

!SideShow

Release 2.000 - 26th Feb '93

About !SideShow

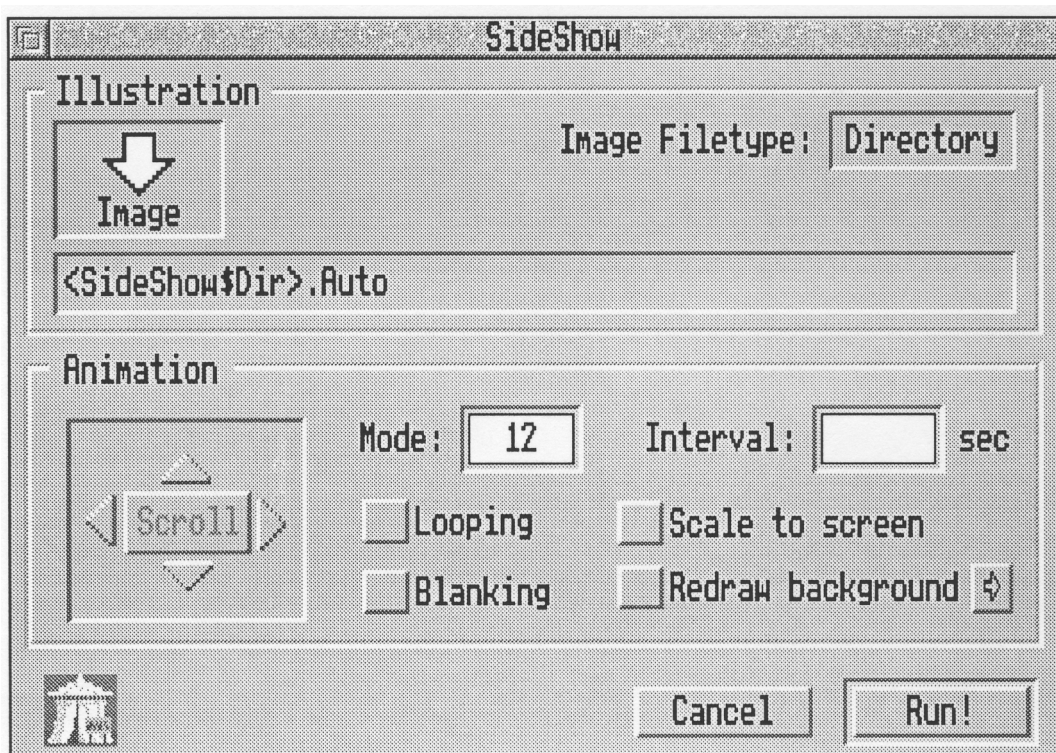
SideShow is a presentation program, equally suitable for video titling and "rolling demo" applications.

Rather than including complicated editing facilities, *!SideShow* uses *!Draw* files to allow easy combinations of text, line drawings and bitmaps (sprites).

If you are unfamiliar with *!Draw* files, refer to the section on *!Draw* in the application user guide supplied with your computer. For more detailed information, you may wish to consider purchasing a specialist book on using *!Draw*.

Running !SideShow.

In the usual RISCOS manner, double-clicking on *!SideShow* in a filer window will install the application on the icon bar. *!SideShow's* control panel will also appear in the centre of the screen:



Initially the *Mode* field will be set to your current desktop screen mode.

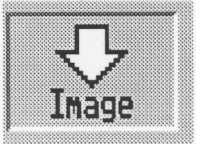
Viewing a demonstration sequence

As supplied, a demo sequence of draw frames is loaded into *!SideShow*. *!SideShow's* default sequence lives in the *Auto* directory inside the application. To find it, simply double click on the *!SideShow* icon in a filer window. You may change any of these files or add more of your own.

Clicking the *Run!* button now will run the example sequence ... press any key to move to the next frame in the sequence.

(Pressing the Escape key will quit the sequence at any point, and return to the desktop.)

Displaying images in !SideShow.

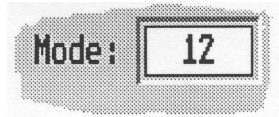


Dragging a drawfile onto to the drop zone will load it into *!SideShow*, and clicking the *Run!* icon will then display it. (You will be warned if there isn't enough memory to load the file).

A directory containing drawfiles may also be dropped onto *!SideShow*, and each drawfile in the directory will be displayed **in alphabetical order**. If there is insufficient memory to load any drawfile in the sequence, it is simply ignored.

!SideShow will complain if there are no drawfiles in the directory, but will ignore any non-drawfiles in a sequence.

Using !SideShow's controls

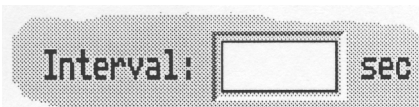


SideShow will run in any mode, though obviously for video titling purposes you are limited to broadcast-frequency (PAL) modes. These include modes 12, 15 and 16. Additional "overscan" modes such as mode 35 in RISCOS 3, and mode 50 provided by *Overlay* (supplied with Genlock cards) allow the complete screen area to be displayed.

SideShow uses the current desktop palette and supremacy settings. This means that you can use *Overlay* and the desktop's *Palette* utility to set up screen colours. (Colours may obviously differ if you use *!SideShow* in a mode with a different number of colours than the desktop).

The *Interval* field sets the time in seconds that *!SideShow* will wait before displaying the next frame.

Pressing a key during this period will interrupt the wait and move to next frame. Values may be entered in a number of ways:



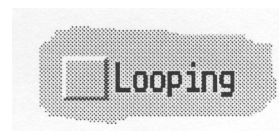
As a whole number, e.g. 3

As a fraction: 1/25

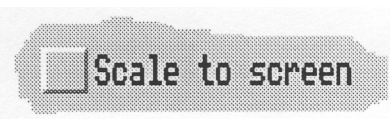
As a multiple: 5*60 (five minutes).

The shortest period is 1/99 of a second, the longest is 5 minutes (values larger than 300 seconds are rounded down to 300).

Entering zero (the default) will wait for indefinitely for a keypress.



By default, when *!SideShow* has displayed a sequence of frames it will return to the desktop. Setting the looping option will cause the sequence to repeat the sequence until the escape key is pressed. This option is useful for "rolling demos", allowing the machine to be left running unattended.



The *Scale To Screen* option will enlarge or reduce the size of the complete drawfile, and centres it on-screen, filling as much of the displayable area as possible. Without this option set images are displayed at 100% at the bottom-left corner of the screen.



For video titling purposes, it is useful for a caption to appear and then disappear at a keypress. *Blanking* provides this facility by hiding an image behind genlocked video display.

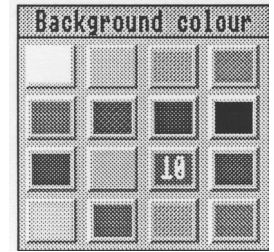
During blanking the image will still be visible on the RGB display, so the mouse pointer is changed as shown to indicate the display state. A normal mouse pointer is visible at other times, but if you don't want it appearing on your video, use *Overlay* to changes its supremacy setting. The *Interval* setting determines the blanking as well as the display time.





menuing over the icon allows this to be changed. Obviously, if each image requires a different colour background, it would be more sensible to add this to the actual drawfile.

To allow simple animation sequences, *!SideShow* normally draws each image on top of the previous. Selecting the *Redraw background* option starts each new image afresh. The background colour defaults to white, but



Scrolling: Issue 2.000

In order to provide our customers with *!SideShow* as quickly as possible as a replacement for Chromatext, it has been necessary to overlook the provision of horizontal and vertical scrolling, although it is fully intended to provide these very soon.

Please ensure you return your product registration cards (enclosed), so that we can automatically send you the full version of the application when it is ready.

Creating Drawfiles fom *!Draw* for use with *!SideShow*

As previously mentioned, it is possible to include any element of a drawfile into *!SideShow*, i.e. text, draw objects and sprites, too.

In order to get the scale of your screens right, it is preferable to work inside a landscape format bounding box. The easiest way to achieve this is to draw yourself a box with the ratio of the horizontal edge to the vertical edge of 4:3 or 5:4, and pretend this is the visible screen area. Zoom up in *!Draw* until the box fills your screen, and then work inside the box. Don't worry about this box having a black outline - you can, of course set the line colour for the box to "none" within *!Draw*, when you're ready to drop the drawfile(s) into *!SideShow*.

Then by using *!SideShow's Scale to Screen* option, the drawfile will be scaled up to fill the output screen as best it can, by fitting the longest edges together. Scaling will be accurate, without loss of aspect ratio, and text will have no problems scaling in proportion because of the nature of RISC OS' outline fonts system.

It would be a good idea to create a master drawfile for a sequence of screens, with the right-sized box ready to be worked in. Remember to save each successive drawfile with a different name, so as not to overwrite the original. In doing this, remember also that sequences are mounted by addressing each drawfile in the directory **in alphabetical order**.

We do not aim to tell you how to go about designing screens in *!Draw*. *!Draw* is a very versatile RISC OS application, with scope for you to exercise your imagination to the full! The best results are those you achieve by experimentation.

Have fun.