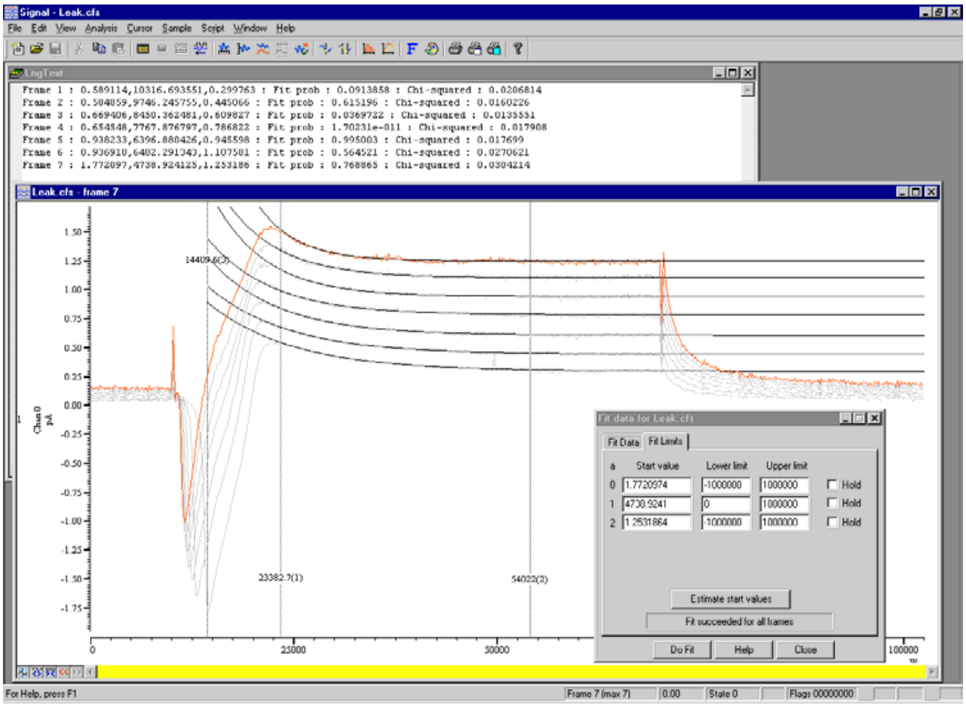


This is a tabbed dialog with the first tab being Fit Data. The upper portion of this page of the dialog displays a template of the function to be fitted. This can be chosen from selector next to the word "Fit". This is followed by a selector for the waveform channel containing the data to be fitted. Two more selectors specify the time range for the fit to take place. The "relative to" field specifies the point on the X axis to be taken as zero. This is mainly useful for exponentials as it affects their amplitudes. The function may be fitted to a range of frames and these are specified in the final selector. The fitting process will continue until the sum of the squares of the errors between the observed data and the function are minimised. Under certain circumstances this may never happen and so a limit on the number of iterations is required. Finally there are a couple of check boxes which should be used if you would like the fitted curve to be drawn over the raw data or if you would like Signal to automatically estimate the start values for fitting each frame. Below these there is an area which displays additional information about the fit.

The second page of the dialog is Fit Limits. This allows the start values for the fit to be set as well as the upper and lower limits of the fit search. Setting good initial values and a reasonable range in which the fit can vary each parameter will greatly increase the chances of a successful fit. To this end a Guess start values button is provided which will usually provide reasonable initial parameters for the fit. The Hold check boxes provide a way of stopping particular parameters from changing during the fitting process.



Each line of output in the log window consists of the information for each frame fitted. After the frame number is the list of fitted parameters. This is followed by a fit probability and a chi squared value. The chi-squared value is the sum of the squares of the errors between the raw data and the fitted curve. Since the variance on the measured data is not known the chi-squared value is un-normalised.



The fit probability is a more complex calculation. In order to calculate this an independent estimate of the value of chi-squared value is needed (i.e. not calculated from the errors). This is done by first calculating the standard deviation of the differences between consecutive errors. If the errors are normally distributed with a mean of zero then the variance of the errors should be twice that of the differences between consecutive errors. If we then assume that the variance of each data point is the same we can use this estimated variance to calculate a normalised value for chi-squared. The probability of the fit is then calculated as the probability of getting a chi-squared value less that this. This is calculated using the incomplete gamma function. Further details of this can be found in *The Signal script language* manual. If there is a problem fitting the function to the data then this will be reported instead of the fit parameters etc. The possible error messages are:

Fit matrix is singular — This can happen if the initial guess is a very long way out or the function does not suit the data.

Fit unable to progress — The fit is as good as it is going to get. Better start values may help.

Max iterations reached — This could be cured by a better initial guess or by fitting over a slightly different area of data.

Append frame

This command appends a new, blank, frame to the end of a file or memory view. This command can be used on a file view to generate an extra frame that will be used to hold processed data; for example a frame containing leak subtraction data that will be subtracted from other frames in the file. The extra frame can be used as part of a script, or it can be manipulated via the frame buffer.

If the command is used to append a frame to a memory view created by processing, the new frame will have its own processing parameters. When the frame is appended, the process dialog is provided to define the new frame's processing parameters. This allows for result views with different frames holding the results of processing different sets of source frames. For example, if you were sampling with multiple states, you might want to produce a multiple frame average with each frame holding the results of averaging source frames with a different state. Multiple-frame memory views, with attached processing parameters, can be saved as part of a Signal sampling configuration.

Append frame copy

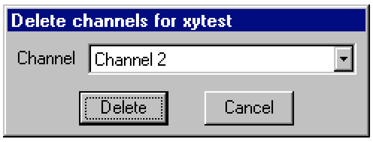
This command appends a new frame, containing a copy of the data in the current frame, to the end of a file or memory view. This command is not available for memory views created by processing.

Delete frame

This command removes the current frame from the file or memory view. It is only available if the current frame has been appended and not yet written to disk. The last frame in a memory view and file view frames stored on disk may not be deleted.

Delete channel

This command, available only for XY views, is used to remove a channel permanently from an XY data document. Once the channel has been deleted, it cannot be retrieved.



The frame buffer

The frame buffer is an extra frame of data that is automatically provided by Signal. Every open data file and memory document has a separate frame buffer that is used in conjunction with the document data, this buffer is shared by all of the views of that document. The frame buffer can be used to carry out arithmetic on frames (for example, subtract the average of frame 1 to 4 from all frames), either interactively via the commands described below or by using the multiple frames dialog, also described below.

The way to think of the frame buffer is as an extra frame of data that is behind the current frame in the view. When you change to a different current frame, the buffer moves too so that it is always associated with the current frame. Understanding this association is important because all of the frame buffer arithmetic commands described below work with the current frame. If you are displaying the frame buffer the buffer moves so that it is in front of the current frame, but it is still closely associated with the current frame. When the buffer is shown the view title changes to show that the buffer is visible, the current frame number is still shown in brackets because the user still needs to be aware of which frame is current in order to use the buffer.

Clear buffer Ctrl+0

This command clears the data in all channels of the frame buffer to zero. This is the initial state of the buffer after a CFS data file has been loaded.

Copy to buffer Ins

This command copies the data in the current frame into the frame buffer.

Copy from buffer Ctrl+Ins

This command copies the data in the frame buffer, and any count of sweeps averaged, to the current data frame.

Exchange buffer Shift+Ins

This command exchanges the data in the frame buffer, and any count of sweeps averaged, with the data in the current frame.

Add to buffer +

This command adds the data in the current frame to the data in the frame buffer. There is an alternative form of this command, available only through the Ctrl + shortcut and the Multiple frames dialog, which adds the buffer data to the data in the current frame.

Subtract buffer -

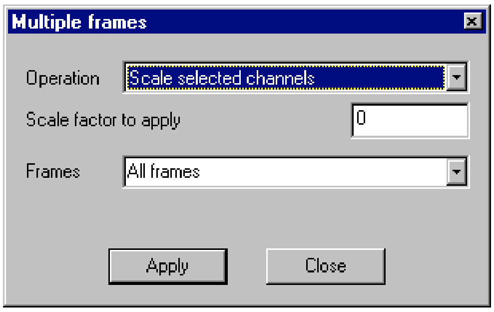
This command subtracts the data in the frame buffer from the current frame. There is an alternative form of this command, available only through the Ctrl - shortcut and the Multiple frames dialog, which subtracts the current frame data from the buffer.

Average into buffer
Enter

This command adds the data in the current frame into an average accumulating in the frame buffer. The addition is carried out in such a way that the buffer holds the average of frames accumulated. If the buffer contains data before averaging starts, this will be included as the first frame of the average. There is an alternative form of this command, available only through the Ctrl Enter shortcut and the Multiple frames dialog, which removes the current frame data from an average in the buffer - this will not work correctly if the frame was not accumulated into the buffer average in the first place.

If you mix the buffer averaging commands with the normal addition and subtraction operations, you will find that normal addition and subtraction on the buffer 'resets' the average by setting the count of sweeps to far to one. The actual addition and subtraction act as you would expect.

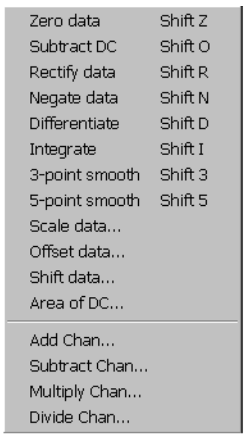
Multiple frames
Ctrl+M



This command provides a dialog that can be used to carry out numerous operations on multiple frames in the document. The dialog contains a selector for the operation to be carried out, a field to enter any operation data required (this is hidden if the operation does not need it) plus a standard set of controls to specify frames in the data document to be used. The dialog can be used repeatedly by pressing the Apply button, it doesn't disappear until Close is clicked.

The operations available from the dialog include all of the frame buffer operations, all of the channel data modification options, plus tag and untag frames. For operations that modify the frame buffer, such as accumulating an average or summing frames, the effect is straightforward. For operations that modify file view data, such as rectifying channels or subtracting buffer data from frames, the changed data must be saved to disk if the action is to have an effect (otherwise the changed file data will be discarded). This is because Signal only holds one frame from a data file in memory at a time; frames are loaded from disk as required and discarded when another frame is wanted. Use the File menu Data update mode option to ensure that changed frame data is saved, either unconditionally or by querying the user. For memory views, all of the document data is held in memory and changes to frame data are always saved.

Modify channels



This command provides a pop-up menu specifying the data modifications that are available. All the modifications operate on the selected waveform channels or on all visible waveform channels if none are selected. If the frame buffer is being shown then they operate on frame buffer data. In either case, all data points in the channels are modified. Most of the modifications are also available via the keyboard shortcuts shown in the menu. As for the frame buffer operations, changes to the frame data will be saved, or not, according to the file data update mode.

The behaviour of the modifications themselves are mostly straightforward. Subtract DC measures the mean value of the channel data, then subtracts this DC offset value from all data points. Normally, the DC level is measured over the visible frame area, the X range for the DC measurement can be set using the Area of DC item. Differentiation replaces each data point with the difference between that point and the previous point and divides the result by the sample interval; the first data point is set to zero. Integration replaces each data point with the sum of all data points up to and including that point multiplied by the sample interval. 3-point and 5-point smoothing replace each point with the average of the 3 or 5 points centred on that point. The scale and offset data options provide a dialog in which a numeric value can be entered. Scaling the data multiplies each data point by the number entered, offsetting adds the number entered to each data point. The shift data option rotates data points to the left or right by a specified number of points using a similar dialog; enter a negative number of points to shift left and a positive number of points to shift right. Note that these operations are rotates; data points that fall off one side of the frame are shifted back in on the other side, so the operation can be reversed without loss of data. There are special shortcuts Shift < and Shift > to shift left and right by one point.

The last four options are for inter-channel arithmetic. A channel will be prompted for and this channel will be applied with the appropriate operand to the other channels on a point by point basis.

Tag frame Ctrl+T

This command is used to tag or untag the current frame in the current view. When the current frame is tagged, this menu item is shown checked. All data frames in files handled by Signal can be tagged or untagged, the tagged status of a frame is displayed as part of the application status bar and can be interrogated by scripts. Frame tagging can be used for any purpose you require; all commands requiring a frame selection are able to operate on all tagged frames or all untagged frames.

The command toggles the tag state of the current frame, changing tagged frames to untagged and vice-versa.

Digital filters

This opens the Digital filtering dialog, which can create FIR filters and apply them to waveform channels (see the *Digital filtering* chapter for details).

Keyboard analysis control

Windows software is usually orientated towards control by means of the mouse and menus, but it is often convenient to use the keyboard instead. For interactive analysis of the data, using the keyboard can often be much faster. With this in mind, Signal includes keyboard shortcuts designed to handle most common data manipulation requirements:

Channel arithmetic

Zero channels	Shift Z
Negate data	Shift N
Rectify data	Shift R
Subtract DC level	Shift O
Differentiate data	Shift D
Integrate data	Shift I
3-point smooth	Shift 3
5-point smooth	Shift 5
Shift 1 point left	Shift <
Shift 1 point right	Shift >

Frame buffer operations

Toggle display of frame buffer	Ctrl B
Clear buffer data	Ctrl 0 (zero)
Add frame to buffer (average)	Enter
Add frame to buffer	+
Add buffer data to frame	Ctrl +
Subtract frame from buffer (average)	Ctrl Enter
Subtract buffer data from frame	-
Subtract frame from buffer	Ctrl -
Copy frame data to buffer	Insert
Copy buffer data to frame	Ctrl Insert
Exchange buffer and frame data	Shift Insert
Multiple frames dialog	Ctrl M

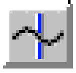
All of these shortcuts are documented with the appropriate menu commands. All analysis shortcuts are listed here for convenience, there are more shortcuts provided for display manipulation; see the View menu chapter.

Cursor menu

A cursor is a vertical or horizontal dashed line drawn in a Signal data view, to mark or obtain a position. The Cursor menu creates and destroys cursors, defines the possible active behaviour of vertical cursors, changes the display to make them visible, changes their labelling mode and obtains the values of channels where they cross the cursors and between the cursors.

The Cursor menu handles vertical and horizontal cursors. Up to 10 cursors of each type, numbered 1 to 10, can be active in each data view. Cursors can be dragged over and past each other and horizontal cursors can be dragged from channel to channel. Cursors in separate windows are independent of each other. When a window is duplicated, the cursors are also duplicated.

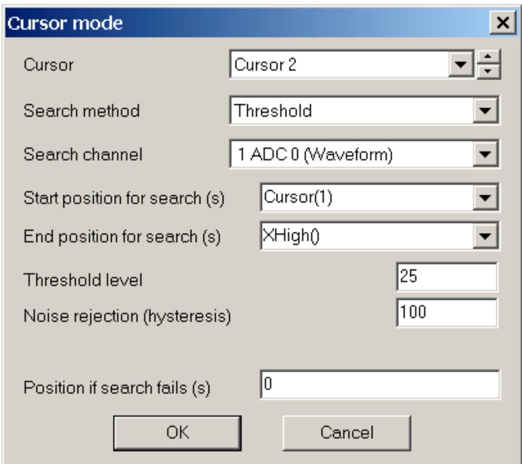
New Cursor Ctrl+|



This menu command duplicates the action of the new cursor button at the bottom left of data views. The command is available when a data view is the current window and there are less than four cursors already active in the view. A new vertical cursor is added at the centre of the window. The cursor is given the lowest available cursor number and is labelled with the cursor label style for the window.

Cursor mode

This menu command opens a dialog from which you can select a cursor and its search mode. The command is available when a data view is the current window and there are cursors present in the view.



Normally, Signal cursors are static; they stay where they are put. Using the cursor mode dialog, a cursor can be made active; it will move to the position of a feature in the data, if it can be found. This search-and-move occurs whenever the view changes to a different frame or when the current frame data is changed.

The cursor mode dialog allows you to define the feature searched for, the channel to be searched, the limits to the search and to set parameters defining the feature being searched for. The Cursor mode item at the top selects the feature to be searched for or defines the cursor as static. The Active channel item sets the (single) channel to be searched, while the start and end time items set the time range to be searched. Note that the start or end time for a search can be a cursor position, which itself could be active and changing. If the start time is later than the end time then Signal will search backwards for the feature. At the bottom of the dialog are up to three parameters (Noise rejection (hysteresis), Threshold level and Width for slope) used to define the feature more precisely, one or more of these may be present depending upon the cursor mode selected. The possible cursor modes are:

Static

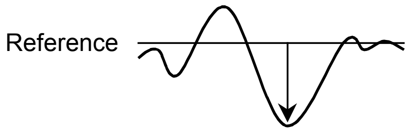
When you add a new cursor, it starts out with a mode of Static. In this state, the cursor stays where you put it; it is not changed by a change in the position of a lower numbered cursor.

Maximum value, Minimum value

The result is the position of the maximum or minimum value.

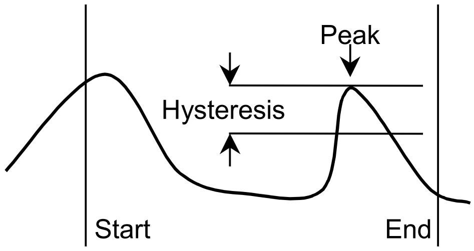
Maximum excursion

There is an extra field in this mode for the Reference level. The cursor is positioned at the point that is the maximum distance in the y direction away from the reference level.



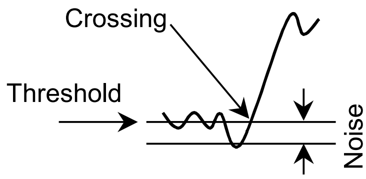
Peak, Trough

For these mode there is an extra field, Noise rejection (hysteresis), which defines how much the data must rise before a peak and fall after it (or fall before a trough and rise after it), for it to be accepted as a peak. In the diagram, which shows a peak search, the first peak is not detected because the data did not rise by Hysteresis within the time range.



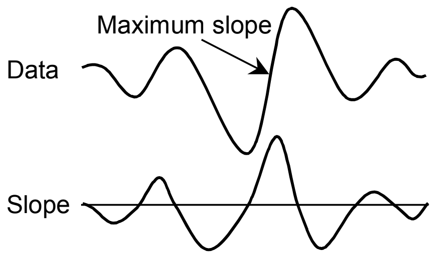
Rising threshold, Falling threshold, Threshold

In these modes there are two new fields, Threshold and Noise rejection (hysteresis). The data must cross Threshold from a level that is more than Noise rejection away from it. For the Rising threshold mode, the data must increase through the threshold, for Falling threshold mode it must fall through the threshold. In Threshold mode the crossing can be in either direction. The picture shows a rising threshold.



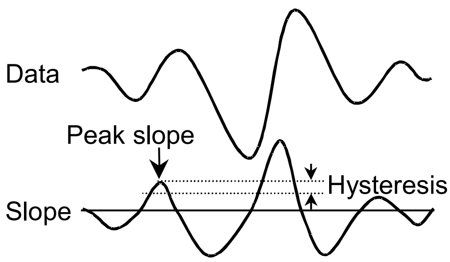
Steepest rising, Steepest falling, Steepest slope (+/-)

These modes have the extra field Width for slope measurement that sets length of data used to evaluate the slope at each data point. The result is the position of the maximum, minimum or maximum absolute value of the slope.



Slope peak, Slope trough

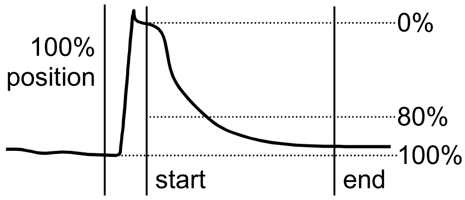
These two analysis modes calculate the slope of the data in the search range, and then search for the first peak or trough in the result that meets the Hysteresis specification. The Width for slope measurement field sets the length of data used to evaluate the slope at each data point. The Hysteresis field sets how much the slope must rise before a peak and fall after it (or fall before a trough and rise after it), for it to be accepted as a peak. The Hysteresis units are y axis units per second.



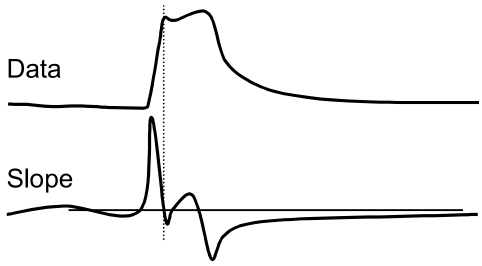
Slope threshold, +ve slope threshold, -ve slope threshold

The result is the position at which the slope crosses a particular threshold. The units for the threshold are y axis units per second

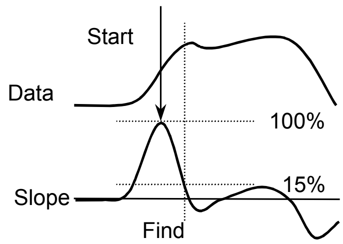
Repolarisation % This mode finds the point at which a waveform returns a given percentage of the distance to a baseline inside the search range. The start of the search range defines the position of 0% repolarisation. The additional fields, 100% position and Width, identify the 100% level (this can lie outside the search range). The Repolarise % field (drawn at 80% in the picture) sets a threshold level in percent relative to the 0% and 100% levels. The position is the first point in the search range that crosses the threshold.



Turning point This mode finds the first point in the search range where the slope of a waveform changes sign. Put another way, it finds a localised peak or trough in the waveform. The Width for slope measurement field sets the data range to calculate the slope. The picture shows this method used to find the top of a sharp rise where Maximum mode would get the wrong place. To use this you would probably set a cursor on the peak slope and start the search from that point



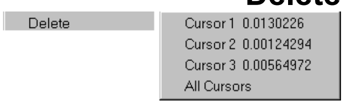
Slope% This method can be used to find the start and end of a fast up or down stroke in a waveform. The Width for slope measurement field sets the time width used to calculate the slope. The Slope% field sets the percentage of the slope at the start of the search area to search for. To use this mode you would set a cursor on the maximum or peak slope, then use that as the start point and search for the required percentage. A value of 15% usually works reasonably well.



Expression The cursor position is obtained by evaluating the Position field. This field will normally hold an expression based on cursor positions.

Active cursors can be used to increase the versatility of a number of Signal functions, for example the measurement possibilities from the cursor values and cursor regions window are considerably extended by the use of active cursors. Another noteworthy possibility is in the generation of trend plots, where the use of active cursors allows a wide variety of measurements.

Delete



The delete command activates a pop-up menu from which you can select a cursor to remove, or you can delete all the cursors. The cursors are listed with their number and position as an aid to identification. Deleting a cursor removes it from the window; other cursors are not affected.

Fetch

This activates a pop-up menu from which you can select a cursor to place in the centre of the x axis.

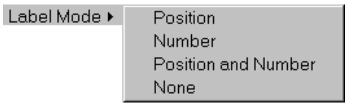
Move To

This command activates a pop-up menu from which you can select the cursor to move to. The cursors are listed with their number and position as an aid to identification. The window is scrolled to display the nominated cursor in the centre of the screen, or as close to the centre as possible. This command does not change the x axis scaling.

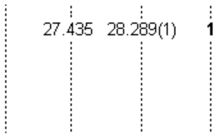
Display All

This command has no effect if there are no cursors. If there is a single cursor, the command behaves as though you had used the Move To command and selected it. When there are multiple cursors, the window is scrolled and scaled such that the earliest cursor is at the left-hand edge of the window and the latest is at the right-hand edge.

Label Mode



There are four cursor labelling styles: no label, position, position and cursor number, and number alone. Select the most appropriate mode for your purposes using the pop-up menu. To avoid confusion between the cursor number and the position, the number is displayed in **bold** type when it appears alone and bracketed with the position. The style applies to all the cursors in the window. You can drag the cursor labels up and down the cursor to suit the data (see page 4).



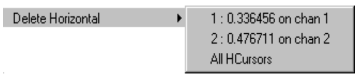
Renumber

When created, cursors take the lowest available cursor number rather than being ordered by position. You can also drag cursors over each other, confusing the ordering further. This command renumbers the cursors by position, with cursor 1 on the left.

New Horizontal Cursor

This menu command is available when a data view is the current window and there are less than four horizontal cursors already active in the view. A new horizontal cursor is added at the centre of the data for the lowest numbered visible channel. The cursor is given the lowest available number and is labelled using the horizontal cursor label style for the window.

Delete Horizontal



The delete command activates a pop-up menu from which you can select a horizontal cursor to remove, or you can delete all of them. The available cursors are listed with their number, position and channel number as an aid to identification. Deleting a cursor removes it from the window; other cursors are not affected.

Fetch Horizontal

This activates a pop-up menu from which you can select a horizontal cursor which will be placed in the centre of the visible y axis for the relevant channel.

Move To Level

This command activates a pop-up menu from which you can select the horizontal cursor to move to. The cursors are listed with their number, positions and channel as an aid to identification. The Y axis of the relevant channel will be scrolled to display the nominated cursor in the centre of the axis, or as close to the centre as possible. This command does not change the y axis scaling.

Display All Horizontal

This command is the equivalent of carrying out the Fetch Horizontal command for all cursors.

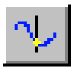
Horizontal Label Mode

There are four labelling styles for horizontal cursors, as for vertical cursors, use this command to select the most appropriate mode for your purposes. To avoid confusion between the cursor number and the position, the number is displayed in **bold** type when it appears alone and bracketed with the position. The style applies to all the cursors in the window. You can drag the cursor labels along the cursor to suit the data (see page 4).

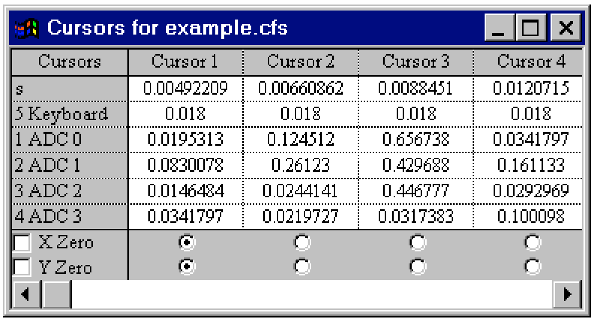
Renumber Horizontal

When created, cursors take the lowest available cursor number rather than being ordered by position. You can also drag cursors over each other, confusing the ordering further. This command renumbers the cursors by position, with cursor 1 at the top.

Display Y values



This command opens a new window containing the values at the position of any cursors in the current data view. Columns for cursors that are absent, or for which there is no data, are blank.



Cursors for example.cfs

Cursors	Cursor 1	Cursor 2	Cursor 3	Cursor 4
s	0.00492209	0.00660862	0.0088451	0.0120715
5 Keyboard	0.018	0.018	0.018	0.018
1 ADC 0	0.0195313	0.124512	0.656738	0.0341797
2 ADC 1	0.0830078	0.26123	0.429688	0.161133
3 ADC 2	0.0146484	0.0244141	0.446777	0.0292969
4 ADC 3	0.0341797	0.0219727	0.0317383	0.100098
X Zero				
Y Zero				

The values displayed depend upon the channel type and display mode. There is an entry in the table for each channel displayed. The values are as follows:

Waveform — The value displayed is the value of the nearest data point that is within one sample period of the cursor position, or nothing if there is no data point that is close enough. Waveform value measurements are not affected by the drawing mode.

Marker as Rate — The value displayed is the height of the rate bin that the cursor crosses. If the cursor lies precisely on the boundary between two rate bins, the cursor is considered to lie in the bin to the right.

Marker — The value shown is the time of the next marker at or to the right of the cursor.

The Time zero check box enables relative cursor time measurements. If checked, the cursor marked with the radio button is taken as the reference time, and the remaining cursor times are given relative to it. The reference cursor displays an absolute time, not 0.

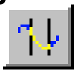
The Y zero check box enables relative cursor value measurements. The radio buttons to the right of the check box select the reference cursor. The remaining channels display the difference between the values at the cursor and the values at the reference. The values for the reference cursor are not changed.

Selecting and copying data

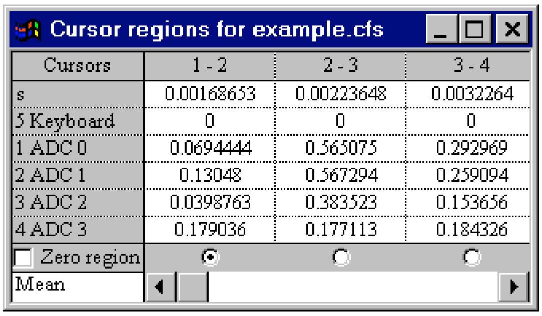
You can select areas of this window by clicking on them with the mouse. Hold down the Shift key for extended selections. You can select entire rows and columns by clicking in the cursor and channel title fields. Use the Ctrl key to select non-contiguous rows and columns.

Selected rows and columns can be copied to the clipboard by right-clicking on the values window and using the Copy command in the popup menu that appears. If you use the Log command the selected text is copied and pasted directly into the log window in one operation. You can also print the selected portions of the window by right-clicking and using the Print command in the popup menu, or use the Font command to change the window font.

Cursor Regions



This command opens a cursor region window for the current data view. This window calculates values for data regions between the cursors. One column can be designated the Zero region by checking the box and selecting the column with a radio button. The value in this column is then subtracted from the values in the other columns. The pop-up menu in the bottom-left corner indicates and controls how the values are calculated.



Cursor regions for example.cfs

Cursors	1 - 2	2 - 3	3 - 4
s	0.00168653	0.00223648	0.0032264
5 Keyboard	0	0	0
1 ADC 0	0.0694444	0.565075	0.292969
2 ADC 1	0.13048	0.567294	0.259094
3 ADC 2	0.0398763	0.383523	0.153656
4 ADC 3	0.179036	0.177113	0.184326
Zero region			
Mean			

Cursor region measurements

The region set by a pair of cursors is defined as the data starting at the first cursor up to, but not including, the data at the second cursor. For a waveform channel, the values calculated are:

Curve area — Each waveform point makes a contribution to the area of its amplitude above a line joining the endpoints multiplied by the time between samples on the channel.

Mean — The mean value of all the waveform points in the region. If there are no samples between the cursors the field is blank.

Slope — The slope of the least squares best fit line to waveform points in the region.

Area — Each waveform point makes a contribution to the area of its amplitude multiplied by the time between samples on the channel. If a zero region is specified, the amount subtracted from the other regions is scaled by the relative width of the regions.

Sum — The sum of all the waveform points in the region. If there are no samples between the cursors the field is blank.

Modulus — Each waveform point makes a contribution to the area of its absolute amplitude value multiplied by the time between samples on the channel. This is equivalent to rectifying the data, then measuring the area. If a zero region is specified, the amount subtracted from the other regions is scaled by the relative width of the regions.

Maximum — The value shown is the maximum value found between the cursors.

Minimum — The value shown is the minimum value found between the cursors.

Amplitude — The value shown is the difference between maximum and minimum values found between the cursors.

SD	The value shown is the standard deviation from the mean of the values found between the cursors. If there are no values between the cursors the field is blank.
RMS	The value shown is the RMS level of the values found between the cursors. If there are no values between the cursors the field is blank.
Extreme	The value shown is the maximum absolute value found between the cursors. Thus if the maximum value was +1, and the minimum value was -1.5, then this mode would display 1.5.
Peak	The value shown is the maximum value found between the cursors measured relative to the baseline formed by joining the two points where the cursors cross the data.
Trough	The value shown is the minimum value found between the cursors measured relative to the baseline formed by joining the two points where the cursors cross the data.

The measurements available for marker type channels are Mean, Sum, Maximum, Minimum, Amplitude and Extreme. If you select other measurements the result is a blank field. The values calculated for the measurements are:

Mean	The total number of markers between the cursors divided by the time difference between the cursors. This could be thought of as the mean marker rate.
Sum	The total number of markers between the cursors.
Maximum	The maximum inter-marker interval, or the maximum histogram value for Rate display mode.
Minimum	The minimum inter-marker interval, or the minimum histogram value for Rate display mode.
Amplitude	The difference between the Maximum and Minimum values.
Extreme	The largest absolute value of Maximum and Minimum, this will always be the same as Maximum for marker channels.

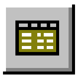
Selecting and copying data

You can select areas of this window by clicking on them with the mouse. Hold down the Shift key for extended selections. You can select entire rows and columns by clicking in the cursor and channel title fields. Use the Ctrl key to select non-contiguous rows and columns.

Selected rows and columns can be copied to the clipboard by right-clicking on the values window and using the Copy command in the popup menu that appears. If you use the Log command the selected text is copied and pasted directly into the log window in one operation. You can also print the selected portions of the window by right-clicking and using the Print command in the popup menu, or use the Font command to change the window font.

The sampling menu is divided into three regions. The first region is used before sampling to configure the channels required for data capture and provides support for users with a serial line controlled signal conditioner, for example the CED 1902. The signal conditioner option is only available with an online file. The second region of the menu is used during sampling to show or hide the sampling and output control panels. The third region matches the sampling control panel and can be used during sampling to start, continue and end the sampling process, enable and disable sweep triggers and to enable and disable data storage to disk.

Sampling configuration



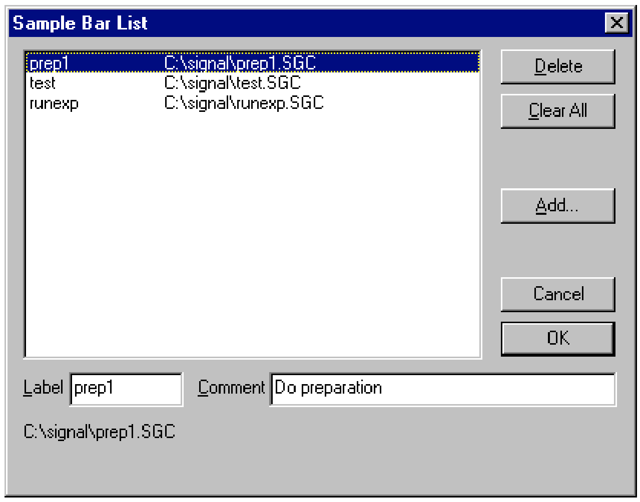
This command opens the Sampling Configuration dialog which defines the data capture parameters that are used when you select the File menu New command (see page 18 for details). The sampling configuration may be saved and loaded from the File menu Save Configuration and Load Configuration commands. You can also access this command from the Signal toolbar.

Sample Bar

You can show and hide the Sample Bar and manage the Sample Bar contents from the Sample menu. The Sample Bar is a dockable toolbar with up to 20 user-defined buttons. Each button is linked to a Signal configuration file. When you click a button, the associated configuration file is loaded and a new data file is opened, ready for sampling. You can also show and hide the Sample Bar by clicking the right mouse button on any Signal toolbar or on the Signal background.

Sample Bar List

The Sample menu Sample Bar List… command opens the Sample List dialog from where you can control the contents of the Sample Bar.



The Add button opens a file dialog in which you can choose one or more Signal configuration files (`*.SGC`) to add to the bar. If a file holds a label or comment, it is used, otherwise the first 8 characters of the file name form the label and the comment is blank.

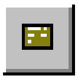
You can select an item in the list and edit the label and comment. This does not change the contents of the configuration file. You can re-order buttons in the bar by dragging items in the list. Delete removes the currently selected item. Clear All deletes all items.

The list of files in the Sample Bar is saved in the registry when Signal closes and is loaded when Signal opens. Each different logon to Windows has a different configuration in the registry, so if your system has three different users each has their own Sample Bar settings. Alternatively, you can have different experimental configurations by logging on with a different user name.

Signal conditioner

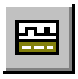
Signal two supports serial line controlled programmable signal conditioners. These devices amplify and filter waveform signals, and can provide other specialist functions. If a suitable conditioner is installed in your system, this menu command opens the conditioner dialog (see page **122**, the *Programmable signal conditioners* chapter for a full description).

Show Sampling controls

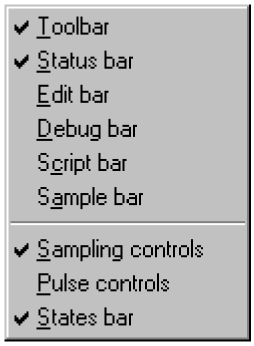


This command, or its toolbar equivalent, hides and shows the sampling control panel, the menu item is checked when the control panel is visible. The main controls within the control panel are duplicated in this menu as the Start sampling, Continue sampling, Triggered sweeps, Write to disk at sweep end, Pause at sweep end, Abort sampling and Restart sampling commands, (see page 29 in the *Sampling data* chapter for details of the control panel commands and what they do).

Show Pulse controls

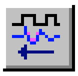


This command, and its toolbar equivalent, hides and shows the pulse definition dialog that is available during sampling if pulse output is in use. This dialog can be used to change the output pulses while data acquisition is in progress, changes made will be saved in the current sampling configuration. The menu item is checked when the control panel is visible.



The sampling control panel, pulse controls and states control bar can all be shown and hidden by using the popup menu generated by clicking the right mouse button on an unused part of the Signal window (the blank parts of the toolbar area are suitable and always visible) during sampling.

Sample now



This command is only available on the toolbar. It is equivalent to selecting New in the File menu then choosing Data Document. That is to say: it prepares Signal to start sampling with the current sampling configuration.

Show Sequencer controls

This command hides and shows the sequencer control panel that is available during sampling if the output sequencer is in use. Output sequencing during sampling is not supported in Signal version 2.00.

Start sampling

This command starts sampling off, it is the equivalent of the Start button in the sampling control panel.

Stop sampling

When sampling has been started off, the Start sampling command changes to Stop sampling. This is the equivalent of the Finish button in the sampling control panel. There is no warning before this command takes effect.

Continue sampling

This command is enabled when sampling is paused after collecting a sweep, it enables sampling of the next sweep. The command is the equivalent of the Continue button in the sampling control panel.

Triggered sweeps

This command toggles the state of the Sweep trigger checkbox in the sampling control panel. The menu item displays a checkbox when this option is selected.

Write to disk at sweep end

This command toggles the state of the Write to disk at sweep end checkbox in the sampling control panel. The menu item displays a checkbox when this option is selected.

Pause at sweep end

This command toggles the state of the Pause at sweep end checkbox in the sampling control panel. The menu item displays a checkbox when this option is selected.

Abort sampling

This command aborts sampling and discards any sampled data. It is the equivalent of the Abort button in the sampling control panel.

Restart sampling

This command restarts sampling by discarding all data and returning sampling to the state it was in before it was started. It is the equivalent of the Restart button in the sampling control panel.

Script menu

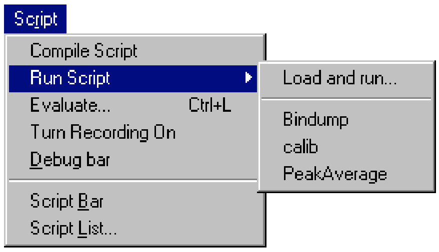
The Script menu gives you access to the scripting system. From it you can compile a script, run a loaded script, evaluate a script command for immediate execution and record your actions as a script. You can find details of the script language and a description of the script window in the separate manual *The Signal script language* and in the on-line help. The script menu commands are:

Compile Script

This command is enabled when the current view holds a script. It is equivalent to the Compile button in the script window. Signal checks the syntax of the script, and if it is correct, it generates a compiled version of the script, ready to run.

Run Script

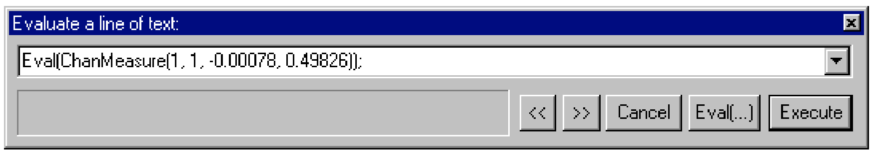
This command pops-up a list of all the scripts that have been loaded so that you can select a script to run. Signal compiles the selected script and if there are no errors, runs the script. If you run a script twice in succession, Signal only compiles it for the first run, saving the compilation time. If a script stops with a run time error, the script window is brought to the front and the offending line is highlighted.



You can also select the Load and run… option from which you can select a script to run. The script is hidden and run immediately (unless a syntax error is found in it).

Evaluate Ctrl+L

This command opens the evaluate window in which you can type a line of script commands for immediate compilation and execution. The window remembers the last ten lines of script enters, which are shown in the drop-down list, you can also cycle round the saved lines using the << and >> buttons. The Execute button directly executes the line entered, Eval(...) adjusts the line entered to include an Eval() on the script result (which displays the result), to save you the bother of adding it. You can execute any script that can be typed in one line, which can include variable declarations.



Turn Recording On/Off

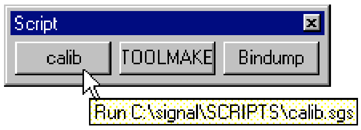
You can record your actions into a script that will produce equivalent actions. Use this command to turn recording on and off. When you turn recording on, Signal begins to save script commands corresponding to your actions. While script recording is in progress, the rightmost indicator in the Signal status bar will display the text REC as a reminder. When you turn recording off, a new script window opens that holds the saved script commands. If you then compile and run this script, the actions that you performed while recording was on will be repeated.

You can use this mechanism to record a sequence of actions that you wish to rerun at some later date, to find out what script commands correspond to a given menu command or user action or to record a sequence of actions that can be copied into another script or edited to produce a complete scripted 'application'.

Debug bar

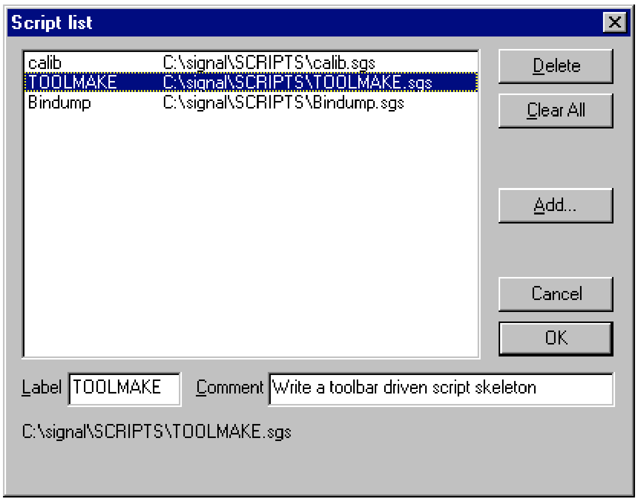
You can show and hide the debug bar from this menu when the current view is a script. You can also show and hide the debug bar by clicking the right mouse button on an unused area of the Signal application (unused parts of the toolbar area are suitable and always visible).

Script Bar



You can show and hide the Script Bar and manage the Script Bar contents from the Script menu. You can also show and hide the Script Bar by clicking the right mouse button on any Signal toolbar or on the Signal background. The Script Bar is a dockable toolbar with up to 20 user-defined buttons. Each button is linked to a Signal script file. When you click a button, the associated script is loaded and run. There is also a user-defined comment associated with each button which appears as a tool-tip when the mouse pointer lingers over a button.

Script List



This command opens the Script List dialog from where you can control the contents of the Script Bar.

The Add buttons opens a file dialog in which you can choose one or more Signal script files (`*.SGS`) to add to the bar. If the first line of a script starts with a single quote followed by a dollar sign, the rest of the line is interpreted as a label and a comment, otherwise the first 8 characters of the file name form the label and the comment is blank. The label is separated from the comment by a vertical bar. The label can be up to 8 characters long and the comment up to 80 characters. A typical first line might be:

```
'$ToolMake|Write a toolbar driven script skeleton
```

You can select an item in the list and edit the label and comment. This does not change the contents of the script file. You can re-order buttons in the bar by dragging items in the list. The Delete button removes the selected item. Clear All removes all items from the list.

The list of files in the Script Bar is saved in the registry when Signal closes and is loaded when Signal opens. Each different logon to Windows has a different configuration in the registry, so if your system has three different users each has their own Script Bar settings. Alternatively, you can have different experimental configurations by logging on as a different user name.

Window menu

The Window menu has seven permanently available commands in two sections. The first section holds three commands, one to duplicate a data document window and two to hide and show windows.

The second section holds five commands, four to arrange windows and the final one to close all windows.

The remaining space in the menu holds a list of all the windows that belong to the Signal application. If you select one of the windows in the list, the window is brought to the front and made the current window. The list shows the current window checked.

Duplicate window

This command creates a duplicate window with all the attributes (list of displayed channels, display modes, colours, cursors and size) of the original window. Once you have created the new window, it is independent of the original. Duplicating a window allows you to have different views of the same data with different scales and different channels visible.

You can close all windows associated with a data document using the File menu Close All command (see page **51**). This will remember the position and state of all windows associated with the document.

Hide

This command makes a window invisible. This is often used with script windows and sometimes is used to hide data windows during sampling when only the memory views with analysis results are required.

Show

This command lists all hidden windows. Select a hidden window to make it visible.

Tile Horizontally

You can arrange all the visible Signal windows so that they are arranged in a horizontally tiled pattern by using this command. Horizontal tiling arranges the windows so that they tend to be short and wide, the exact arrangement depends upon the number of windows.

Tile Vertically

You can arrange all the visible Signal windows so that they are arranged in a vertically tiled pattern by using this command. Vertical tiling arranges the windows so that they tend to be tall and thin, again the exact arrangement depends upon the number of windows.

Cascade

All windows are set to a standard size and are overlaid with their title bars visible.

Arrange Icons

You can use this command to tidy up the windows that you have iconised in Signal.

Close All

This command closes all windows in the Signal application. You are asked if you want to save the contents of any text windows that have changed. The positions of data document windows are all saved.

Help menu

Using help

Signal supports context sensitive help and also duplicates the contents of this manual in the help file. You can activate context sensitive help with the F1 key, or by pressing the Help button, from most dialogs to get a description of the dialog and its fields. You can use the Help menu Index command to get a dialog holding the help contents, an index to help keywords and a word search system to find topics that are not covered by the contents and index.

From a script view or the script evaluate dialog you can obtain help by placing the cursor on any keyword in the script and pressing F1. To get help on a script function, type the function name followed by a left hand bracket, for example `MemChan(`, then make sure that the cursor lies to the left of the bracket and in the function name and press F1. Pressing the help button (the button with a question-mark) at the top right of the script window provides overall script language help.

The help is implemented using the standard Windows help system, with contents, indexes, hypertext links, keyword searches, help history, bookmarks and annotations. If you are unsure about using Windows help, use the Help menu Using help command to get detailed instructions.

About Signal

This entry in the Help menu opens an information dialog that contains the serial number of your licensed copy of Signal, plus your name and organisation as entered during installation. Please quote the serial number if you call us for software assistance.

Tip of the Day

This command provides a dialog with a small piece of information in a "Did you know?" form. Further details can then be requested. This dialog can also be set to appear when Signal is first run.

View Web site

If you have an Internet browser installed in your system, this command will launch it and attempt to connect to the CED web site (`www.ced.co.uk`). The site contains downloadable scripts, updates to Signal and information about CED products.

Other sources of help

If you are having trouble using Signal, please do the following before contacting the CED Software Help Desk:

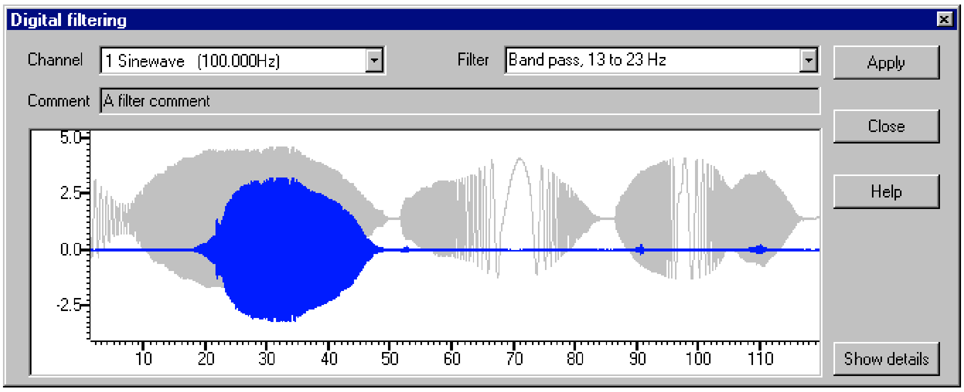
1. Read about the topic in the manual. Use the Index to search for keywords related to the topic.
2. Try the help system for more information. Use the Search facility to find related topics.
3. If none of the above helps, FAX, email or call the CED Software Help Desk (numbers and addresses are to be found at the front of this manual, and in the Contacting CED help page to be found near the start of the help contents). Please include a description of the problem, the Signal serial number and program version number and a description of the circumstances leading to the problem. It would also help us to know the type of computer you use, how much memory it has and which version of Windows you are running.

   If the problem involves a script, it would be useful to have a copy of the script, a data file suitable for use with the script and a description of how to provoke the problem.

Digital filtering

Introduction

The Analysis menu Digital filters… command is available when you have a data file open. You can apply one of twelve stored digital filters to a set of waveform channels over a selection of frames, though you can only preview one channel in the current frame at a time. You can also create your own digital filter. The program implements FIR filters (Finite Impulse Response) optimised to minimise the filter ripple in each filter band (see page 0-104 for more technical information on the filters).

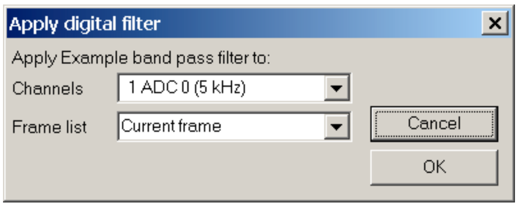


The Filter field of the dialog box selects the filter to apply and the Channel field sets the waveform channel to preview. The Comment field is for any purpose you wish; there is one comment per filter. The dialog shows the original waveform in grey, and a filtered version in the waveform colour.

The Close button shuts the dialog and will ask if you want to save any changed filter and the Help button opens the on-line Help at the digital filtering topic.

Apply

The Apply button opens a new dialog in which you set which channels and frames to filter. The channels should all have the same sampling rate as the one you were previewing in the main dialog. Channels of a differing sampling rate will be ignored

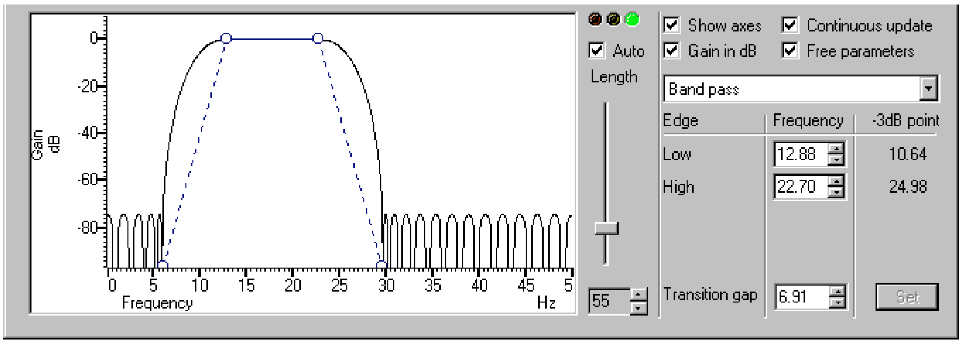


If the filter is of length n, then n/2 points around each input data point are used to produce each output point. When there is no input data available before or after a point, the filter uses a duplicate of the nearest input point as an estimate of the data value. This means that the n/2 output points next to either end of the input data should not be used for any critical purpose.

As applying a filter can be a lengthy process, a progress dialog appears with a Cancel button during the filtering operation.

Show details

The Show details button increases the dialog size to display a new area in which you can design and edit filters. Click this button again to hide the new dialog area.



The picture in the new area shows the frequency response of the filter. The Gain in dB check box sets the y axis scale to dB if checked, linear if not checked. The Show axes check box controls the axes of the raw and filtered data display. The frequency response display shows the ideal filter as solid lines for each defined band linked by dotted lines which mark each Transition gap between the bands. All transition gaps have the same width. The calculated frequency response is drawn as solid lines and is greyed when the filter specification has been changed and the response has not been calculated to match.

The circles can be dragged sideways to make the edges of the bands steeper or less steep or you can edit the band edges as numbers in the Frequency panel on the right. You can also drag the bands sideways and the band gaps. The mouse pointer changes to an appropriate symbol to indicate the feature you are dragging.

If you edit the numbers in the Frequency field, the Set button is enabled so you can force a recalculation of the filter.

The filters produced by the program are not defined in terms of -3dB corner frequencies and n dB per octave as is often the case for traditional analogue filters. The -3dB point column is present to help users who are more comfortable describing filter band edges in terms of the 3 dB point.

If you check the Continuous update box, the filter is updated while you drag the filter features around. If you have a slow computer and this feels ponderous you can clear the check box, in which case the filter is not recalculated until you stop changing features.

If you check the Free parameters box, dragged features are not limited by the next band and will push bands along horizontally. If you clear the box, the horizontal motion of a dragged feature is limited by the next moveable object.

To the right of the frequency response display is a slider that controls the number of filter coefficients. In general, the more coefficients, the better the filter. However, the more coefficients, the longer it takes to compute them and the longer to filter the data. If you check the Auto box, the program will adjust the number of coefficients for you to produce a useful filter. The "traffic light" display above the slider shows green if the filter is good, amber if the result is usable but not ideal, and red if the result is hopeless.

If you change a filter or create a new filter, you will be prompted to save the filter bank when you close the digital filter dialog.

Filter bank

A digital filter definition is complex and it would be tedious to specify all the properties of a filter each time you wanted to apply one to data. To avoid this, Signal contains a filter bank of 12 filter definitions. This filter bank is saved to the file `Filtbank.cfb` when you close Signal and reloaded when you open it. When you use the digital filter dialog, you specify which filter you want by the filter name. Script users identify the filter by an index number in the range 0 to 11.

Filter types

The type of the filter is set by the drop down list to the right of the display. If you need a filter that is not in this list you can generate it from the script language. Users of the `FIRMake()` script language command should note that the bands referred to here are pass bands. In the script language there are additional stop bands between the pass bands. There are currently 12 different filter types:

All pass

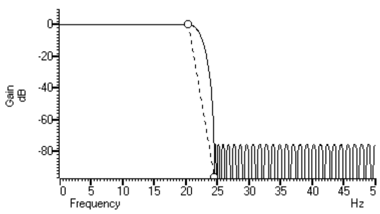
This has no effect on your signal. This filter type covers the case where you apply a low pass filter designed for a higher sampling rate to a waveform with a much lower sampling rate, so that the pass band extends beyond half the sampling frequency of the new file.

All stop

This removes any signal; the output is always zero. This filter type is provided to cover the case where you apply a high pass filter designed for a higher sampling rate to a waveform with a much lower sampling rate, so that the stop band extends beyond half the sampling frequency of the new file.

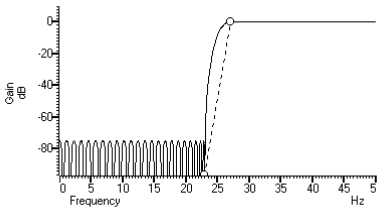
Low pass

This filter attempts to remove the high frequencies from the input signal. The Frequency field holds one editable number, Low pass, the frequency of the upper edge of the pass band. The stop band starts at this frequency plus the value set by the Transition gap field.



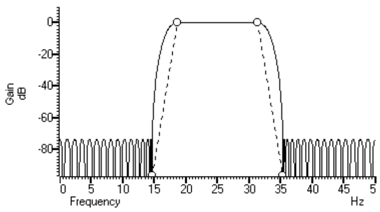
High pass

A high pass filter removes low frequencies from the input signal. The Frequency field holds one editable number, High pass, the frequency of the lower edge of the pass band. The stop band starts at this frequency less the value set by the Transition gap field.



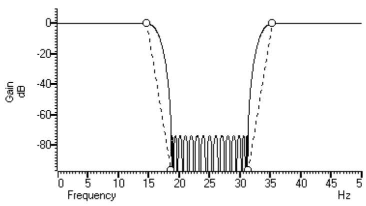
Band pass

A band pass filter passes a range of frequencies and removes frequencies above and below this range. The Frequency field has two editable numbers, Low and High, which correspond to the two edges of the pass band. The stop band below runs up to Low-Transition gap, and the stop band above from High+Transition gap to one half the sampling rate.



Band stop

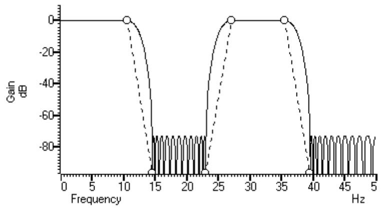
A band stop filter removes a range of frequencies. The Frequency field has two editable numbers, High (the upper edge of the first pass band) and Low (the lower edge of the upper pass band). The stop band below runs from High+Transition gap



up to Low-Transition gap.

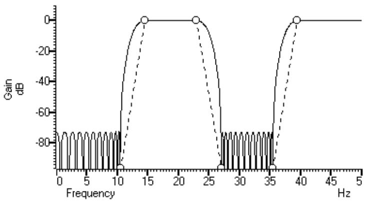
One and a half low pass

This filter has two pass bands, the first running from zero Hz and the second in the frequency space between the upper edge of the first pass band and one half the sampling rate. The Frequency field has three editable numbers: Band 1 high, Band 2 low and Band 2 high. These numbers correspond to the edges of the pass bands.



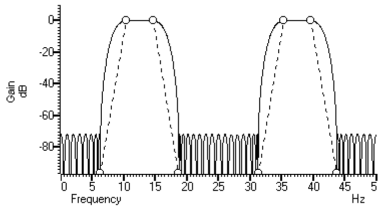
One and a half high pass

This filter has two pass bands. The second runs up to one half the sampling rate. The first band lies in the frequency space between 0 Hz and the lower edge of the second band. The Frequency field has three editable numbers: Band 1 low, Band 1 high and Band 2 low. These numbers correspond to the edges of the pass bands.



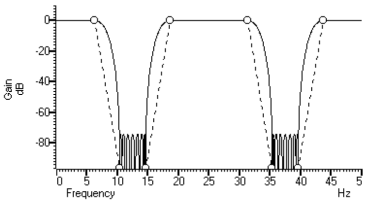
Two band pass

This filter passes two frequency ranges and rejects the remainder. Both 0 Hz and one half the sampling frequency are rejected. The Frequency field has 4 numeric fields: Band 1 low, Band 1 high, Band 2 low and Band 2 high. These fields correspond to the four edges of the two bands.



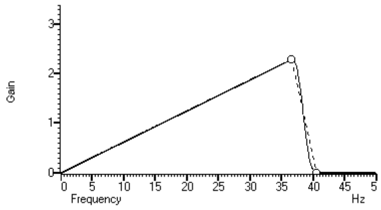
Two band stop

This filter passes three frequency ranges and rejects the remainder. Both 0 Hz and one half the sampling rate are passed. The Frequency field has 4 numeric fields: Band 1 high, Band 2 low, Band 2 high and Band 3 low. These fields correspond to the four edges of the three bands.



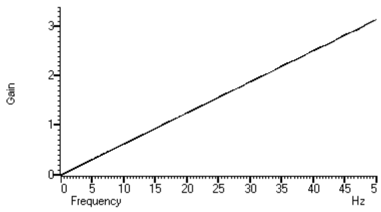
Low pass differentiator

This filter is a combination of a differentiator (that is the output is proportional to the rate of change of the input) and a low pass filter. The y axis scale is linear, rather than in dB (although you can display it in dB if you wish). There is one editable number in the Frequency field, Low pass, the end of the differential section of the filter.



Differentiator

The output of the filter is proportional to the rate of change of the input. The y axis scale is linear, rather than in dB (although you can display it in dB if you wish). The Frequency field is empty as there is only one band and it extends from 0 Hz to half the sampling rate.

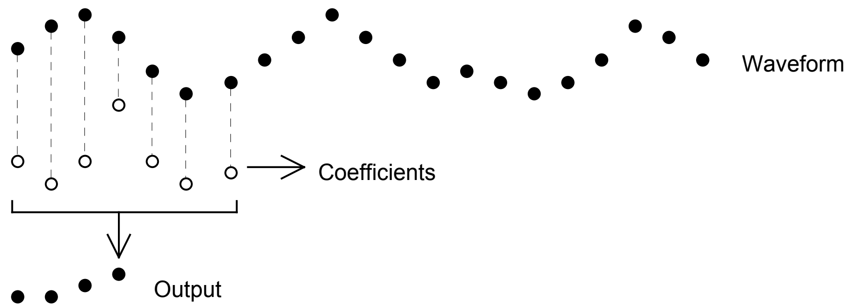


FIR filters

The `FIRMake()`, `FIRQuick()` and `FiltCalc()` script commands and the Analysis menu Digital filters… dialog generate FIR (Finite Impulse Response) filter coefficients suitable for a variety of filtering applications. The generated filters are optimal in the sense that they have the minimum ripple in each defined band. These filter coefficients are used to modify a sampled waveform, usually to remove unwanted frequency components. The algorithmic heart of the filter coefficient generation is based on the well-known FORTRAN program written by Jim McClellan of Rice University in 1973 that implements the *Remez exchange algorithm* to optimise the filter.

The theory of FIR filters is beyond the scope of this document. Readers who are interested in learning more about the subject should consult a suitable text book, for example *Theory and Application of Digital Signal Processing* by Rabiner and Gold published by Prentice-Hall, ISBN 0-13-914101.

FIR filtering



This diagram shows the general principle of the FIR filter. The hollow circles represent the filter coefficients, and the solid circles are the input and output waveforms. Each output point is generated by multiplying the waveform by the coefficients and summing the result. The coefficients are then moved one step to the right and the process repeats.

From this description, you can see that the filter coefficients (from right to left) are the *impulse response* of the filter. The impulse response is the output of a filter when the input signal is all zero except for one sample of unit amplitude. In the example above with 7 coefficients, there is no time shift caused by the filter. With an even number of coefficients, there is a time shift in the output of half a sample period.

Frequencies

The Analysis menu Digital filters… command deals with frequencies in Hz as this is comfortable for us to work with. However, if you calculate a FIR filter for one sampling rate, and apply the same coefficients to a waveform sampled at another rate, all the frequency properties of the filter are scaled by the relative sampling rates. That is, the frequency properties of an FIR filter are invariant when expressed as fractions of the sampling rate, not when expressed in Hz.

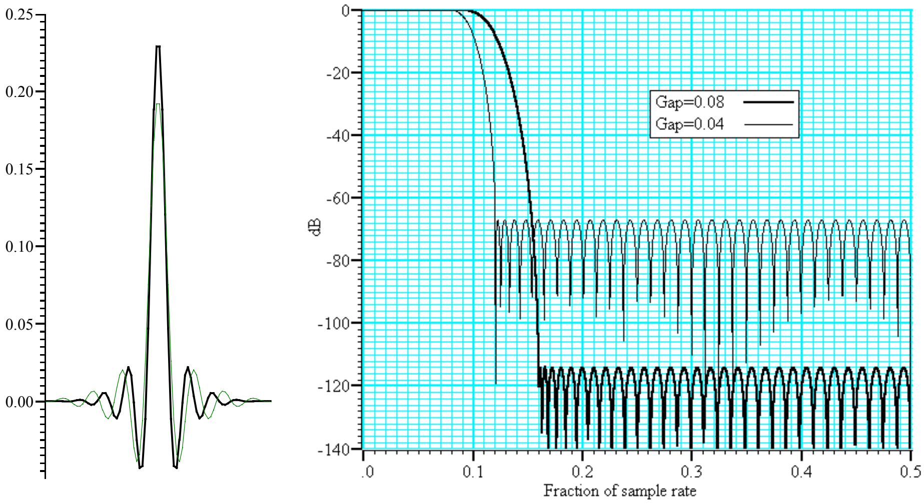
It is usually more convenient when dealing with real signals to describe filters in terms of Hz, but this means that each time a filter is applied to a waveform the sampling rate must be checked. If the rate is different from the rate for which the filter was last used, the coefficients must be recalculated. Unless you use the `FIRMake()` script command, Signal takes care of all the frequency scaling and recalculation for you. The remainder of this description is to help users of the `FIRMake()` script command, but the general principles apply to all the digital filtering commands in Signal.

Users of the `FIRMake()` script command must specify frequencies in terms of fractions of the sample rate from 0 to 0.5. For example, if you were sampling at 10 kHz and you wanted to refer to a frequency of 500 Hz, you would call this 500/10000 or 0.05.

Example filter The heavy lines in the next diagrams show the results obtained by `FIRMake()` when it designed a low pass filter with 80 coefficients with the specification that the frequency band from 0 to 0.08 should have no attenuation, and that the band from 0.16 to 0.5 should be removed. We can specify the relative weight to give to the ripple in each band. In this case, we said that it was 10 times more important that the *stop band* (0.16 to 0.5) should pass no signal than the *pass band* should be completely flat.

We have shown the coefficients as a waveform for interest as well as the frequency response of the filter. The shape shown below is typical for a band pass filter. One way of understanding the action of the FIR filter is to think of the output as the correlation of the waveform and the filter coefficients.

Coefficients and frequency response for low pass filters



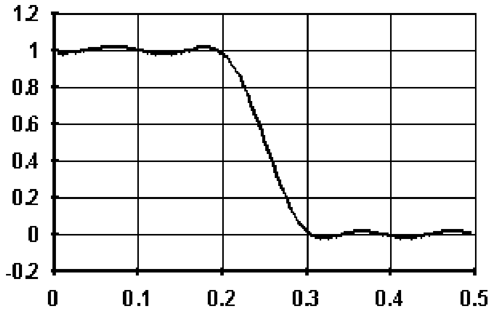
The frequency response is shown in dB which is a logarithmic scale. A ratio r is represented by 20 $\log_{10}(r)$ dB. A change of 20 dB is a factor of 10 in amplitude, 6 dB is approximately a factor of 2 in amplitude. The graph shows that a frequency in the stop band is attenuated by over 110 dB (a factor of 300,000 in amplitude with respect to the signal before it was filtered).

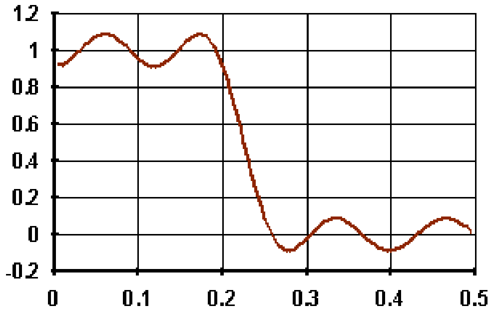
Because we didn't specify what happened between a frequency of 0.08 and 0.16 of the sampling rate, the optimisation pays no attention to this region. You might ask what happens if we make this transition gap smaller. The lighter line in the graph shows the result of halving the width of the gap by making the stop band run from 0.12 to 0.5. The filter is now much sharper. However, you don't get something for nothing. The attenuation in the stop band is reduced from 110 dB to around 70 dB. Although you cannot see it from the graph, the ripple in the pass band also increases by the same proportion (from 1 part in 30,000 to 1 part in 300).

We can restore the attenuation in the stop band by increasing the number of coefficients to around 120. However, there are limits to the number of coefficients it is worth having (apart from increasing the time it takes to calculate the filter and filter the data). Although the process used to calculate coefficients uses double precision floating point numbers, there are rounding errors and the larger the number of coefficients, the larger the numerical noise in the result.

Because the waveform channels are stored in 16-bit integers, there is no point designing filters that attenuate any more than 96 dB as this is a factor of 32768 (2^{15}). Attenuations greater than this would reduce any input to less than 1 bit. If you are targeting data stored in real numbers this restriction may not apply.

It is important that you leave gaps between your bands. The smaller the gap, the larger the ripple in the bands.





This is illustrated by these two graphs. They show the linear frequency response of two low pass filters, both designed with 18 coefficients (we have used so few coefficients so the ripple is obvious). Both have a pass band of 0 to 0.2, but the first has a gap between the pass band and the stop band of 0.1 and the second has a gap of 0.05. We have also given equal weighting to both the pass and the stop bands, so you can see that the ripple around the desired value is the same for each band.

As you can see, halving the gap has made a considerable increase in the ripple in both the pass band and the stop band. In the first case, the ripple is 1.76%, in the second it is 8.7%. Halving the transition region width increased the ripple by a factor of 5.

In case you were worrying about the negative amplitudes in the graphs, a negative amplitude means that a sine-wave input at that frequency would be inverted by the filter. The graphs with dB axes consider only the magnitude of the signals, not the sign.

FIRMake() filter types

`FIRMake()` can generate coefficients for four types of filter: Multiband, Differentiators, Hilbert transformer and a variation on multiband with 3 dB per octave frequency cut. The other routines can generate only Multiband filters and Differentiators.

Multiband filters

The filter required is defined in terms of frequency bands and the desired frequency response in each band (usually 1.0 or 0.0). Bands with a response of 1.0 are called *pass bands*, bands with a response of 0.0 are called *stop bands*. You can also set bands with intermediate responses, but this is unusual. The bands may not overlap, and there are gaps between the defined bands where the frequency response is undefined. You give a weighting to each band to specify how important it is that the band meets the specification. As a rule of thumb, you should make the weight in stop bands about ten times the weight in pass bands.

`FIRMake()` optimises the filter by making the ripple in each band times the weight for the band the same. The ripple is the maximum error between the desired and actual filter response in a band. The ripple is usually expressed in dB relative to the unfiltered signal. Thus the ripple in a stop band is the minimum attenuation found in that band. The ripple in a pass band is the variation of the frequency response from the desired response of unity. In some situations, for example audio filters, quite large ripples in the pass band are tolerable but the same ripple would be unacceptable in a stop band. For example, a ripple of -40 dB in a pass band (1%) is inaudible, but the same ripple in a stop band would allow easily audible signals to pass. By weighting bands you can increase the attenuation in one band at the expense of another to suit your application.

Differentiators

The output of a differentiator increases linearly with frequency and is zero at a frequency of 0. The differentiator is defined in terms of a frequency band and a slope. The frequency response at frequency f is f * *slope*. The slope is usually set so that the frequency response at the highest frequency is no more than 1.

The weight given to each frequency within a band is the weight for that band divided by the frequency. This gives a more accurate frequency response at low frequencies where the resultant amplitude will be the smallest.

Although you can define multiple bands for a differentiator, it is unusual to do so. Almost all differentiators define a single band that starts at 0. Occasionally a differentiator followed by a stop band is needed.

Hilbert transformers

A Hilbert transformer is a very specialised form of filter that causes a phase shift of $-\pi/2$ in a band, often used to separate a signal from a carrier. The theory and use of this form of filter is way beyond the scope of this document. Unless you know that you need this filter type you can ignore it.

Multiband with 3dB/octave cut

This is a variation on the multiband filter that can be used to filter white noise to produce band limited pink noise. The filter is identical to the band pass filter except that the attenuation increases by 3 dB per octave in the band (each doubling of frequency reduces the amplitude of the signal by a factor of the square root of 2). It is used in exactly the same way as the multiband filter.

Low pass filter example

A waveform is sampled at 1 kHz and we are interested only in frequencies below 100 Hz. We would like all frequencies above 150 Hz attenuated by at least 70 dB.

A low pass filter has two bands. The first band starts at 0 and ends at 100 Hz, the second band starts at 150 Hz and ends at half the sampling rate. Translated into fractions of the sampling rate, the two bands are 0-0.1 and 0.15 to 0.5. The first band has a gain of 1, the second band has a gain of 0. We will follow our own advice and give the stop band a weight of 10 and the pass band a weight of 1. We will try 40 coefficients to start with, so a possible script is:

```
var prm[5][2];       'Array for parameters
var coef[40];        'Array for the coefficients
'    band start        band end        function            weight
prm[0][0]:=0.00; prm[1][0]:=0.1; prm[2][0]:=1.0; prm[3][0]:= 1.0;
prm[0][1]:=0.15; prm[1][1]:=0.5; prm[2][1]:=0.0; prm[3][1]:=10.0;
FIRMake(1, prm[][], coef[]);
PrintLog("Pass Band ripple=%.1fdB Stop band attenuation=%.1f\n",
         prm[4][0], prm[4][1]);
```

If you run this, the log view output is:

```
Pass Band ripple=-28.8dB Stop band attenuation=-48.8
```

The attenuation in the stop band is only 48 dB, which is not enough. The ripple in the pass band is around 3% of the signal amplitude. We can increase the stop band attenuation in three ways: by increasing the number of coefficients, by giving the stop band more weight, or by making the gap larger between the bands.

We don't want to give the stop band more weight as this would increase the ripple in the pass band. We could probably reduce the width of the pass band a little as the attenuation of the signal tends to start slowly, but we will leave that adjustment to the end. The best way to improve the filter is to increase the number of coefficients. If we increase the size of `coef[]` to 80 coefficients and run again, the output now is:

```
Pass Band ripple=-58.7dB Stop band attenuation=-78.7
```

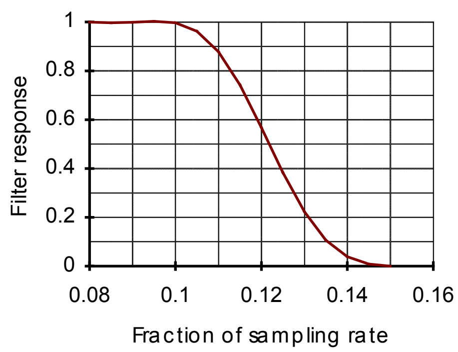
This is much closer to the filter we wanted. You might wonder if there is a formula that can predict the number of coefficients based on the filter specification. There is no exact relationship, but the following formula, worked out empirically by curve fitting, predicts the number of coefficients required to generate a filter with equal weighting in each of the bands and is usually accurate to within a couple of coefficients. The formula can be applied when there are more than two bands, but becomes less accurate as the number of bands increase.

```
' dB     is the mean ripple/attenuation in dB of the bands
' deltaF is the width of the transition region between the bands
' return An estimate of the number of coefficients
Func NCoefMultiBand(dB, deltaF)
return (dB-23.9*deltaF-5.585)/(14.41*deltaF+0.0723);
end;
```

In our example we wanted at least 70 dB attenuation, and we weighted the stop band by a factor of 10 (20 dB). This causes a 10 dB improvement in the stop band at the expense of a 10 dB degradation of the pass band. Thus to achieve 70 dB in the stop band with the weighting, we need 60 dB without it. If we set these values in the formula (`dB` = 60, `deltaF`=0.05) , it predicts that 67.13 coefficients are needed. If we run our script with 67 coefficients, we get 70.9 dB attenuation, which is close enough!

A final finesse

If we look at the frequency response of our filter in the area between the pass band and the stop band, we see that the curve is quite gentle to start with. If you are used to using analogue filters, you will recall that the corner frequency for a low pass analogue filter is usually stated to be the frequency at which the filter response fell by 3 dB which is a factor of $\sqrt{2}$ in amplitude (when the response falls to 0.707 of the unfiltered amplitude).



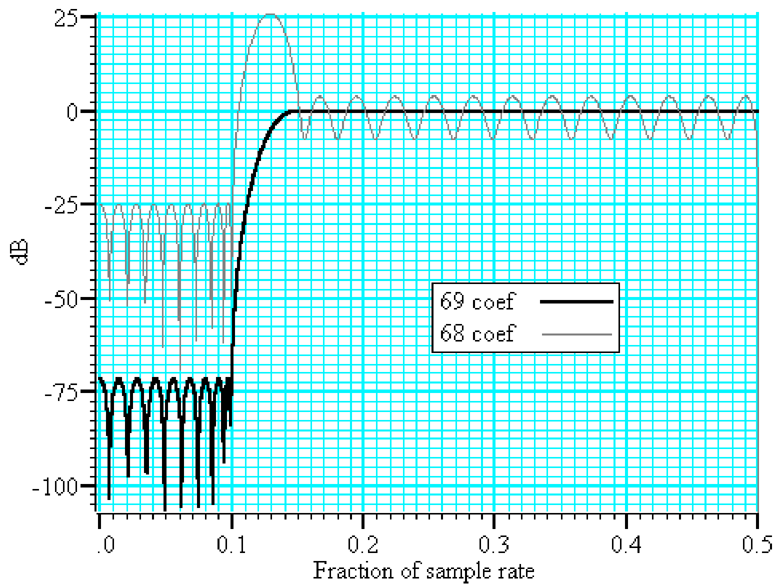
If we use the analogue filter definition of corner frequency, we see that we have produced a filter that passes from 0 to 0.115 of the sampling rate, and we wanted from 0 to 0.1, so we can move the corner frequency back. This will increase the attenuation in the stop band, and reduce the filter ripple, as it widens the gap between the pass band and the stop band. If we move it back to 0.085, the attenuation in the stop band increases to 84 dB. Alternatively, we could move both edges back, keeping the width of the gap constant. This leaves the stop band attenuation more or less unchanged, but means that the start of the stop band is moved lower in frequency.

High pass filter

A high pass filter is the same idea as a low pass except that the first frequency band is a stop band and the second band is a pass band. All the discussion for a low pass filter applies, with the addition that **there must be an odd number of coefficients**. If you try to use an even number your filter will be very poor indeed. The example below shows a script for a high pass filter with the same bands and tolerances as for the low pass filter. We have added a little more code to draw the frequency response in a result view.

```
var prm[5][2];
var coef[69];
'    band start        band end        function            weight
prm[0][0]:=0.00; prm[1][0]:=0.1; prm[2][0]:=0.0; prm[3][0]:=10.0;
prm[0][1]:=0.15; prm[1][1]:=0.5; prm[2][1]:=1.0; prm[3][1]:= 1.0;
FIRMake(1, prm[][], coef[]);
const bins% := 1000;
var fr[bins%];
FIRResponse(fr[], coef[], 0);
SetResult(bins%, 0.5/(bins%-1), 0, "Fr Resp", "Fr","dB");
ArrConst([], fr[]);
Optimise(0);
WindowVisible(1);
```

Effect of odd and even coefficients



The graph shows the results of this high pass filter design with 69 coefficients, which gives a good result, and with 68 coefficients, which does not. In fact, if we had not given a factor of 10 weight (20 dB) to the stop band, the filter with 68 coefficients would not have achieved any cut in the stop band at all!

The reason for this unexpected result is that we have specified a non-zero response at the Nyquist frequency (half the sampling rate). If you imagine a sine wave with a frequency of half the sample rate, each cycle will contribute two samples. The samples will be 180° out of phase, so if one sample has amplitude *a*, the next will have amplitude -*a*, the next *a* and so on. The filter coefficients are mirror symmetrical about the centre point for a band pass filter, so with an even number of coefficients, the result when the input waveform is *a*, -*a*, *a*, -*a*... is 0. Another way of looking at this is to consider that a filter with an even number of coefficients produces half a sample delay. The output halfway between points that are alternately +*a* and -*a* must be 0.

You can use the formula given for the low pass filter to estimate the number of coefficients required, but you must round the result up to the next odd number.

General multiband filter

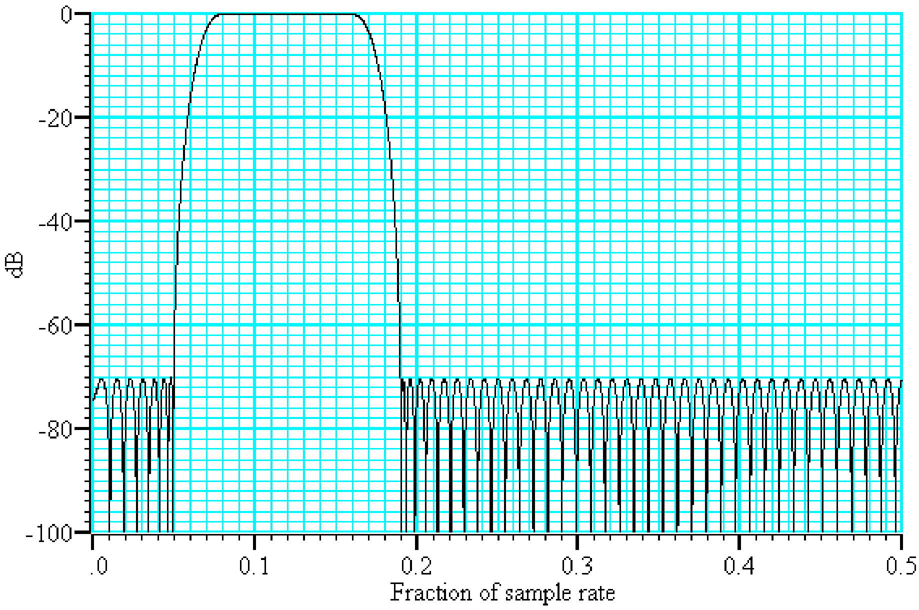
You can define up to 10 bands. However, it is unusual to need more than three. The most common cases with three bands are called band pass and band stop filters. In a band pass filter, you set a range of frequencies in which you want the signal passed unchanged, and set the frequency region below and above the band to pass zero. In a band stop filter you define a range to pass zero, and set the frequency ranges above and below to pass 1.

You must still allow transition bands between the defined bands, exactly as for the low and high pass filters, the only difference is that now you need two transition bands, not one. Also, if you want a non-zero response at the Nyquist frequency, you must have an odd number of coefficients.

For our example we will take the case of a signal sampled at 250 Hz. We want a filter that passes from 20 to 40 Hz (0.08 to 0.16) with transition regions of 7.5 Hz (0.03). If we say it is 10 times more important to have no signal in the stop band than ripple in the pass band, and we want 70 dB cut in the stop band we will get 50 dB ripple in the pass band (because a factor of 10 is 20 dB). To use the formula for the number of coefficients we need the mean attenuation/ripple in dB and the width of the transition region. The two stop bands have an attenuation of 70 dB and the pass band has a ripple of 50 dB, so the mean value is (70+50+70)/3 or 63.33 dB. We have two transition regions (both the same width). In the general case of transition regions of different sizes, use the smallest transition region in the formula. Plugging these values into the formula predicts 113 coefficients, however only 111 are needed to achieve 70 dB.

```
var prm[5][3];              ' 3 bands for band pass
var coef[111];              ' 111 coefficients needed
'    band start         band end        function          weight
prm[0][0]:=0.00; prm[1][0]:=0.05; prm[2][0]:=0.0; prm[3][0]:=10.0;
prm[0][1]:=0.08; prm[1][1]:=0.16; prm[2][1]:=1.0; prm[3][1]:= 1.0;
prm[0][2]:=0.19; prm[1][2]:=0.50; prm[2][2]:=0.0; prm[3][2]:=10.0;
FIRMake(1, prm[][], coef[]);
```

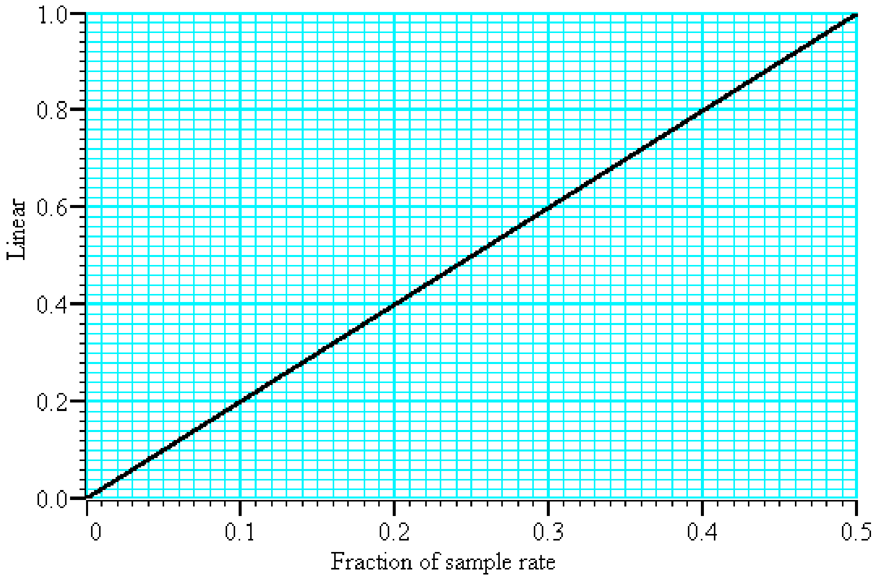
Band pass filter with 111 coefficients



Differentiators

A differentiator filter has a gain that increases linearly with frequency over the frequency band for which it is defined. There is also a phase change of 90° (π/2) between the input and the output.

Ideal differentiator with slope of 2.0



You define the differentiator by the number of coefficients, the frequency range of the band to differentiate and the slope. The example above has a slope of 2. Within each band (normally only 1 band is set) the program optimises the filter so that the amplitude of the ripple (error) is proportional to the response amplitude. A differentiator is normally defined to operate over a frequency band from zero up to some frequency `f`. If `f` is 0.5, or close to it, you must use an even number of coefficients, or the result is very poor. You can estimate the number of coefficients required with the following function:

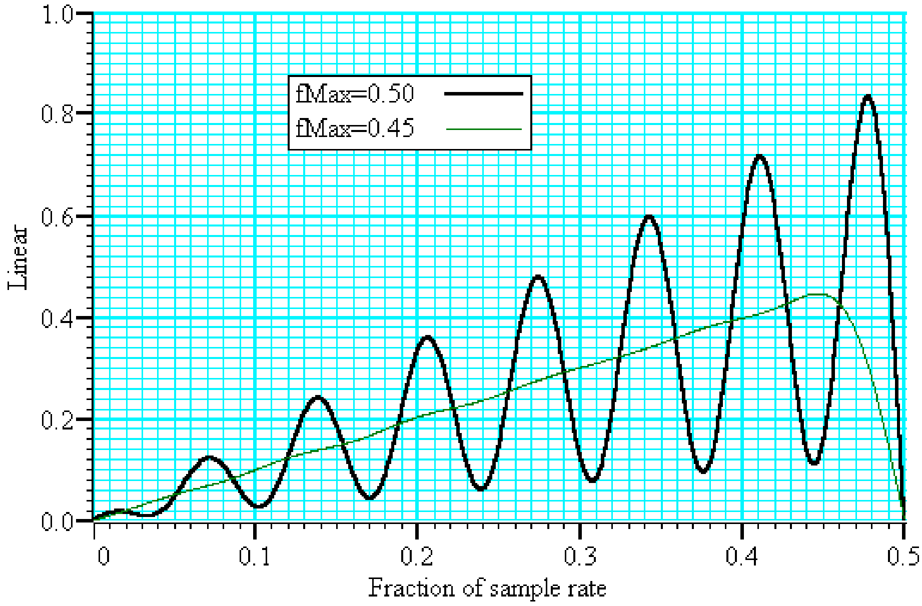
```
' dB    the proportional ripple expressed in dB
' f     the highest frequency set in the band
' even% Non-zero if you want an even number of coefficients
func NCoefDiff(dB, f, even%)
if (f<0) or (f>0.5) then return 0 endif;
f := 0.5-f;
var n%;
if (even%) then
   n% := (dB+43.837*f-35.547)/(0.22495+29.312*f);
   n% := (n%+1) band -2;   'next even number
else
   if f=0.0 then return 0 endif;
   n% := dB/(29.33*f);
   n% := n% bor 1;         'next odd number
endif;
return n%;
end
```

For an even number of coefficients this is unreliable when `f` is close to 0.5. For an odd number, no value of `n` works if `f` is close to 0.5.

These equations were obtained by curve fitting and should only be used as a guide. To make a differentiator that uses a small number of coefficients, use an even number of coefficients and don't try to span the entire frequency range. If you cannot tolerate the half point shift produced by using an even number of coefficients and must use an odd number, you must set a band that stops short of the 0.5 point. Remember, that by not specifying the remainder of the band you have no control over the effect of the filter in the unspecified region. However, for an odd number of points, the gain at the 0.5 point will be 0 whatever you specify for the frequency band.

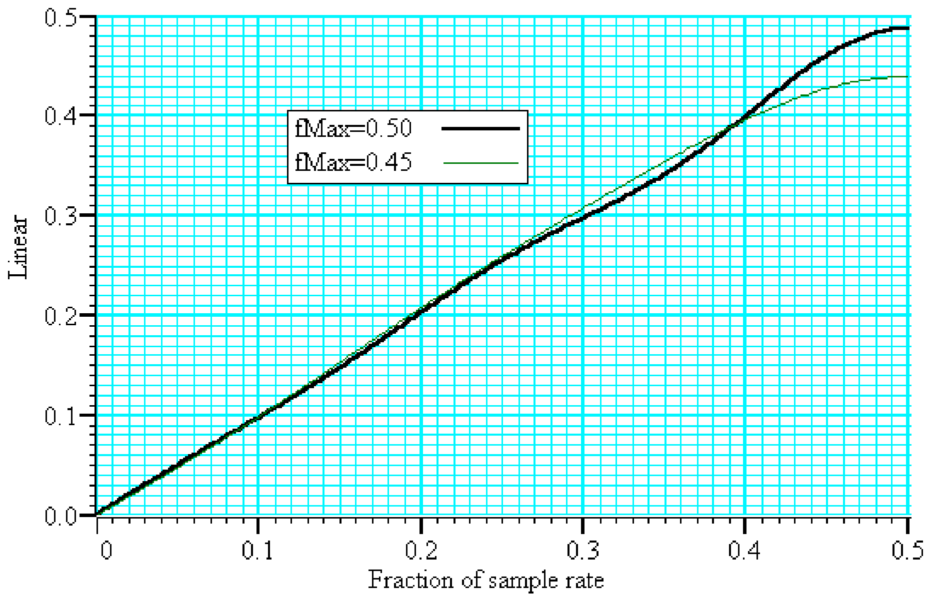
The graph below shows the effect of setting an odd number of coefficients when generating a differentiator that spans the full frequency range. The second curve shows the improvement when the maximum frequency is reduced to 0.45.

Differentiators with 31 coefficients



If you must span the full range, use an even number of coefficients. The graph below shows the improvement you get with an even number of coefficients. The ripple for the 0.45 case is about the same with 10 coefficients as for 31.

Differentiators with 10 coefficients




```
var prm[4][1];       ' 1 bands for differentiator
var coef[10];        ' 10 coefficients needed
'    band start          band end           slope           weight
prm[0][0]:=0.00; prm[1][0]:=0.45; prm[2][0]:=1.0; prm[3][0]:=1.0;
FIRMake(2, prm[][], coef[]);
```

Hilbert transformer

A Hilbert transformer phase shifts a band of frequencies from F_{low} to F_{high} by $-\pi/2$. The target magnitude response in the band is to leave the magnitude unchanged. F_{low} must be greater than 0 and for the minimum magnitude overshoot in the undefined regions, F_{high} should be $0.5-F_{low}$. The magnitude response at 0 is 0, and if an odd number of coefficients is set, then the response at 0.5 is also 0. This means that if you want F_{high} to be 0.5 (or near to it), you must use an even number of coefficients.

There is a special case of the transformer where there is an odd number of coefficients and $F_{high} = 0.5-F_{low}$. In this case, every other coefficient is 0. This is no help to the MSF and MSF programs, but users who write their own software can use this fact to minimise the number of operations required to make a filter.

It is extremely unlikely that a Hilbert transformer will be of any practical use in the context of Signal, so we do not consider them further. You can find more information about this type of filter in *Theory and Application of Digital Signal Processing* by Rabiner and Gold.

CFS file conversion

Filing systems used by CED

CED uses two data filing systems, CFS and SON. The CFS system, which is used by Signal, is also used by SIGAVG, Patch and VCLAMP, CHART, MassRAM, EEG and EEGER. CED have placed the CFS in the public domain to encourage its widespread use. Contact CED if you require more information about CFS.

The SON system is used by Spike2 (for DOS, Windows and the Macintosh), the VS software and CHART. SON is well suited to a multi-channel asynchronous data stream with a mixture of events, markers and waveform data with data access keyed by time.

Data stored in one file format is usually readily convertible into the other format. `C2S.EXE` and `S2C.EXE` are DOS programs that convert between the two formats so that Signal can exchange data with applications that use SON. The need for these programs have largely been replaced by the import facility now available in both Signal and Spike2 for Windows.

S2C - SON to CFS file conversion

S2C.EXE converts SON files to an equivalent CFS file. Each section of contiguous data is converted into a CFS data section, with the relative times of the various sections preserved. The data types accepted are waveform data and up to two marker channels. The first marker channel, if it appears to act as a section marker as used by CHART and Spike2, is used to indicate the start and end of the data sections. To use the program:

```
S2C source {dest} {-G} {-Tchan} {-Ltime} {-Z} {-Scode} {-Ecode}
```

`source` specifies the source SON file with a default extension of `.SMR`

`dest` specifies the destination CFS file, default extension `.CFS`. If omitted the source file name will be used with `.CFS` extension

`-G` Gaps in the ADC data define the CFS data sections (frames).

`-Tchan` Events on channel <chan> define the start of CFS data sections (frames).

`-Ltime` Length of a sweep is limited to <time> seconds.

`-Z` Force zero time at the start of each CFS data section (frame).

`-Scode` With `-T` option, markers with code <code> define the start of CFS data sections.

`-Ecode` With `-T` option, markers with code <code> define the end of CFS data sections.

S2C with neither the `-G` or `-T` uses the first marker channel found in the SON file to define the start and end of CFS data sections, code 01 ends the data section and code 00 starts another. This is similar to the effect of `S2C -T31 -S00 -E01`.

C2S - CFS to SON file conversion

`C2S.EXE` transfers data from CFS files to an equivalent SON file. The data sections in the source CFS file are converted into a single SON file with multiple data sections placed end-to-end, separated by a small time gap with the start and end of each section marked on the keyboard channel (channel 31) with marker codes 0 and 1. The data types accepted are equal-spaced channels containing 16 bit integer data and matrix data channels in a CFS marker format which are converted into SON waveform data and marker data respectively. To use the program:

```
C2S source {dest} {-txxx}
```

`source` Specifies the source CFS file with a default extension of `.CFS`

`dest` Specifies the destination SON file, default extension `.SMR`. If omitted the source file name is used with a `.SMR` extension.

`-t` The `-t` option specifies the fundamental SON timing interval in µs. If the option is omitted C2S calculates the interval. You would use this option if you wished to work at a higher time resolution in the file than that implied by the waveform sampling rates.

MCF Modify CFS File

Overview

`MCF.EXE` is a DOS utility program that reads one or more CFS data files and creates a new modified file. The modifications possible range from data extraction and channel deletion to waveform rectification, digital filtering and file amalgamation. It is likely that features of this program will be incorporated into Signal in the future; some, such as digital filtering, are already available via the script language.

The MCF program is command line driven. This means that all the information required to use it is supplied on the DOS command line. The options are:

```
MCF source [sourc2] dest -op [-ssects] [-ssects2] [-cchans]
```

`source` specifies the source CFS file for the data.

`sourc2` is an optional parameter, only used for the amalgamate operation, which specifies the second source file for the amalgamate operation.

`dest` specifies the name of the output CFS file that will be used to hold the result of the MCF operation.

`-op` specifies the operation to be carried out. `op` is a single character as follows:

- `x` Extract data from source (the default operation)
- `d` Delete channels or data sections
- `r` Rectify Int2 channels
- `a` Amalgamate channels from source and sourc2
- `h` High pass filter Int2 channels (time constant follows)
- `k` Calibrate Int2 channels (calibration file name follows)
- `f` Filter Int2 channels (coefficient file name and ratio follows)
- `n` No scaling down of data before filtering
- `q` Quiet mode - minimum screen output during operation
- `o` Ownership mode - don't alter file source information
- `p` Patch mode - allow Patch style file creation

The details of each of these operations are given below.

`-ssects` specifies a set of data sections from the source file. The data sections are specified as a list of items separated by commas, each item can be a single number or a range. For example, `-s1,2,5-8` specifies data sections 1, 2, and 5 to 8. Data section numbers from 1 to 16000 are allowed. The list specifies the data sections to be processed, the sections to be copied to the destination file for the extract option or the sections to be removed for the delete option. If no data section list is provided, all of the data sections will be processed or extracted.

`-ssects2` specifies a set of data sections for the second source file, if appropriate.

`-cchans` specifies a set of channels from the source file as a list separated by commas as for the data sections. For example `-c1-3,5,18` specifies channels 1, 2, 3, 5 and 18. This list specifies the channels to be processed, extracted or deleted according to the processing option selected. If no channel list is supplied, all channels are selected.

Extract data

The `-x` option is used to extract the specified channels and data sections from `source` and copy them, unchanged, into `dest`. For example:

```
mcf file1 dest -x -c1,2,3
```

copies channels 1, 2 and 3 only from `file1.cfs` to `dest.cfs`, using all data sections. If both channels and data sections are specified, only the specified channels in the specified data sections are extracted.

Delete data

The `-d` option is used to delete the specified channels and data sections from `source` and copy the rest, unchanged, into `dest`. For example:

```
mcf file1 dest -d -c1,2,3 -s9,10
```

copies all of the channels except channels 1, 2 and 3 from `file1.cfs` to `dest.cfs`, leaving out data sections 9 and 10. The delete option requires that either the channel list or the data section list be supplied.

Rectify Int2 channels

The `-r` option is used to copy all of the channels from `source` into `dest`, rectifying the specified Int2 equalspaced channels. If the `-s` specifier is used, only the data from the specified sections will be rectified. For example:

```
mcf file1 dest -r -c1,2,3
```

copies all of the data in `file1.cfs` to `dest.cfs`, rectifying Int2 data in channels 1, 2 and 3 in all data sections. If any of these channels do not hold Int2 equalspaced data then they are copied unchanged.

Amalgamate files

The `-a` option is used to amalgamate specified data sections from two source files into one destination file (see below for the compatibility conditions for these two source files). The `-c` option, if supplied, limits the range of data channels to be copied. For example:

```
mcf file1 file2 dest -a -s1,2,3 -s3,5 -c1,2,3
```

copies data sections 1, 2, and 3 from `file1.cfs`, plus data sections 3, and 5 from file `file2.cfs` to `dest.cfs`, using channels 1, 2 and 3 from both source files.

The two data section specifiers select data sections from the first and second source file. If neither of the data section specifiers is supplied, then all of the existing data sections from both source files will be copied into the destination file in the right order. If there is only one specifier it will apply to the first source file only and all of the data sections from the second source file will be copied. If the `-c` option is not supplied then all the existing channels will be copied.

Compatibility of source files

In order to be compatible, the two source files must fulfil the following conditions:

1. The files must have the same number of channels, file variables and data section variables.
2. The channels must be of the same kind and type. Matrix channel must be grouped in the same manner.
3. The file variables and data section variables must have the same type and size.

If any of these conditions is not satisfied the process will terminate and the relevant error message will be printed out on the screen.

High Pass Filter Int2 channels

The `-h` option is used to copy all of the channels from `source` into `dest` while filtering with the equivalent of an RC high pass filter. A time constant in seconds may be specified after the `-h` (if omitted 1 second is used). If the `-s` specifier is used, only the data from the specified sections will be filtered. For example:

```
mcf file1 dest -h0.1 -c1,2,3
```

copies all of the data in `file1.cfs` to `dest.cfs` filtering (with a time constant of 0.1 seconds) Int2 data in channels 1, 2 and 3 in all data sections. If any of these channels do not hold Int2 equalspaced data then they are copied unchanged.

Calibrate Int2 channels

The `-k` option is used to copy all of the channels from `source` into `dest` while calibrating the specified Int2 channels. The `-k` option is extended to include the name of a text file holding the calibration data, with the default file extension of .txt. The `-s` option, if supplied, selects the data sections to be recalibrated. For example:

```
mcf file1 dest -kmycal -c1,2,3 -s4,5,6
```

copies all of the data in `file1.cfs` to `dest.cfs`, recalibrating Int2 equalspaced data in channels 1, 2 and 3 from data sections 4, 5 and 6 using the data read from the file `mycal.txt`. If any of these channels do not hold Int2 equalspaced data then they are copied unchanged.

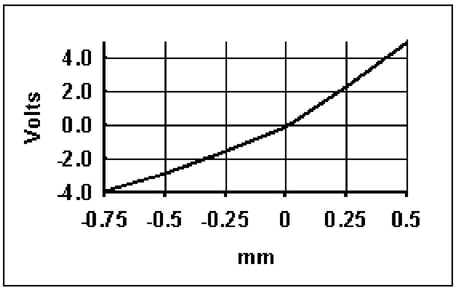
The calibration file specifies the units and range of values after calibration, plus a number of input values and their corresponding output values. These input and output values are used to construct a calibration 'curve' which defines the calibration process. Both values are real (floating point) numbers, the input values specify values as set by the scaling in the source channels, the output values are values in the scaling required for the output data. The calibration process converts each data point in the source file to an output value by interpolating between the calibration values. Input values that are outside the supplied range of values will be truncated. This option can be used to provide non-linear calibration of data, data compression or expansion, inversion and truncation in addition to simple calibration.

The lines in the calibration file holds the following information:

Line	Contents
1	The calibration file title. This is for information only and can be blank, but it must be present.
2	If blank, all channels in the -c list are calibrated. If non-blank, this holds the input units. Only channels with matching units (ignoring case) are calibrated.
3	The units for the calibrated data. If blank, the original units are retained.
4	Two numbers that set the maximum range of the output. The first number is the minimum value, the second is the maximum. This determines the mapping between the integer numbers used in the file and the values displayed. Setting too wide a range can result in loss of accuracy.
5	Channel comment for the calibrated data. If blank, the original comment is used.
6	The number of calibration points that follow.
7	The first calibration point as two numbers separated by a space or spaces. The first number is the input value, the second is the output value.
8	The second calibration point.
6+n	The n^{th} calibration point.

Lines in the file that start with a semicolon are comments and are ignored. An example may make this clearer. Consider a rather non-linear strain gauge whose output we have sampled as Volts. We also have a table of Volts and a displacement in mm:

Volts	-3.88	-2.81	-1.52	-0.043	2.351	5.0
mm	-0.75	-0.50	-0.25	0.00	0.25	0.50



We want to convert the data from Volts, into millimetres. We know that the output range will only span the range of the calibration, so we can set this to -0.75 to 0.50. A suitable calibration file, assuming that the input units were set to Volt, would be as follows:

```
;Strain gauge calibration
gauge 11a23 calibration
Volt
mm
-0.75 0.5
Voltage converted to mm
6
-3.88  -0.75
-2.81  -0.5
-1.52  -0.25
-0.043  0.0
2.351  0.25
5.00  0.50
```

If there are any input values below -3.88 Volts, they will be treated as though they were -0.75 mm. Similarly, inputs above 5.00 Volts will be limited to 0.50 mm. Both the source and destination units are specified, so only channels with units that match will be calibrated and have their units changed.

If there are problems in the format or contents of the file, MCF will refuse to use it. Common problems are caused by setting input values that exceed the possible range of the input data or calibration values that exceed the range of the output set in line 4.

This calibration file will convert from data directly logged as volts to mm using the calibration curve data. Input values below -3.88 volts cannot be converted and will be truncated. Both the source and destination units are specified so that only channels with units of 'Volt' will be calibrated. A new file comment is also provided. It is your responsibility to ensure that the source file is correctly scaled to provide readings in volts.

Digital filter Int2 channels

The `-f` option copies all channels from `source` to `dest` and filters specified waveform channels with a FIR (Finite Impulse Response) filter. The `-f` option is followed by the name of a text file holding filter coefficients. If no file extension is given, `.txt` is assumed. An optional down-sampling ratio can be specified after the file name, separated by a comma. For example:

```
mcf file1 dest -fmycoeff,2 -c1,2,3 -s4,5,6
```

copies all of the data in `file1.cfs` to `dest.cfs`, filtering Int2 equalspaced data in channels 1, 2 and 3 in sections 4, 5 and 6 using the coefficient data read from the file `mycoeff.txt`. All source data is down-sampled by a factor of 2 (only every second point kept) before filtering. If any of these channels do not contain Int2 equalspaced data they are copied unchanged.

Digital filtering is a topic beyond the scope of this manual. This option is provided for readers who are familiar with the technique and have access to software (such as the FIRMAKE program, available from CED) to calculate the coefficients to generate a particular filter.

Filter coefficient file

The filter coefficient file is a text file with the following format:

Line	Contents
1	The gain factor for the filter. This is a real number that is used to rescale the data after the filter operation, normally set to 1.0. The scale factor in the Signal file for the channel is divided by this number. It has no other effect.
2	The units for the filtered data. If non-blank, this will be used to set the units for all filtered channels.
3	The channel title for the filtered data. If non-blank, this will be used to set the channel title for all filtered channels.
4	The channel comment for the filtered data. If non-blank, this will be used to set the channel comment for all filtered channels.
5	The number of coefficients to follow.
6	The first coefficient, a real number in the range 1.0 to -1.0.
7	The second coefficient.
5+n	The n^{th} coefficient.

Lines in the filter coefficient file that start with a semicolon are comments and are ignored. A simple coefficient file that implements differentiation would look like:

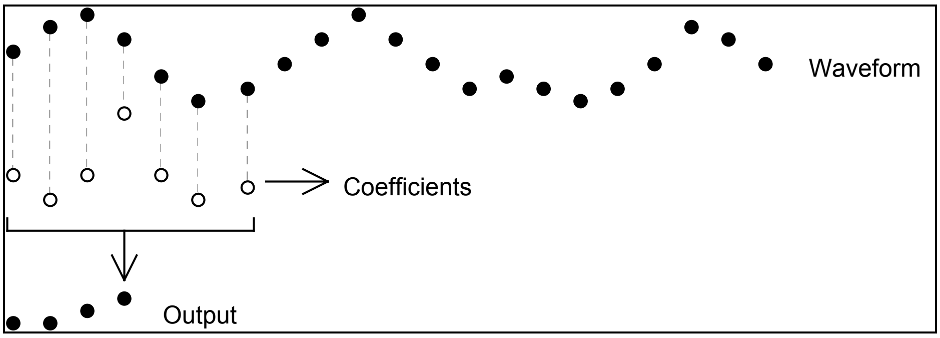
Example coefficient file

```
; A trivial coefficient file to differentiate
; Units, channel and title blank to leave channel unchanged
1.0



3
-0.5
0
0.5
```

Details of the filtering

The FIR filter is implemented by multiplying consecutive data points by a set of coefficients and summing the result to produce the output. The filtering operation can be thought of as a moving correlation of a short piece of test signal with the waveform.

FIR filtering



The coefficients are stepped along the waveform. At each step, each coefficient is multiplied by a corresponding data value and the sum of these multiplications is the output. The time of each output point is the mean of the times of all points used to generate the output. With an odd number of coefficients, the outputs occur at the same times as the inputs. With an even number, the outputs fall midway between the inputs.

We add extra data points at the start and the end of the input data to allow us to produce output to span the times of the input data. These extra points are given the value 0. If there are n coefficients, the first and last n/2 output points should be ignored.

Example filter files

We supply example filter coefficient files with the Signal system in the FILTERS directory. We have implemented low pass, high pass and notch filters and differentiators.

The low pass and high pass filters have two frequency bands, the pass band and the stop band. In the pass band (a band is a range of frequencies) the signal passes through the filter with little or no change in amplitude (but there can be changes in phase). In the stop band the signal is attenuated (the filters provided have a minimum attenuation of 40 dB in the stop band, which is a factor of 100). Between the pass band and the stop band is a transition region where the signal changes between the zero attenuation of the pass band and the 40 dB attenuation of the stop band.

The notch filters pass all frequencies except a narrow range around a centre frequency. This can be pictured as two pass bands separated by a narrow stop band. These filters are often used to remove mains interference.

We specify the frequency ranges to be used to define the bands as a fraction of the waveform channel sampling rate. You can find the sampling rate of a channel in a file by using the script BinSize() function. The file names used for the coefficient files have the form: `LPpppsss.TXT`, `HPsssppp.TXT` and `Nssswww.TXT`. The initial character indicates the type of the filter (LP = Low pass, HP = High pass, N = Notch). The remaining letters indicate the band edges of the filters in thousandths of the sampling rate. There cannot be frequencies above 0.5 of the sampling rate present in a data file. For example:

`LP150200.TXT` A low pass filter with a pass band from 0 to 0.150 of the sampling rate and a stop band from 0.200 to 0.5 of the sampling rate.

`HP050100.TXT` A high pass filter which rejects data from DC to 0.050 of the sampling rate and passes data from 0.100 of the sampling rate upwards.

`N200050.TXT` A notch filter centred at 0.200 of the sampling rate with a width of 0.050 of the sampling rate.

`D14.TXT` A differentiator with a width of 14 points - the slope is measured over an area around each point, the more points in the differentiator the greater the smoothing.

The choice of 40 dB as the attenuation in the stop band is entirely arbitrary, but should be useful for general purpose applications. Much higher attenuations can be achieved at the cost of more coefficients and slower filtering performance.

Notch filter generation

The program `NOTCH.EXE` (in the `FILTERS` directory) can be used to calculate coefficient files for notch filters. A notch filter is normally used to remove single unwanted frequencies (these are usually due to mains interference). The input to the program is the notch frequency as a fraction of the sampling rate, and the name of the output file:

```
NOTCH freq fname
```

`freq` The fraction of the sampling frequency in the range 0.001 to 0.5. Values outside this range cause the program to stop.

`fname` The name of the output file to hold the coefficients.

As an example, assume that we have sampled some data at 1000 Hz on channel 1 of a file and that this data is polluted by 50 Hz mains interference (or 60 Hz if you are in America or parts of Japan). The fraction of the sampling rate required is 50/1000 (60/1000) which is 0.050 (0.060). To attenuate this frequency:

```
NOTCH 0.05 n050.txt
MCF datafile newfile -fn050 -c1
```

Assuming that the old data was in DATAFILE.CFS, the result is now in the new file NEWFILE.CFS.

The program produces between 101 and 159 coefficients and the width of the notch is of order 1% of the sampling rate. Thus in our example above, the notch will affect frequencies in the range 40-60 Hz (50-70 Hz).

If mains interference is your problem, use the power spectrum analysis to discover if you are also suffering from 3rd and 5th harmonics of the mains frequency. If you are you may find that creating additional notch filters at these frequencies helps.

No scaling down before filtering

The -n option is used to prevent MCF automatically down-scaling the data before filtering. This down-scaling will prevent the filtering operation from overflowing, but it can make the resulting filtered data rather small. You can prevent this by using the -n option, but you will have to check that overflow has not occurred during filtering.

Quiet mode

The -q option is used to prevent the output of progress information to the screen during MCF operation. This option is primarily intended for use from batch files, it does not suppress the output of error messages.

Ownership mode

The -o option is used to prevent MCF altering the file source information in file variable zero. This holds information on the program that created the file, and is used by applications to recognise self-created files. MCF normally modifies this information to indicate that the file has been processed by MCF, this is required to prevent some programs from making incorrect assumptions about the file arrangement. Increasingly, it is more appropriate for the file ownership information to be left unchanged by use of the -o option, so that the file is seen to be acceptable.

Patch mode

The -p option allows use of MCF with files created by versions of the CED Patch Clamp suite prior to version 5.50 (released in July 91). The early Patch software used the CFS Y scaling factors in a way that is incompatible with modern CFS usage. Because of this, the MCF program detects these patch files and automatically corrects the scaling factors. The modified data file is usable with all CED software, except the old Patch software which created the original file. If you use the -p option, MCF will create CFS files compatible with old Patch, but only if the file was created by pre 5.50 Patch originally. The file created by MCF will not be recognised as Patch format a second time unless the -o option was also used.

Programmable Signal Conditioners

Signal can control programmable signal conditioners using the computer serial line ports and can use the signal conditioners to alter input signal gains, offsets or filtering before or during sampling. Three types of signal conditioner are supported: the CED 1902 Mk III, the Power1401 with programmable gains option and the Axon Instruments CyberAmp. You can open the conditioner control panel from either the sampling configuration dialog port setup page or from the Sample menu.

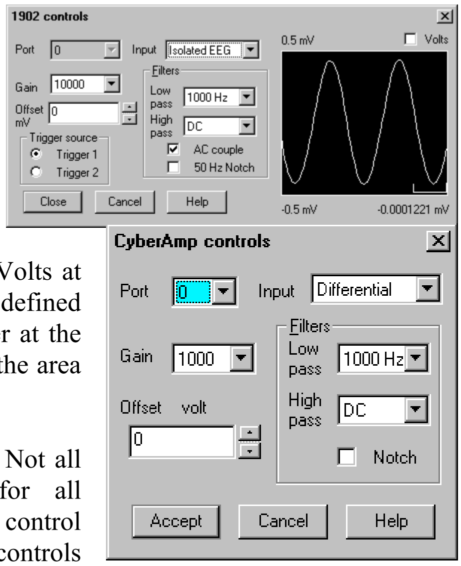
What a signal conditioner does

A signal conditioner takes an input signal and amplifies, shifts and filters it so that it can be sampled effectively by the data acquisition unit. Many input signals from experimental equipment are too small, or are masked by high and or low frequency noise, or are not voltages and cannot be connected directly to the 1401.

Signal conditioners may also have specialist functions, for example converting transducer inputs into a useful signal, or providing mains notch filters. The CED 1902 has options for isolated inputs and specialised front ends include ECG with lead selection, magnetic stimulation artefact clamps and EMG rectification filtering and filtering. With the Power1401 with programmable gains option only the gain may be set from within Signal.

You should consult the documentation supplied with your signal conditioner to determine the full range of capabilities.

Control panel



The control panel is in two halves. The left-hand half holds the controls that change the conditioner settings, the right-hand half displays data from the conditioner. The right half is omitted if Signal is sampling data or if the 1401 is not available for any other reason.

If the right-hand half is present, the Volts check box causes the data to be displayed in Volts at the conditioner input in place of user units as defined by the Channel parameters dialog. The number at the bottom right is the mean level of the signal in the area marked above the number.

Signal conditioners differ in their capabilities. Not all the controls listed below may appear for all conditioners. This example is the CyberAmp control panel (with the right-hand half omitted). The controls are:

Port This is the 1401 ADC port number whose conditioner settings are being adjusted. Only ports for which a conditioner was found are shown.

Input If your signal conditioner has a choice of input options, you can select the input to use with this field. The choice of input may also affect the ranges of the other options.

Gain This dialog field sets the gain to apply to the signal selected by the Input field. Signal tracks changes of gain (and offset) and will change the channel scaling factors in the ports configuration to preserve the y axis scale. To make the best use of the accuracy of the 1401 family, you should adjust the gain of your signal so that the maximum input signal does not exceed the limits of the data displayed on the right of the control panel.

Offset Some signals are biased away from zero and must be offset back to zero before they can be amplified. If you are not interested in the mean level of your signal, only in the fluctuations, you may find it much simpler to AC couple (1902) or high-pass filter (CyberAmp) the signal and leave the offset at zero.

Low-pass filter A low-pass filter reduces high-frequency content in your signal. Filters are usually specified in terms of a corner frequency, which is the frequency at which they attenuate the power in the signal by a factor of two and a slope, which is how much they increase the attenuation for each doubling of frequency. Sampling theory tells us that you must sample a signal at a rate that is as least twice the highest frequency component present in the data. If you do not, the result may appear to contain signals at unexpected frequencies due to an effect called aliasing. As the highest frequency present will be above the corner frequency you should sample a channel at several times the filter corner frequency (probably between 3 and 10 times depending on the signal and the application).

You can choose a range of filter corner frequencies, or you can choose to have the data unfiltered (for use when the signal is already filtered due to the source).

High-pass filter A High-pass filter reduces low-frequency components of the input signal. High-pass filters area specified in the same way as low-pass filters in terms of a corner frequency and a slope, except that the slope is the attenuation increase for each halving of frequency. If you set a high-pass filter, a change in the mean level of the signal will cause a temporary change in the output, but the output will return to zero again after a time which depends on the corner frequency of the filter. The lower the corner frequency, the longer it takes for mean level change to decay to zero.

Notch filter A notch filter is designed to remove a single frequency, usually set to the local mains power supply (50 Hz or 60 Hz, depending on country).

Reset calibration The Reset Calib. button resets the calibration information to show raw volts taking into account the current gain and offset. The units for the calibration will be set to V and the port Full and Zero values adjusted as appropriate. On the CyberAmp, this option will use the 'native' calibration information and units specified by a SmartProbe, if present.

The remaining options are for the 1902 only:

AC couple This is present for the 1902 only, and can be thought of as a high-pass filter with a corner frequency of 0.16 Hz. However, it differs from the high-pass filters as it is applied to the signal at the input; the high-pass filters in the 1902 are applied at the output.

Trigger 1 The 1902 provides two conditioned trigger inputs, and one output. This control selects which of the inputs is connected to the output.

Setting the channel gain and offset

If you change the gain or offset in the control panel, Signal will adjust the port Full and Zero values in the sampling configuration to compensate so as to keep the y axis showing the correct values. This means that if you change the gain, the signals will still be correctly calibrated in the file. However, the first time you calibrate the channel you must tell the system how to scale the signal into y axis units.

For example to set up the y axis scales in microvolts you do the following:

1. Open the Sampling Configuration dialog.
2. Select the ADC port in the Ports setup page.
3. Press the Conditioner button to open the conditioner control panel.
4. Adjust the gain to give a reasonable signal. Make a note of the gain *G* you have set.
5. Close the signal conditioner control panel.
6. Set the Units field of the Channel parameters to uV.
7. Set the Full field to 5000000/*G*.

You only need do steps 6 and 7 once. Any subsequent change to the conditioner gain will adjust the channel Full value to leave the units in microvolts.

For the more general case imagine you have a transducer that measures some physical quantity (Newtons, for example) and it has an output of 152.5 Newtons per V. If you wanted the y axis scaled in Newtons, you would replace steps 6 and 7 above with:

6. Set the Units field of the Channel parameters dialog to N.
7. Set the Full field to (5 * 152.5)/*G*.

To work this out you must express the transducer calibration in terms of Units per Volt (in this case Newtons per Volt), multiply this by 5 to get the Full value at five volts, then divide it by the gain of the conditioner.

If you have set an offset in the conditioner, and you want to preserve the mean signal level, you should null it out by changing the offset in the Channel parameters dialog.

Conditioner connections

Signal normally expects the first channel of signal conditioning to be connected to 1401 ADC port 0, the second to port 1 and so on. Connect the conditioner output BNC (labelled Amp Out on the 1902, and OUT on the CyberAmp) to the relevant 1401 input. This arrangement can be adjusted to start with another port, but in all cases the conditioner channel number must match the ADC port to which it is connected.

Signal conditioners connect to the computer with a serial line. You will have received a suitable cable with the unit. Basic communication and connection information is stored in the file `CEDCOND.INI` in the system directory of your computer. This file holds:

```
[General]
Port=COM1
```

The `Port` value sets the communications port to use. We would recommend you use the Edit menu Preferences… to change the port. If this file is missing, COM1 is used. Preferences… can also set a diagnostic option, enabled in the file by:

```
Dump=1
```

If this entry is included, Signal writes a log file called `CEDCOND.LOG` to the current directory when it initialises the conditioner.

As mentioned above, signal conditioners normally start at channel zero and go up to the last channel. The conditioners are searched for assuming this arrangement; the search starts at channel 0 and continues up until a missing channel is encountered. This search process can be adjusted by two more lines that can optionally be put into `CEDCOND.INI`:

```
First=4
Last=12
```

The search for conditioners will start with the channel specified as `First`, or zero if this isn't set and continue until the channel specified as `Last` has been checked and a missing channel has been encountered.

The arrangement is somewhat different for CyberAmps, as they have multiple channels. The channel numbers are specified by the rotary address switch at the back of the amplifier, note that the first amplifier must have the address set to zero, not 1 as normally shipped. The channels are numbered off in the order of the address switches (assuming 380s and 320s are not mixed), the `Last` value can be used to extend the search.

Index

—T—

—U—

—V—