

Export Module Interface for Adobe Photoshop™

©1990 Thomas Knoll

Introduction

This document describes version 3 of Adobe Photoshop's export module interface. Plug-ins based on the following specifications should contain a 3 in their 'PiMI' resource.

The code resource and file type for export modules is '8BEM'.

Record Structure

The *stuff* parameter contains a pointer to a structure of the following type:

```
ExportRecord = RECORD
    serialNumber:    LONGINT;
    abortProc:       ProcPtr;
    progressProc:    ProcPtr;
    maxData:         LONGINT;
    imageMode:       INTEGER;
    imageSize:       Point;
    depth:           INTEGER;
    planes:          INTEGER;
    imageHRes:       Fixed;
    imageVRes:       Fixed;
    redLUT:          PACKED ARRAY [0..255] OF CHAR;
    greenLUT:        PACKED ARRAY [0..255] OF CHAR;
    blueLUT:         PACKED ARRAY [0..255] OF CHAR;
    theRect:         Rect;
    loPlane:         INTEGER;
    hiPlane:         INTEGER;
    data:            Ptr;
    rowBytes:        LONGINT;
    filename:        Str255;
    vRefNum:         INTEGER;
    dirty:           BOOLEAN;
    selectBBox:      Rect;
END;
```

Calling Order

When the user invokes the plug-in by selecting its name from the Export submenu, Photoshop loads the plug-in's resource into memory and calls it with the following sequence of *selector* values (see the header file for their actual values):

(1) *exportSelectorPrepare*

This allows the plug-in to adjust Photoshop's memory allocation algorithm. Before this call, Photoshop sets the *maxData* field to the maximum number of bytes it would be able to free up. If plug-in module then has the option of reducing this number during this call. Reducing the number can speed up operation, since freeing up the maximum amount of memory requires Photoshop to move all of the image data for any currently open images out of of RAM and into its virtual memory file.

If the plug-in knows that it memory requirements will be limited (if it can process the image data in strips, or if the maximum resolution image it can process is small), it should reduce *maxData* to its actual requirements during this call. This will allow small exports to be performed entirely in RAM.

If, as is often the case, the plug-in only needs a small amount memory, but will operate faster if given more, a tradeoff has to be made. One solution is to divide the *maxData* field by 2, thus allocating half the memory to Photoshop and half to the plug-in.

(2) *exportSelectorStart*

Most plug-ins will display their dialog box, if any, during this call.

During this call, the plug-in should set *theRect* and the *loPlane* and *hiPlane* fields to let Photoshop know what area of the image it wishes to process first.

The total number of bytes requested should be less than *maxData*. If the image is larger than *maxData*, the plug-in must process the image in strips. Horizontal strips are most efficient, but any pattern is allowed (even overlapping ones).

(3) *exportSelectorContinue*

During this routine, the plug-in should process the image data pointed to by the *data* field. It should then adjust *theRect* and the *loPlane* and *hiPlane* fields to let Photoshop know what area of the image it wishes to process next. If the entire image has been processed, it should set *theRect* to an empty rectangle.

The requested image data is pointed to by the *data* field. If more than one plane has been requested (*loPlane* \neq *hiPlane*), the data is interleaved. The offset from one row to the next is indicated by the *rowBytes* field. This is not necessarily equal to the width of *theRect*—there may be additional pad bytes at the end of each row.

(4) *exportSelectorFinish*

This call allows the plug-in to clean up after an image export. This call is made *if and only if* the *exportSelectorStart* routine returns without error (even if the *exportSelectorContinue* routine returns an error).

If Photoshop detects a command-period between calls to the *exportSelectorContinue* routine, it will call the *exportSelectorFinish* routine (be careful here, since normally the plug-in would be expecting another *exportSelectorContinue* call).

Record Fields

(1) *serialNumber*

Contains Adobe Photoshop's serial number. Plug-in modules can use this value for copy protection, if desired.

(2) *abortProc*

Contains a pointer to a function with the following Pascal calling conventions:

```
FUNCTION TestAbort: BOOLEAN;
```

The plug-in may call this function several times a second during long operations to allow the user to abort the operation. If the function returns TRUE, the operations should be aborted. As a side effect, this changes to cursor to a watch, and moves the watch hands periodically.

(3) *progressProc*

Contains a pointer to a procedure with the following Pascal calling conventions:

PROCEDURE UpdateProgress (done, total: LONGINT);

The plug-in may call this two-argument procedure periodically to update a progress indicator. The first parameter is the number of operations completed; the second is the total number of operations.

This procedure should only be called during the actual main operation of the plug-in, not during long operations during the preliminary user interface.

Photoshop automatically suppresses display of the progress graph during short operations.

(4) *maxData*

Photoshop initializes this field to the maximum of number of bytes it can free up. The plug-in may reduce this value during the *exportSelectorPrepare* routine. The *exportSelectorContinue* routine should process the image in strips no larger than *maxData*, less the size of any large tables or scratch areas it has allocated.

(5) *imageMode*

The mode of the image being exported (Gray Scale, RGB Color, etc.). See the header file for possible values. The *exportSelectorStart* should return an *exportBadMode* error if it is unable to process this mode of image.

(6) *imageSize*

The image's width (*imageSize.h*) and height (*imageSize.v*) in pixels.

(7) *depth*

The image's resolution in bits per sample. The only possible settings are 1 for bitmap mode images, and 8 for all other modes.

(8) *planes*

The number of channels in the image. For example, if an RGB image without alpha channels is being processed, this field will be set to 3.

(9) *imageHRes* and *imageVRes*

The image's horizontal and vertical resolution in terms of pixels per inch. These are fixed point numbers (16 binary digits).

(10) *redLUT*, *greenLUT* and *blueLUT*

If an indexed color mode image is being processed, these fields will contain its color table.

(11) *theRect*

The *exportSelectorStart* and *exportSelectorContinue* routines should set this field to request an area of the image for processing. Should be set to an empty rectangle when complete.

(12) *loPlane* and *hiPlane*

The *exportSelectorStart* and *exportSelectorContinue* routines should set these fields to the first and last planes to process next.

(13) *data*

This field contains a pointer to the requested image data. If more than one plane has been requested (*loPlane* \neq *hiPlane*), the data is interleaved.

(14) *rowBytes*

The offset between rows for the requested image data.

(15) *filename*

The name of the file the image was read from. File exporting modules should use this field as the default name for saving.

(16) *vRefNum*

The volume reference number of the file the image was read from.

(17) *dirty*

File exporting modules should clear this field to prevent to the user being prompted to save any unsaved changes when the image is eventually closed.

(18) *selectBBox*

The bounding box of the current selection. If there is no current selection, this is an empty rectangle.