Black (RISC OS)
!ChangeFSI JPEG Output Functional Specification

Distribution:	COMPANY CONFIDENTIAL
Title:	Black !ChangeFSI JPEG Output Functional Specification
Drawing Number:	1303,018/FS/3P95
Issue:	1.01/3P95
Author(s):	Graham Simms
Date:	31st January 1995
Change Number:	xxxx
Last Issue:	1

Contents:

History

31/10/94	GS	Issue 1	Started
31/01/95	GS	Issue 1.01	Output dialogue boxes changed after discussion.

Outstanding Issues

None.

Overview

With the introduction of JPEG support within the O/S for Black, it was decided that some facility for the generation of JPEGs was required. The obvious place for this was within the existing !ChangeFSI application which is already in use to convert from many different image formats into Sprites.

The JPEG standard itself does not specify an image format, only a large set of procedures and formats some of which are used by each JPEG image format. The images produced will be in the JFIF image format which is the most widely used JPEG format. We support JFIF version 1.02.

The software will be based on the Independent JPEG Group's JPEG software. This will be modified to allow raw image data to be converted into a JPEG in memory one row at a time, which should minimise the amount of memory required to create the JPEG.

The JPEG compression will be performed by a module which will provide three SWI calls to enable compression of data to a JPEG. The module will be *RMEnsured by !ChangeFSI at run time, with the module being stored in !System.modules.

User Interface

The following changes will be made to the !ChangeFSI user interface.

Icon Bar Menu

The Output option on the icon bar menu has been split into 'Sprite Output' and 'JPEG Output', each leading to a separate dialogue box. See Figure 1. The current output mode is now indicated by a tick next to one of these options.

The fast option is now greyed out when !ChangeFSI is run on a machine with VRAM because it has no effect.

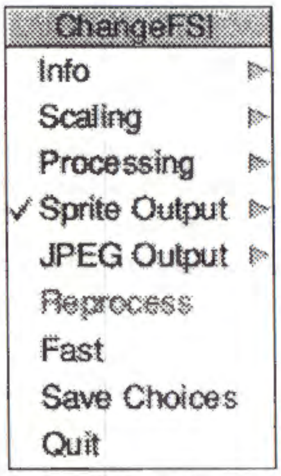

Figure 1

Sprite Output Dialogue Box

This dialogue box has been been redesigned to try and make it more intuitive to use. See Figure 2. Most of the changes are cosmetic, involving the moving and relabelling of the current icons. The 'Mode' icon is no longer writable and an 'Old mode' box has been added in which you can type in a mode number. This makes it much harder for the user to input an invalid mode.

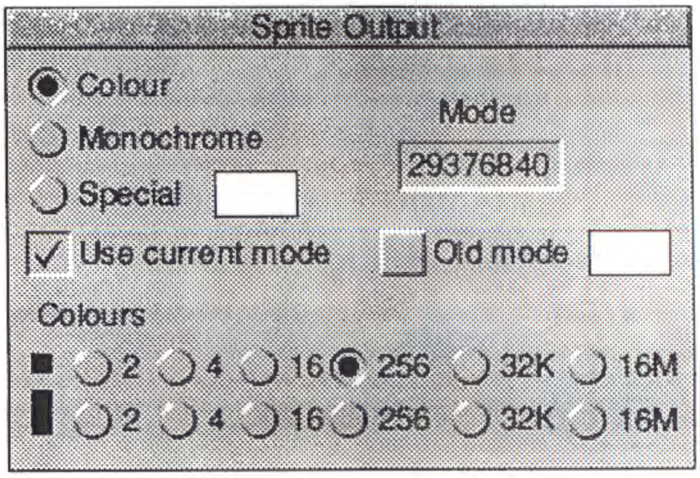

Figure 2

JPEG Output Dialogue Box

This is a new dialogue box which allows the user to set up parameters for conversion to JPEG. See Figure 3. It contains two radio buttons which allow the user to select either colour or monochrome JPEG output and a writable icon to hold a numeric quality value which can be in the range 0 to 100. There is also a set of bump icons which can be used to alter the quality setting.

Selecting any icon in either the 'Sprite Output' or 'JPEG Output' dialogue boxes causes the parent menu item to be ticked. eg if you open the 'JPEG Output' dialogue box and click on the up arrow to increase the quality, the 'JPEG Output' option becomes ticked in the root icon bar menu.

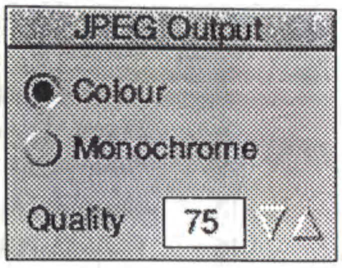

Figure 3

Image Menu

The menu that appears when you click the menu button on top of a displayed image has been alter to reflect the fact that both Sprites and JPEGs can be displayed. All references to Sprites have been changed to refer to images. See Figures 4 and 5.

The 'saveas' box now displays either a Sprite or JPEG icon depending on the type of image to be saved.

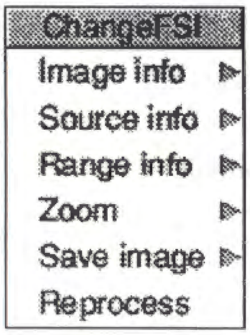

Figure 4

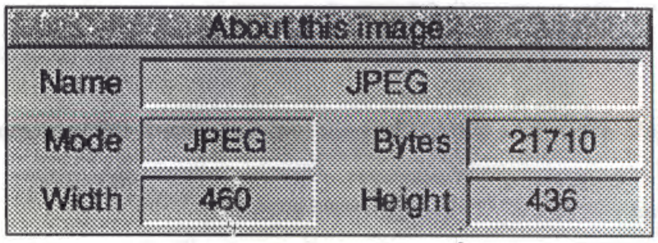

Figure 5

Programmer Interface

!ChangeFSI will compress data into a JPEG by calling SWIs within the CompressJPEG module. This module can also be used by other applications to compress raw image data into a JPEG.

CompressJPEG SWI Interface

The CompressJPEG module supports three SWI calls which can be used together to convert raw data to JPEG. The compression is started by calling the CompressJPEG_Start SWI which sets up the compression environment. Then each row of the source image is compressed with a separate call to the CompressJPEG_WriteLine SWI. Finally the compression is finished by calling the CompressJPEG_Finish SWI. CompressJPEG has been allocated SWI chunk &4A500.

CompressJPEG_Start (SWI &4A500)

This SWI starts the JPEG compression process and sets np various parameters.

On Entry

R0 = Pointer to buffer for JPEG data.

R1 = Size of JPEG data buffer.

R2 = Pointer to block of parameters (see below).

R3 = Pointer to workspace area for JPEG compression.

If R3 = 0 then CompressJPEG module will allocate its own space.

R4 = If R3 != 0 then size of workspace Area.

On Exit

R0 = JPEG Tag to be passed to other SWIs.

Use

The JPEG compression routines cannot guarantee to compress the image by a fixed amount, so it is advisable to give as large a buffer as possible to place the JPEG in whilst compression is in progress.

The following parameter block must be supplied to this SWI:

Offset	Content
0	Width of image in pixels
4	Height of image in pixels
8	Quality value [0-100] - lower quality results in a smaller image.
12	Number of 8bit components in source: 3 for 24bit colour; 1 for 8bit greyscale
16	Horizontal DPI of image, or 0 if unknown
20	Vertical DPI of image, or 0 if unknown

If you don't want CompressJPEG to allocate its own workspace from RMA then you can supply it with a workspace buffer. The amount of workspace required is governed by the following formulae:

For a colour (24bit) image

buffer_size = 20000 + ((image width rounded up to a multiple of 16)*30)

For a greyscale (8bit) image

buffer_size = 20000 + ((image width rounded up to a multiple of 16)*9)

The V flag will be set, and an error returned if the JPEG workspace area becomes full.

CompressJPEG_WriteLine (SWI &4A501)

Compresses one row of source pixels into the JPEG buffer.

On Entry

R0 = JPEG Tag

R1 = Pointer to a row of pixels.

On Exit

All registers preserved.

Use

This SWI should be called once for each row of the source data. The format that should be used is:

For Colour

A buffer of continuous RGB values. ie a byte stream of the format R, G, B, R, G, B ,R....

For Greyscale

A buffer of continuous 8bit gray values.

The V flag will be set, and an error returned if the JPEG buffer becomes full.

CompressJPEG_Finish (SWI &4A502)

Tidies up JPEG buffer.

On Entry

R0 = JPEG Tag.

On Exit

R0 = Size of JPEG image within the buffer.

Use

This call is used to finish writing out JPEG data and return the size of the complete JPEG image.

CompressJPEG *Commands

The CompressJPEG module will have no *Commands.

It will respond in the standard way to '*help CompressJPEG' ie

```
*Help CompressJPEG
==> Help on keyword CompressJPEG
Module is: CompressJPEG    0.03 (03 Nov 1994)
```

External Dependencies

The image is held in memory as a JPEG and therefore requires a JPEG capable version of SpriteExtend to plot it. If !ChangeFSI is run on a pre-Black machine then a blank white window will be displayed.

Development And Acceptance Tests

The new version of !ChangeFSI should continue to work in the same manner when the output is a Sprite. When JPEG output is selected all options will either work in the same manner as for a Sprite, or they will be greyed out.

!ChangeFSI will be tested on RISCOS 3.5, 3.1 and 2. It is only required to work on RISCOS 3.5+ within the scope of Black, but there is a requirement on !ChangeFSI that it continues to run on all major versions of RISCOS.

The Compress JPEG module itself will occupy no more than 40k and will be loaded by !ChangeFSI only when JPEG output is required. The conversion of an image to JPEG will use no more temporary file space, and less memory than the same conversion to Sprite. The only exceptions to this are image formats where the scanlines are stored in the wrong order for conversion to JPEG, in this case the whole source image will have to be buffered.