

Fastgraph/Image 6.0

for Windows®

Reference Manual

Copyright © 1996-2003 by Ted Gruber Software, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without express written permission from Ted Gruber Software. The software described in this publication is furnished under a license agreement and may be used or copied only in accordance with the terms of that agreement.

This publication and its associated software are sold without warranties, either expressed or implied, regarding their merchantability or fitness for any particular application or purpose. The information in this publication is subject to change without notice and does not represent a commitment on the part of Ted Gruber Software. In no event shall Ted Gruber Software be liable for any loss of profit or any other commercial damage, including but not limited to special, incidental, consequential, or other damages resulting from the use of or the inability to use this product, even if Ted Gruber Software has been notified of the possibility of such damages.

Fastgraph/Image for Windows version 6.03

Fastgraph® is a registered trademark of Ted Gruber Software, Inc.

Windows®, Direct3D®, DirectDraw®, and DirectX® are registered trademarks of Microsoft Corporation.

All other brand and product names mentioned in this publication are trademarks or registered trademarks of their respective holders.

Table of Contents

Introduction.....	1
fgi_avidone().....	2
fgi_avihead().....	3
fgi_aviopen().....	4
fgi_avipal().....	5
fgi_bmphead().....	6
fgi_bmppal().....	7
fgi_close().....	8
fgi_display().....	9
fgi_filesize().....	11
fgi_flichead().....	12
fgi_flicopen().....	13
fgi_flicskip().....	14
fgi_jpeghead().....	15
fgi_loadpcx().....	16
fgi_open().....	17
fgi_pcxhead().....	18
fgi_pcxpal().....	19
fgi_read().....	20
fgi_readnext().....	21
fgi_seek().....	22
fgi_showavi().....	23
fgi_showbmp().....	25
fgi_showflic().....	27
fgi_showjpeg().....	29
fgi_showpcx().....	30
fgi_showppr().....	32
fgi_showspr().....	33
fgi_version().....	34

Introduction

The *Fastgraph/Image 6.0 Reference Manual* provides an alphabetical summary of Fastgraph/Image functions, with the following information presented for each function:

- function prototypes or declarations for each supported language
- a description of the function itself
- the number of parameters, their purpose, and their data types
- the meaning and data type of the function's return value (if any)
- information about important restrictions pertaining to the function
- references to similar functions, or other functions that affect the function
- example programs in the *Fastgraph/Image 6.0 User's Guide* that use the function

This manual includes information about Fastgraph/Image *legacy functions* in case you encounter them in programs developed with earlier versions of Fastgraph/Image. Legacy functions are still included in Fastgraph/Image, but they have been replaced by other functions and may not be supported in future releases. We therefore recommend that new applications avoid using the legacy functions, and where possible, you eliminate them from existing programs. Legacy functions are identified as such in the function descriptions, and the "Replaced by" section lists the individual function or group of functions with the same or enhanced functionality.

fgi_avidone()

Prototype

```
C/C++  void fgi_avidone (void *Context);  
C#     void fgi.avidone (ref byte Context);  
Delphi procedure fgi_avidone (var Context);  
VB     Sub fgi_avidone (Context() As Any)  
VB.NET Sub fgi_avidone (ByRef Context As Byte)
```

Description

The **fgi_avidone()** function closes the AVI file associated with the specified context descriptor, for an AVI file stored in an FGI library.

Parameters

Context is the name of a 48-byte buffer containing the AVI file context descriptor.

Return value

none

Restrictions

none

See also

fgi_aviopen()

fgi_avihead()

Prototype

```
C/C++  int fgi_avihead (char *FileName, void *Header, int hLib);
C#     int fgi.avihead (string FileName, ref byte Header, int hLib);
Delphi function fgi_avihead (FileName : string; var Header; hLib :
integer) : integer;
VB     Function fgi_avihead (ByVal FileName As String, Header() As
Any, hLib As Long) As Long
VB.NET Function fgi_avihead (ByVal FileName As String, ByRef Header As
Byte, ByVal hLib As Integer) As Integer
```

Description

The **fgi_avihead()** function reads the header of an AVI file stored in an FGI library. Refer to Appendix E of the *Fastgraph 6.0 User's Guide* for details about the AVI header.

Parameters

FileName is the name of the AVI file. It may only include a file name and extension (such as CLOCK.AVI) and must be terminated by a null character (that is, a zero byte).

Header is the name of the buffer to receive the AVI file header. Its size must be at least 56 bytes.

hLib is the file handle for the FGI library containing *FileName*.

Return value

- 0 = Success
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library
- 3 = The requested file is not an AVI file

Restrictions

none

See also

fgi_avipal(), fgi_aviplay(), fgi_avisize(), fgi_showavi()

fgi_aviopen()

Prototype

```
C/C++  int fgi_aviopen (char *FileName, void *Context, int hLib);
C#     int fgi.aviopen (string FileName, ref byte Context, int hLib);
Delphi function fgi_aviopen (FileName : string; var Context; hLib :
integer) : integer;
VB     Function fgi_aviopen (ByVal FileName As String, Context() As
Any, hLib As Long) As Long
VB.NET Function fgi_aviopen (ByVal FileName As String, ByRef Context
As Byte, ByVal hLib As Integer) As Integer
```

Description

The **fgi_aviopen()** function opens an AVI file stored in an FGI library.

Parameters

FileName is the name of the AVI file. It may only include a file name and extension (such as CLOCK.AVI) and must be terminated by a null character (that is, a zero byte).

Context is the name of a 48-byte buffer that will receive the AVI file context descriptor. The descriptor values will only be meaningful if the return value is zero.

hLib is the file handle for the FGI library containing *FileName*.

Return value

- 0 = Success
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library
- 3 = The requested file is not an AVI file
- 4 = Error initializing the AVI video stream
- 5 = Error allocating memory
- 6 = The codec needed for the specified AVI file is not available
- 7 = Error creating or writing to a temporary file

Restrictions

none

See also

fgi_avidone()

Examples

FGIW3

fgi_avipal()

Prototype

```

C/C++  int fgi_avipal (char *FileName, void *Palette, int hLib);
C#     int fgi.avipal (string FileName, ref byte Palette, int hLib);
       int fgi.avipal (string FileName, int NullParam, int hLib);
Delphi  function fgi_avipal (FileName : string; var Palette; hLib :
       integer) : integer;
VB     Function fgi_avipal (ByVal FileName As String, Palette() As
       Any, hLib As Long) As Long
VB.NET  Function fgi_avipal (ByVal FileName As String, ByRef Palette As
       Byte, ByVal hLib As Integer) As Integer
       Function fgi_avipal (ByVal FileName As String, ByVal NullParam
       As Integer, ByVal hLib As Integer) As Integer

```

Description

The **fgi_avipal()** function retrieves the palette of an AVI file stored in an FGI library. The palette values are returned as RGB color components, each between 0 and 255.

Parameters

FileName is the name of the AVI file. It may only include a file name and extension (such as CLOCK.AVI) and must be terminated by a null character (that is, a zero byte).

Palette is the name of the array that will receive the AVI palette values. The palette values are returned as RGB color components, each between 0 and 255. The first three bytes of *Palette* will contain the RGB values for color 0, the next three for color 1, and so forth. The size of the *Palette* array must be at least three times the number of colors in the AVI image. You can also specify NULL for the *Palette* parameter (*nil*^ for Delphi, *ByVal 0* for Visual Basic, or *BYVAL %NULL* for PowerBASIC). In this case **fgi_avipal()** will return the AVI color depth but no palette values.

hLib is the file handle for the FGI library containing *FileName*.

Return value

- >0 = The number of colors in the AVI palette
- 0 = The AVI file does not have a palette (probably a high color or true color AVI file)
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library
- 3 = The requested file is not an AVI file

Restrictions

none

See also

fgi_avihead(), fg_setdacs(), fgi_showavi()

fgi_bmphead()

Prototype

```
C/C++  int fgi_bmphead (char *FileName, void *Header, int hLib);
C#     int fgi.bmphead (string FileName, ref byte Header, int hLib);
Delphi function fgi_bmphead (FileName : string; var Header; hLib :
integer) : integer;
VB     Function fgi_bmphead (ByVal FileName As String, Header() As
Any, ByVal hLib As Long) As Long
VB.NET Function fgi_bmphead (ByVal FileName As String, ByRef Header As
Byte, ByVal hLib As Integer) As Integer
```

Description

The **fgi_bmphead()** function reads the header of a BMP file stored in an FGI library. Refer to Appendix E of the *Fastgraph 6.0 User's Guide* for details about the BMP header.

Parameters

FileName is the name of the BMP file. It may only include a file name and extension (such as PICTURE.BMP) and must be terminated by a null character (that is, a zero byte).

Header is the name of the buffer to receive the BMP file header. Its size must be at least 54 bytes.

hLib is the file handle for the FGI library containing *FileName*.

Return value

- 0 = Success
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library
- 3 = The requested file is not a BMP file

Restrictions

none

See also

fgi_bmppal(), fg_bmpsize(), fgi_showbmp()

fgi_bmppal()

Prototype

```

C/C++  int fgi_bmppal (char *FileName, void *Palette, int hLib);
C#      int fgi_bmppal (string FileName, ref byte Palette, int hLib);
        int fgi_bmppal (string FileName, int NullParam, int hLib);
Delphi  function fgi_bmppal (FileName : string; var Palette; hLib :
        integer) : integer;
VB      Function fgi_bmppal (ByVal FileName As String, Palette() As
        Any, ByVal hLib As Long) As Long
VB.NET  Function fgi_bmppal (ByVal FileName As String, ByRef Palette As
        Byte, ByVal hLib As Integer) As Integer
        Function fgi_bmppal (ByVal FileName As String, ByVal NullParam
        As Integer, ByVal hLib As Integer) As Integer

```

Description

The **fgi_bmppal()** function retrieves the palette of a BMP file stored in an FGI library. The palette values are returned as RGB color components, each between 0 and 255.

Parameters

FileName is the name of the BMP file. It may only include a file name and extension (such as PICTURE.BMP) and must be terminated by a null character (that is, a zero byte).

Palette is the name of the array that will receive the BMP palette values. The palette values are returned as RGB color components, each between 0 and 255. The first three bytes of *Palette* will contain the RGB values for color 0, the next three for color 1, and so forth. The size of the *Palette* array must be at least three times the number of colors in the BMP palette. You can also specify NULL for the *Palette* parameter (*nil*[^] for Delphi, *ByVal 0* for Visual Basic, or *BYVAL %NULL* for PowerBASIC). In this case **fgi_bmppal()** will return the image's color depth but no palette values.

hLib is the file handle for the FGI library containing *FileName*.

Return value

- >0 = The number of colors in the BMP palette
- 0 = The BMP file does not have a palette (probably a 24-bit BMP file)
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library
- 3 = The requested file is not a BMP file

Restrictions

none

See also

[fgi_bmphead\(\)](#), [fg_setdacs\(\)](#), [fgi_showbmp\(\)](#)

fgi_close()

Prototype

```
C/C++  void fgi_close (int hLib);  
C#     void fgi.close (int hLib);  
Delphi procedure fgi_close (hLib : integer);  
VB     Sub fgi_close (ByVal hLib As Long)  
VB.NET Sub fgi_close (ByVal hLib As Integer)
```

Description

The **fgi_close()** function closes an FGI library file previously opened with **fgi_open()**.

Parameters

hLib is the file handle for the FGI library file to close, as returned by **fgi_open()**.

Return value

none

Restrictions

none

See also

fgi_open()

Examples

FGIW1, FGIW2, FGIW3, FGIW4, FGIW5, FGIW6

fgi_display()

Prototype

```

C/C++  int fgi_display (char *FileName, int Flags, int hLib);
C#     int fgi.display (string FileName, int Flags, int hLib);
Delphi  function fgi_display (FileName : string; Flags, hLib : integer)
        : integer;
VB      Function fgi_display (ByVal FileName As String, ByVal Flags As
        Long, ByVal hLib As Long) As Long
VB.NET  Function fgi_display (ByVal FileName As String, ByVal Flags As
        Integer, ByVal hLib As Integer) As Integer

```

Description

The **fgi_display()** function displays a BMP, JPEG, PCX, AVI, flic, or pixel run file stored in an FGI library. If the image is a BMP, JPEG, AVI, or flic file, it will be displayed with its upper left corner at (0,0), unless the *Flags* parameter specifies the image will be displayed relative to the current graphics cursor position. If the image is a PCX file, it will be displayed in the active virtual buffer at the position specified in the file header, unless we specify otherwise through the *Flags* parameter. If it is a pixel run file, its lower left corner will be at the graphics cursor position in the active virtual buffer.

You can use Fastgraph's **fg_imagebuf()** function to specify the size and address of the internal buffer when displaying images with **fgi_display()**. The default internal buffer size is 4,096 bytes. Image display is typically faster when a larger buffer is used.

Parameters

FileName is the name of the image file to display. It may only include a file name and extension (such as PICTURE.PCX) and must be terminated by a null character (that is, a zero byte).

Flags is a bit mask that controls how the image is displayed if it is a BMP, JPEG, PCX, AVI, or flic image (this parameter is ignored for pixel run images). Please refer to the descriptions of the **fgi_showbmp()**, **fgi_showjpeg()**, **fgi_showpcx()**, **fgi_showavi()**, and **fgi_showflic()** functions for the meanings of the individual flag bits as they apply to each image type.

hLib is the file handle for the FGI library containing *FileName*.

Return value

- 0 = Success
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library
- 3 = The requested file is not a BMP, JPEG, PCX, AVI, flic, or pixel run file

Restrictions

A logical palette must be defined and realized in order to use the palette values stored in an image file.

AVI and FLI/FLC images will play one time. If you need to play such files more than once, use **fgi_showavi()** or **fgi_showflic()**.

fgi_display() (continued)

See also

`fg_imagebuf()`, `fgi_read()`, `fgi_showavi()`, `fgi_showbmp()`, `fgi_showflic()`, `fgi_showjpeg()`,
`fgi_showpcx()`

Examples

FGIW1

fgi_filesize()

Prototype

```
C/C++   long fgi_filesize (char *FileName, int hLib);
C#      int fgi.filesize (string FileName, int hLib);
Delphi  function fgi_filesize (FileName : string; hLib : integer) :
        longint;
VB      Function fgi_filesize (ByVal FileName As String, ByVal hLib As
        Long) As Long
VB.NET  Function fgi_filesize (ByVal FileName As String, ByVal hLib As
        Integer) As Integer
```

Description

The **fgi_filesize()** function returns the size in bytes of a file stored in an FGI library. If successful, **fgi_filesize()** leaves the FGI file pointer positioned at the start of the requested file.

Parameters

FileName is the name of the file of interest. It may only include a file name and extension (such as PICTURE.PCX) and must be terminated by a null character (that is, a zero byte).

hLib is the file handle for the FGI library containing *FileName*.

Return value

- >0 = The size of the requested file in bytes
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library

Restrictions

none

See also

fgi_read(), fgi_readnext(), fgi_seek()

Examples

FGIW4, FGIW6

fgi_flichead()

Prototype

C/C++ `int fgi_flichead (char *FileName, void *Header, int hLib);`
C# `int fgi.flichead (string FileName, ref byte Header, int hLib);`
Delphi `function fgi_flichead (FileName : string; var Header; hLib : integer) : integer;`
VB `Function fgi_flichead (ByVal FileName As String, Header() As Any, ByVal hLib As Long) As Long`
VB.NET `Function fgi_flichead (ByVal FileName As String, ByRef Header As Byte, ByVal hLib As Integer) As Integer`

Description

The **fgi_flichead()** function reads the header of an FLI or FLC file stored in an FGI library. Refer to Appendix E of the *Fastgraph 6.0 User's Guide* for details about the FLI/FLC header.

Parameters

FileName is the name of the FLI/FLC file. It may only include a file name and extension (such as MOVIE.FLI) and must be terminated by a null character (that is, a zero byte).

Header is the name of the buffer to receive the FLI/FLC file header. Its size must be at least 128 bytes.

hLib is the file handle for the FGI library containing *FileName*.

Return value

- 0 = Success
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library
- 3 = The requested file is not an FLI or FLC file

Restrictions

none

See also

`fg_flicplay()`, `fg_flicsize()`, `fg_showflic()`

fgi_flicopen()

Prototype

```

C/C++  int fgi_flicopen (char *FileName, void *Context, int hLib);
C#     int fgi.flicopen (string FileName, ref byte Context, int hLib);
Delphi function fgi_flicopen (FileName : string; var Context; hLib :
integer) : integer;
VB     Function fgi_flicopen (ByVal FileName As String, Context() As
Any, ByVal hLib As Long) As Long
VB.NET Function fgi_flicopen (ByVal FileName As String, ByRef Context
As Byte, ByVal hLib As Integer) As Integer

```

Description

The **fgi_flicopen()** function opens an FLI or FLC file (collectively called flic files) stored in an FGI library for subsequent processing by other low-level flic file support functions. If successful, the file pointer will be positioned at the beginning of the first frame.

Parameters

FileName is the name of the FLI/FLC file. It may only include a file name and extension (such as MOVIE.FLI) and must be terminated by a null character (that is, a zero byte).

Context is the name of a 20-byte buffer that will receive the flic file context descriptor. The descriptor values will only be meaningful if the return value is zero.

hLib is the file handle for the FGI library containing *FileName*.

Return value

- 0 = Success
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library
- 3 = The requested file is not an FLI or FLC file

Restrictions

none

See also

fg_flicplay(), fgi_flicskip(), fgi_showflic()

Examples

FGIW2

fgi_flicskip()

Prototype

```
C/C++  int fgi_flicskip (void *Context, int nFrames);
C#     int fgi.flicskip (ref byte Context, int nFrames);
Delphi function fgi_flicskip (var Context; nFrames : integer) :
integer;
VB     Function fgi_flicskip (Context() As Any, ByVal nFrames As Long)
As Long
VB.NET Function fgi_flicskip (ByRef Context As Byte, ByVal nFrames As
Integer) As Integer
```

Description

The **fgi_flicskip()** function advances one or more frames in a flic file that was previously opened with **fgi_flicopen()**.

Parameters

Context is the name of a 20-byte buffer containing the flic file context descriptor.

nFrames is the number of frames to skip in the flic file, starting from the current file position. If *nFrames* is negative, the flic file position will be set to the first frame.

Return value

The number of frames skipped. This value may be less than *nFrames* if the end-of-file is reached before skipping the requested number of frames. If *nFrames* is negative, the return value will be zero.

Restrictions

none

See also

fg_flicplay(), fgi_flicopen()

Examples

FGIW2

fgi_jpeghead()

Prototype

```
C/C++  int fgi_jpeghead (char *FileName, void *Header, int hLib);
C#     int fgi.jpeghead (string FileName, ref byte Header, int hLib);
Delphi function fgi_jpeghead (FileName : string; var Header; hLib :
integer) : integer;
VB     Function fgi_jpeghead (ByVal FileName As String, Header() As
Any, ByVal hLib As Long) As Long
VB.NET Function fgi_jpeghead (ByVal FileName As String, ByRef Header
As Byte, ByVal hLib As Integer) As Integer
```

Description

The **fgi_jpeghead()** function reads the header of a JPEG file stored in an FGI library. Strictly speaking, JPEG files do not have formal headers, but **fgi_jpeghead()** returns relevant information from the file's start of frame segment. We call it a header for consistency with other image file formats. Refer to Appendix E of the *Fastgraph 6.0 User's Guide* for details about the JPEG header.

Parameters

FileName is the name of the JPEG file. It may only include a file name and extension (such as PICTURE.JPG) and must be terminated by a zero byte.

Header is the name of the 10-byte buffer to receive the JPEG file header.

hLib is the file handle for the FGI library containing *FileName*.

Return value

- 0 = Success
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library
- 3 = The requested file is not a baseline JPEG file

Restrictions

none

See also

fg_jpegsizes(), fgi_showjpeg()

fgi_loadpcx()

Prototype

```
C/C++  int fgi_loadpcx (char *FileName, int Flags, int hLib);
C#     int fgi.loadpcx (string FileName, int Flags, int hLib);
Delphi function fgi_loadpcx (FileName : string; Flags, hLib : integer)
      : integer;
VB     Function fgi_loadpcx (ByVal FileName As String, ByVal Flags As
      Long, ByVal hLib As Long) As Long
VB.NET Function fgi_loadpcx (ByVal FileName As String, ByVal Flags As
      Integer, ByVal hLib As Integer) As Integer
```

Description

The **fgi_loadpcx()** legacy function displays a PCX file stored in an FGI library. It is equivalent to the **fgi_showpcx()** function and is provided for compatibility with the DOS version of Fastgraph/Image.

256-color PCX files are reduced to the 236 non-system colors if color reduction is enabled. 16-color and monochrome PCX files are always remapped to colors 10 to 25 to avoid conflicts with the system colors.

Parameters

FileName is the name of the PCX file. It may only include a file name and extension (such as PICTURE.PCX) and must be terminated by a null character (that is, a zero byte).

Flags is a series of flags that controls how the image is displayed. Refer to the description of **fgi_showpcx()** for the meanings of the flags.

hLib is the file handle for the FGI library containing *FileName*.

Return value

- 0 = Success
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library
- 3 = The requested file is not a PCX file
- 4 = The PCX file cannot be loaded into the active virtual buffer
- 5 = Error allocating memory

Restrictions

A logical palette must be defined and realized in order to use the palette values stored in the PCX file.

24-bit PCX files can only be loaded into direct color virtual buffers.

Replaced by

fgi_showpcx()

fgi_open()

Prototype

```
C/C++  int fgi_open (char *FileName);
C#     int fgi.open (string FileName);
Delphi function fgi_open (FileName : string) : integer;
VB     Function fgi_open (ByVal FileName As String) As Long
VB.NET Function fgi_open (ByVal FileName As String) As Integer
```

Description

The **fgi_open()** function opens an FGI library file and returns a file handle through which the files in that library can be accessed. For information about creating and maintaining FGI libraries, refer to Chapter 3 of the *Fastgraph/Image 6.0 User's Guide*.

Parameters

FileName is the name of the FGI library to open. A device and path name may be included as part of the FGI library's file name. It must be terminated by a null character (that is, a zero byte).

Return value

If the FGI library is opened successfully, **fgi_open()** returns the file handle through which other Fastgraph/Image functions reference the FGI library. If **fgi_open()** cannot open *FileName*, it returns zero.

Restrictions

none

See also

fgi_close()

Examples

FGIW1, FGIW2, FGIW3, FGIW4, FGIW5, FGIW6

fgi_pcxhead()

Prototype

```
C/C++  int fgi_pcxhead (char *FileName, void *Header, int hLib);
C#     int fgi.pcxhead (string FileName, ref byte Header, int hLib);
Delphi function fgi_pcxhead (FileName : string; var Header; hLib :
integer) : integer;
VB     Function fgi_pcxhead (ByVal FileName As String, Header() As
Any, ByVal hLib As Long) As Long
VB.NET Function fgi_pcxhead (ByVal FileName As String, ByRef Header As
Byte, ByVal hLib As Integer) As Integer
```

Description

The **fgi_pcxhead()** function reads the header of a PCX file stored in an FGI library. Refer to Appendix E of the *Fastgraph 6.0 User's Guide* for details about the PCX header.

Parameters

FileName is the name of the PCX file. It may only include a file name and extension (such as PICTURE.PCX) and must be terminated by a null character (that is, a zero byte).

Header is the name of the buffer to receive the PCX file header. Its size must be at least 128 bytes.

hLib is the file handle for the FGI library containing *FileName*.

Return value

- 0 = Success
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library
- 3 = The requested file is not a PCX file

Restrictions

none

See also

fgi_pcxpal(), fgi_pcxrange(), fgi_pcxsize(), fgi_showpcx()

fgi_pcxpal()

Prototype

```

C/C++  int fgi_pcxpal (char *FileName, void *Palette, int hLib);

C#     int fgi.pcxpal (string FileName, ref byte Palette, int hLib);
       int fgi.pcxpal (string FileName, int NullParam, int hLib);

Delphi function fgi_pcxpal (FileName : string; var Palette; hLib :
       integer) : integer;

VB     Function fgi_pcxpal (ByVal FileName As String, Palette() As
       Any, ByVal hLib As Long) As Long

VB.NET Function fgi_pcxpal (ByVal FileName As String, ByRef Palette As
       Byte, ByVal hLib As Integer) As Integer
       Function fgi_pcxpal (ByVal FileName As String, ByVal NullParam
       As Integer, ByVal hLib As Integer) As Integer

```

Description

The **fgi_pcxpal()** function retrieves the palette of a PCX file stored in an FGI library. The palette values are returned as RGB color components, each between 0 and 255.

If the PCX file includes an extended (256-color) palette, **fgi_pcxpal()** will return the values in the extended palette. Otherwise, **fgi_pcxpal()** will return the values from the 16-color palette in the PCX header.

Parameters

FileName is the name of the PCX file. It may only include a file name and extension (such as PICTURE.PCX) and must be terminated by a null character (that is, a zero byte).

Palette is the name of the array that will receive the PCX palette values. The palette values are returned as RGB color components, each between 0 and 255. The first three bytes of *Palette* will contain the RGB values for color 0, the next three for color 1, and so forth. The size of the *Palette* array must be at least three times the number of colors in the PCX palette. You can also specify NULL for the *Palette* parameter (*nil*^ for Delphi, *ByVal 0* for Visual Basic, or *BYVAL %NULL* for PowerBASIC). In this case **fgi_pcxpal()** will return the image's color depth but no palette values.

hLib is the file handle for the FGI library containing *FileName*.

Return value

- >0 = The number of colors in the PCX palette (16 or 256)
- 0 = The PCX file does not have a palette (24-bit PCX file)
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library
- 3 = The requested file is not a PCX file

Restrictions

none

See also

fgi_pcxhead(), fg_setdacs(), fgi_showpcx()

fgi_read()

Prototype

```
C/C++   int fgi_read (char *FileName, void *Buffer, int hLib);
C#      int fgi.read (string FileName, ref byte Buffer, int hLib);
Delphi  function fgi_read (FileName : string; var Buffer; hLib :
integer) : integer;
VB      Function fgi_read (ByVal FileName As String, Buffer() As Any,
ByVal hLib As Long) As Long
VB.NET  Function fgi_read (ByVal FileName As String, ByRef Buffer As
Byte, ByVal hLib As Integer) As Integer
```

Description

The **fgi_read()** function reads the contents of a file stored in an FGI library into an array or allocated memory block.

Parameters

FileName is the name of the file to retrieve. It may only include a file name and extension (such as PICTURE.PCX) and must be terminated by a null character (that is, a zero byte).

Buffer is the name of the array or memory block that will receive the file contents.

hLib is the file handle for the FGI library containing *FileName*.

Return value

- 0 = Success
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library

Restrictions

none

See also

fgi_readnext()

Examples

FGIW4, FGIW5

fgi_readnext()

Prototype

```

C/C++  int fgi_readnext (void *Buffer, unsigned int nSize, int hLib);
C#     int fgi.readnext (ref byte Buffer, int nSize, int hLib);
Delphi  function fgi_readnext (var Buffer; nSize, hLib : integer) :
        integer;
VB     Function fgi_readnext (Buffer() As Any, ByVal nSize As Long,
        ByVal hLib As Long) As Long
VB.NET  Function fgi_readnext (ByRef Buffer As Byte, ByVal nSize As
        Integer, ByVal hLib As Integer) As Integer

```

Description

The **fgi_readnext()** function reads the contents of the “next” file stored in an FGI library into an array or allocated memory block. The “next” file is only meaningful immediately after calling **fgi_filesize()** or **fgi_seek()**.

Parameters

Buffer is the name of the array or memory block that will receive the file contents.
nSize is the number of bytes to read from the FGI file.
hLib is the file handle for the FGI library.

Return value

0 = Success
-1 = Error reading the FGI library
-2 = The number of bytes read was less than *nSize*

Restrictions

You can only use **fgi_readnext()** immediately after calling **fgi_filesize()** or **fgi_seek()** for the same FGI library.

See also

fgi_filesize(), fgi_read(), fgi_seek()

Examples

FGIW6

fgi_seek()

Prototype

```
C/C++  int fgi_seek (char *FileName, int hLib);
C#     int fgi.seek (string FileName, int hLib);
Delphi  function fgi_seek (FileName : string; hLib : integer) :
        integer;
VB     Function fgi_seek (ByVal FileName As String, ByVal hLib As
        Long) As Long
VB.NET  Function fgi_seek (ByVal FileName As String, ByVal hLib As
        Integer) As Integer
```

Description

The **fgi_seek()** function positions the FGI file pointer at the beginning of the requested file. This function is called internally by other Fastgraph/Image functions and is not usually called directly by applications.

Parameters

FileName is the name of the file of interest. It may only include a file name and extension (such as PICTURE.PCX) and must be terminated by a null character (that is, a zero byte).

hLib is the file handle for the FGI library containing *FileName*.

Return value

- 0 = Success
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library

Restrictions

none

See also

fgi_readnext()

fgi_showavi()

Prototype

```

C/C++  int fgi_showavi (char *FileName, int nCount, int Flags, int
        hLib);

C#     int fgi.showavi (string FileName, int nCount, int Flags, int
        hLib);

Delphi  function fgi_showavi (FileName : string; nCount, Flags, hLib :
        integer) : integer;

VB     Function fgi_showavi (ByVal FileName As String, ByVal nCount As
        Long, ByVal Flags As Long, hLib As Long) As Long

VB.NET  Function fgi_showavi (ByVal FileName As String, ByVal nCount As
        Integer, ByVal Flags As Integer, ByVal hLib As Integer) As
        Integer

```

Description

The **fgi_showavi()** function plays an AVI file stored in an FGI library. Each frame in the AVI file is played in the active virtual buffer and then scaled to fit the client area.

Parameters

FileName is the name of the AVI file. It may only include a file name and extension (such as CLOCK.AVI) and must be terminated by a null character (that is, a zero byte).

nCount is the number of times to play the AVI image. If count is zero, the AVI plays continuously. You can stop the AVI play at any time by pressing the Escape key.

Flags is a series of flags that controls how the AVI is played:

Flag	Meaning
FG_AT_XY	If specified, play the AVI file relative to the current graphics position. If not, play it relative to (0,0).
FG_IGNOREAVIPALETTE	If specified, play the AVI file using the current palette. If not, use the palette values stored in the AVI file. Not meaningful for high color or true color AVI files.
FG_NODELAY	If specified, play the AVI file with no delay between frames. If not, delay between frames as specified in the AVI header.

hLib is the file handle for the FGI library containing *FileName*.

Return value

- 0 = Success
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library
- 3 = The requested file is not an AVI file
- 4 = Error initializing the AVI video stream
- 5 = Error allocating memory
- 6 = The codec needed for the specified AVI file is not available
- 7 = Error creating or writing to a temporary file

fgi_showavi() (continued)

Restrictions

When using DirectX, the active virtual buffer must not be locked.

See also

`fgi_avihead()`, `fgi_avipal()`, `fg_aviplay()`, `fg_avisize()`, `fgi_display()`, `fg_realize()`

Examples

FGIW3

fgi_showbmp()

Prototype

```
C/C++  int fgi_showbmp (char *FileName, int Flags, int hLib);
C#     int fgi.showbmp (string FileName, int Flags, int hLib);
Delphi function fgi_showbmp (FileName : string; Flags, hLib : integer)
       : integer;
VB     Function fgi_showbmp (ByVal FileName As String, ByVal Flags As
       Long, ByVal hLib As Long) As Long
VB.NET Function fgi_showbmp (ByVal FileName As String, ByVal Flags As
       Integer, ByVal hLib As Integer) As Integer
```

Description

The **fgi_showbmp()** function displays a BMP file stored in an FGI library.

For 256-color virtual buffers, 256-color BMP files are reduced to the 236 non-system colors if color reduction is enabled. 16-color and monochrome BMP files are always remapped to colors 10 to 25 to avoid conflicts with the system colors.

Parameters

FileName is the name of the BMP file. It may only include a file name and extension (such as PICTURE.BMP) and must be terminated by a null character (that is, a zero byte).

Flags is a series of flags that controls how the image is displayed:

Flag	Meaning
FG_AT_XY	If specified, display the image relative to the current graphics position. If not, display the image relative to (0,0).
FG_IGNOREPALETTE	If specified, display the image using the current palette. If not, use the palette values stored in the BMP file. Not meaningful for 24-bit BMP files.
FG_KEEPCOLORS	If specified, disable color reduction and remapping. If not, enable color reduction and remapping to avoid using the Windows system colors. Not meaningful for 24-bit BMP files.

hLib is the file handle for the FGI library containing *FileName*.

Return value

- 0 = Success
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library
- 3 = The requested file is not a BMP file
- 4 = The BMP file cannot be loaded into the active virtual buffer
- 5 = The BMP file is an unsupported RLE BMP file
- 6 = Error allocating memory

fgi_showbmp() (continued)

Restrictions

A logical palette must be defined and realized in order to use the palette values stored in the BMP file.

24-bit BMP files can only be loaded into direct color virtual buffers.

See also

`fgi_bmphead()`, `fgi_bmppal()`, `fg_bmpsize()`, `fgi_display()`, `fg_makebmp()`, `fg_realize()`

Examples

FGIW5

fgi_showflic()

Prototype

```

C/C++  int fgi_showflic (char *FileName, int nCount, int Flags, int
        hLib);

C#     int fgi.showflic (string FileName, int nCount, int Flags, int
        hLib);

Delphi  function fgi_showflic (FileName : string; nCount, Flags, hLib :
        integer) : integer;

VB     Function fgi_showflic (ByVal FileName As String, ByVal nCount
        As Long, ByVal Flags As Long, ByVal hLib As Long) As Long

VB.NET  Function fgi_showflic (ByVal FileName As String, ByVal nCount
        As Integer, ByVal Flags As Integer, ByVal hLib As Integer) As
        Integer

```

Description

The **fgi_showflic()** function plays an FLI or FLC file (collectively called flic files) stored in an FGI library. Each frame in the flic file is played in the active virtual buffer and then scaled to fit the client area. For 256-color virtual buffers, flic files are reduced to the 236 non-system colors if color reduction is enabled.

Parameters

FileName is the name of the flic file. It may only include a file name and extension (such as MOVIE.FLI) and must be terminated by a null character (that is, a zero byte).

nCount is the number of times to play the flic file. If count is zero, the flic plays continuously. You can stop the flic play at any time by pressing the Escape key.

Flags is a series of flags that controls how the image is played:

Flag	Meaning
FG_AT_XY	If specified, play the flic file relative to the current graphics position. If not, play it relative to (0,0).
FG_IGNOREFLICPALETTE	If specified, play the flic file using the current palette. If not, use the palette values stored in the flic file.
FG_KEEPCOLORS	If specified, disable color reduction and remapping. If not, enable color reduction and remapping to avoid using the Windows system colors.
FG_NODELAY	If specified, play the flic file with no delay between frames. If not, delay between frames as specified in the flic header.

hLib is the file handle for the FGI library containing *FileName*.

Return value

- 0 = Success
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library
- 3 = The requested file is not an FLI or FLC file

fgi_showflic() (continued)

Restrictions

A logical palette must be defined and realized in order to use the palette values stored in the flic file.

When using DirectX, the active virtual buffer must not be locked.

See also

`fgi_display()`, `fgi_flichead()`, `fg_flicplay()`, `fg_flicsize()`, `fg_realize()`

Examples

FGIW2

fgi_showjpeg()

Prototype

```

C/C++  int fgi_showjpeg (char *FileName, int Flags, int hLib);
C#     int fgi.showjpeg (string FileName, int Flags, int hLib);
Delphi  function fgi_showjpeg (FileName : string; Flags, hLib :
integer) : integer;
VB     Function fgi_showjpeg (ByVal FileName As String, ByVal Flags As
Long, ByVal hLib As Long) As Long
VB.NET  Function fgi_showjpeg (ByVal FileName As String, ByVal Flags As
Integer, ByVal hLib As Integer) As Integer

```

Description

The **fgi_showjpeg()** function displays a JPEG file stored in an FGI library. The JPEG image can be either grayscale or color, but it must be a baseline JPEG file. Baseline JPEG files use Huffman encoding and cannot use the progressive, hierarchical, or lossless compression and storage modes.

Parameters

FileName is the name of the JPEG file. It may only include a file name and extension (such as PICTURE.JPG) and must be terminated by a zero byte.

Flags is a series of flags that controls how the image is displayed:

Flag	Meaning
FG_AT_XY	If specified, display the image relative to the current graphics position. If not, display the image relative to (0,0).

hLib is the file handle for the FGI library containing *FileName*.

Return value

- 0 = Success
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library
- 3 = The requested file is not a JPEG file
- 4 = The specified file is not a baseline JPEG file or does not have a valid JPEG structure
- 5 = Error allocating memory

Restrictions

This function is meaningful only with direct color virtual buffers.

See also

fgi_display(), fgi_jpeghead(), fg_jpegsize()

fgi_showpcx()

Prototype

```
C/C++  int fgi_showpcx (char *FileName, int Flags, int hLib);
C#     int fgi.showpcx (string FileName, int Flags, int hLib);
Delphi function fgi_showpcx (FileName : string; Flags, hLib : integer)
       : integer;
VB     Function fgi_showpcx (ByVal FileName As String, ByVal Flags As
       Long, ByVal hLib As Long) As Long
VB.NET Function fgi_showpcx (ByVal FileName As String, ByVal Flags As
       Integer, ByVal hLib As Integer) As Integer
```

Description

The **fgi_showpcx()** function displays a PCX file stored in an FGI library.

256-color PCX files are reduced to the 236 non-system colors if color reduction is enabled. 16-color and monochrome PCX files are always remapped to colors 10 to 25 to avoid conflicts with the system colors.

Parameters

FileName is the name of the PCX file. It may only include a file name and extension (such as PICTURE.PCX) and must be terminated by a null character (that is, a zero byte).

Flags is a series of flags that controls how the image is displayed:

Flag	Meaning
FG_AT_XY	If specified, display the image relative to the current graphics position. If not, display the image at the position indicated in PCX header.
FG_IGNOREPALETTE	If specified, display the image using the current palette. If not, use the palette values stored in the PCX file. Not meaningful for 24-bit PCX files.
FG_KEEPCOLORS	If specified, disable color reduction and remapping. If not, enable color reduction and remapping to avoid using the Windows system colors. Not meaningful for 24-bit PCX files.

hLib is the file handle for the FGI library containing *FileName*.

Return value

- 0 = Success
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library
- 3 = The requested file is not a PCX file
- 4 = The PCX file cannot be loaded into the active virtual buffer
- 5 = Error allocating memory

Restrictions

A logical palette must be defined and realized in order to use the palette values stored in the PCX file.

24-bit PCX files can only be loaded into direct color virtual buffers.

fgi_showpcx() (continued)

See also

`fgi_display()`, `fg_makepcx()`, `fgi_pcxhead()`, `fgi_pcxpal()`, `fg_pcxrange()`, `fg_pcxsize()`, `fg_realize()`

fgi_showppr()

Prototype

```
C/C++  int fgi_showppr (char *FileName, int hLib);
C#     int fgi.showppr (string FileName, int hLib);
Delphi function fgi_showppr (FileName : string; hLib : integer) :
       integer;
VB     Function fgi_showppr (ByVal FileName As String, ByVal hLib As
       Long) As Long
VB.NET Function fgi_showppr (ByVal FileName As String, ByVal hLib As
       Integer) As Integer
```

Description

The **fgi_showppr()** legacy function displays a packed pixel run (PPR) file stored in an FGI library. The image will be positioned so that its lower left corner is at the graphics cursor position in the active virtual buffer.

Parameters

FileName is the name of the PPR file. It may only include a file name and extension (such as PICTURE.PPR) and must be terminated by a null character (that is, a zero byte).

hLib is the file handle for the FGI library containing *FileName*.

Return value

- 0 = Success
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library
- 3 = The requested file is not a PPR file

Restrictions

none

Replaced by

BMP and PCX display functions

fgi_showspr()

Prototype

C/C++ `int fgi_showspr (char *FileName, int hLib);`
C# `int fgi.showspr (string FileName, int hLib);`
Delphi `function fgi_showspr (FileName : string; hLib : integer) : integer;`
VB `Function fgi_showspr (ByVal FileName As String, ByVal hLib As Long) As Long`
VB.NET `Function fgi_showspr (ByVal FileName As String, ByVal hLib As Integer) As Integer`

Description

The **fgi_showspr()** legacy function displays a standard pixel run (SPR) file stored in an FGI library. The image will be positioned so that its lower left corner is at the graphics cursor position in the active virtual buffer.

Parameters

FileName is the name of the SPR file. It may only include a file name and extension (such as PICTURE.SPR) and must be terminated by a null character (that is, a zero byte).

hLib is the file handle for the FGI library containing *FileName*.

Return value

- 0 = Success
- 1 = Error reading the FGI library
- 2 = The requested file is not in the FGI library
- 3 = The requested file is not an SPR file

Restrictions

none

Replaced by

BMP and PCX display functions

fgi_version()

Prototype

```
C/C++  void fgi_version (int *Major, int *Minor);
C#     void fgi.version (out int Major, out int Minor);
Delphi procedure fgi_version (var Major, Minor : integer);
VB     Sub fgi_version (Major As Long, Minor As Long)
VB.NET Sub fgi_version (ByRef Major As Integer, ByRef Minor As
Integer)
```

Description

The **fgi_version()** function returns the major and minor version numbers for your copy of Fastgraph/Image for Windows. For example, if you are using version 6.00, the major version number is 6 and the minor version number is 0.

Parameters

Major receives the major version number.

Minor receives the minor version number, expressed in hundredths.

Return value

none

Restrictions

none