Autodesk®
Scaleform®

GFxExport Reference Guide

This document describes the GFxExport utility tool which preprocesses SWF files into an optimized GFx format for use with Scaleform.

Author: Dmitry Polenur

Version: 2.0
Last Edited: May 25, 2011

Autodesk®
GAMEWARE

Copyright Notice

Autodesk® Scaleform® 4.2

How to Contact Autodesk Scaleform:

Document	GFxExport Reference Guide
Address	Autodesk Scaleform Corporation 6305 Ivy Lane, Suite 310 Greenbelt, MD 20770, USA
Website	www.scaleform.com
Email	info@scaleform.com
Direct	(301) 446-3200
Fax	(301) 446-3199

Table of Contents

1 Introduction

This document describes the GFxExport utility tool in Autodesk® Scaleform®. GFxExport helps accelerate the loading of Flash content by converting it into GFx files which is then loaded into the application.

GFxExport is a command line utility that processes SWF files into a format that is optimized for streamlined loading. During preprocessing, resources are stripped out and extracted into separate files. Textures can be compressed into a more efficient format. Once stripped and converted, the Flash files will have the .gfx extension and are only compatible with Scaleform. The GFx files will then rely on the external data files, which can be managed by the application's resource system.

The GFxExport utility tool has various options for preprocessing the SWF files into different formats which is described in detail in Section 2. When used with these different options, GFxExport can export font textures, compress images, load shared copy of images and so on.

2 Usage

The GFxExport tool is prebuilt for Win32 and located in the bin directory in your SDK. A typical path would be:

C:\Program Files\Scaleform\GFx SDK 4.2 \Bin\gfxexport.exe

GFxExport is used in the following way:

```
gfxexport [file.swf] [file(s)...] [options]
```

For the filenames, wildcards are supported, such as *.swf.

To see the full list of command line options and usage instructions, execute gfxexport.exe without any additional arguments and help output will be printed.

2.1 Image Options

Images can be converted to DDS files with DXT texture compression for optimized loading and run-time memory savings. The ability to use external DDS files is the one of the most important advantages of GFx files over SWF files.

-i <format>
Specifies output format for exporting image data where <format> is one of the following:

- TGA - Truevision (Targa or TGA) (default).
- DDS - DirectDraw Surface (DDS).
- PVR – PVR compression format.
- ETC- Ericsson compression format.
- ORIG - Original format. If the original image format is a lossless bitmap then a TGA file will be produced. If the original image format is JPEG then a JPG file will be produced without re-compression; in the case of JPEG with alpha-channel, both JPG/TGA files will be produced.

-strip_images
Just strip images, do not write them to files.

-share_images
Tries to reuse images in the destination directory rather than write new ones every time.

–replace_images
Replace internal (embedded in SWF) images with external files based on the image linkage ID. For example, if image has linkage ID “myimage.png”, it will be replaced with the image in *myimage.png* file. When used with *–pack* option, the replaced images will also be packed by default; however they can be excluded from packing if linkage ID has "*-nopack*" in it (i.e. "*myimage.png –nopack*").

-id
Specifies the directory for the external images.

-rescale <nearest | hi | low | nextlow | mult4> [filter]
Optional filtering for rescaling (must follow mandatory rescale mode option):

- *Box*
- *Triangle*
- *Quadratic*
- *Cubic*
- *Catrom*
- *Mitchell*
- *Gaussian*
- *Sinc*
- *Bessel*
- *Hanning*
- *Hamming*
- *Blackman*
- *Kaiser*

-pack (Scaleform 3.1 and above)
Pack images into larger textures.

-packsize<size> (Scaleform 3.1 and above)
Set the maximum size for packed textures. Default value is 1024.

-ptresize<no | p2 | mult4 | mult128 > (Scaleform 3.1 and above)
Resize the packed textures. Used with the following options:

- *no* - No resize
- *p2* - Resize to a higher power of 2 (default)
- *mult4* –Resize to a higher multiple of 4.
- *mult128* - Resize to a higher multiple of 128.

-pad
Pad textures to power of 2 or multiple of 4 (set by *ptresize).* This option is the alias for *-pack -packsize 2.*

Note: To exclude images from packed texture, add ***-nopack*** to Linkage identifier of the image in the Flash Studio.

2.1.1 DDS related options (' –i DDS' specified)

-d0
Write uncompressed DDS

-d1c
Use DXT1 for RGB data without alpha channel.

-d1a
Use DXT1 for RGB data with alpha channel.

-d3/-d5
Use DXT3 (default) or DXT5 for RGB data with alpha channel.

-qf, -quick
Fast compression method.

-qn, -quality_normal
Normal quality compression (default).

-qp, -quality_production
Production quality compression.

-qh, -quality_highest
Highest quality compression (this can be very slow).

Note that starting from Scaleform 3.2, options for mipmaps has been changed. Mipmaps are not generated by default (except for font textures).

-mipmap
Generate mipmap levels in DDS file (by default, mipmaps are generated only for fonts).

-mipfilter < Box | Triangle | Kaiser>
Select mipmap filter (Box is default).

-fnomipmaps
Do not generate mipmaps for fonts.

2.1.2 PVR related options (' –i PVR' specified)

-pv2
Use 2 bit per pixel compression.

-pv4
Use 4 bit per pixel compression.

Note that these options are available only for supported mobile and handheld platforms.

2.2 Files and Directories Related Options

-d <dirname>
Set the destination directory for exported data files. If not specified, files are stored in the directory containing the SWF.

-sd
Create subdirectories for each SWF file using the SWF filename. Extracted files are placed in the corresponding subdirectories.

-c
Write compressed stripped .gfx file(s). In practice many developers compress multiple files generated by gfxexport within their own package system, using custom GFx::FileOpener to load them as necessary.

-o
Specify the directory to write .gfx files.

-p <prefix>
Specifies prefix to add to the name of each exported resource. By default, the original SWF filename is used as prefix.

-lwr / -upr
Force all exported files to have lowercase / uppercase names.

-ne
Tell gfxexport not to use export names when image file names are being formed.

-modstamp
Do not run if SWF file is older that existing gfx.

2.3 Font Options

GFxExport can be used to compress font vector data more efficiently and to reduce memory usage. It can also be used to pre-generate font textures, although this is not recommended except for low-end systems. Please refer to our Font Configuration Overview document for more information about the Scaleform font system.

-fonts
Export font textures. If not specified, font textures will not be generated (allowing for either dynamic cache or packing at load time to be done instead).

-fntlst
Export font list and textfield/font map (.fnt-file).

-fc
Use compact fonts

-fcl <size>
Set the normalized size of compact fonts (default size is 256).

-fcm
Merge edges for compact fonts.

-fns <size>
Set the nominal size of texture glyph in pixels (defaults to 48 if no size is specified). The nominal size is the maximum size of one glyph in a texture. Smaller characters are rendered at runtime by using tri-linear mip-map filtering.

-fpp <n>
Specifies the space, in pixels, to leave around the individual glyph image. The default value is 3.

-fts <WxH>
Set the dimensions of the textures that the glyphs get packed into. Default size is 256x256. To specify a square texture, only one dimension can be specified, e.g.: '-fts 128' is 128x128. '-fts 512x128' specifies rectangle texture.

-fs
Force separate textures for each font. By default, fonts share textures.

-strip_font_shapes
Do not write font shapes in resulting GFX file.

-fi <format>
Specifies output format for font textures, where <format> is one of the following:
TGA8 - 8-bit Targa TrueVision (grayscale)
TGA24 - 24-bit Targa TrueVision (grayscale)
TGA32 - 32-bit Targa TrueVision
DDS8 - 8-bit DDS A8
By default, if image format (-i option) is TGA then TGA8 is used for font textures; otherwise DDS A8 is used.

Please note, that -fns, -fpp, -fts, -fs, -fi should be used only with -fonts option

2.4 Gradient Options

-gradients
Export gradient images.

-grs <size>
Set the size of radial gradient image as <size> by <size> pixels. Default size is 64x64.

-nogsh
Do not share gradient images. By default, gradient images are sharable across the SWF file.

-gi <format>
Specifies the output format for gradient textures, where <format> is one of the following:
TGA - 32-bit Targa TrueVision
DDS32 - 32-bit uncompressed DDS
DDS – Uses the same DDS settings as for images (see "DDS Options"). By default, if the image format (-i option) is TGA, then TGA is used for gradient textures; if the image format is compressed DDS, then compressed DDS is used; if image format is uncompressed DDS, then DDS32 is used.

-gsid <dir1 dir2...>
Specifies additional directories to compare against when using -share_images.

-gd <path>
Specifies where gradient images should be written (and compared against, if -share_images is specified)

Please note, that -grs, -gi, -gsid, -gd should be used only with -gradient option

2.5 Info Options

-q
Quiet mode (suppress output).

-x
Quiet progress mode (suppress progress output).
-list
Save list of generated files.

-info
Get information about exported images, but do not export them.

2.5.1 TextField Options

-det
Export list of unique default values of dynamic/input textfields to the text file with .det extensions. Text file will be saved in UTF-8 encoding. Default text field values may be useful for localization.

-ddt
The same as -det option except it does not save values of input textfields. The output files will have .ddt extension.

2.5.2 Fscommand Options

-fstree
Export list of fscommands as a tree (.fst-file)

-fslist
Export list of fsommands as a sorted list (.fsl-file)

-fsparams
Save parameters of fscommands. This option works with -fstree and -fslist options and will save parameters of fscommands to .fst or .fsl file. However it will work only for hardcoded parameters.

2.6 Sound Options

-s <extension>
Specifies the extension for exporting sound files.

3 FAQs Related to GFxExport

This section contains few of the questions that developers may have when using the GFxExport tool.

1. What are the advantages of using GFX files over using SWF files?

The main advantage is the ability to use external DDS (compressed texture) files. When SWF file is loaded, it typically contains embedded images in JPEG format internally; these JPEGs need to be decompressed before being used for rendering - a step that can easily turn 20K of data into 2M or so, depending on the target image size. With compressed textures, however, you can save 4x on video memory use - an amount that is significant (consider 20M used for images vs. 5M). In addition, JPEG decompression takes a lot of processing power during loading, although this cost can certainly be compared against the alternative - loading larger DDS files from disk. In the end, video memory considerations typically win for most developers.

For additional benefit, GFxExport includes some options that can be used to compress font vector data more efficiently to save RAM, such as "*-fc*" (font compaction). This is primarily significant for Asian fonts, as the SWF vector format is already reasonably compact for many cases. GFxExport can also be used to pre-generate font textures or gradients, although this is not recommended except for low-end systems.

2. Why is my GFx file larger than the original SWF?

SWF files are compressed by default. You can use "*-c*" option to produce compressed GFx files. In practice, however, many developers often compress multiple GFx-generated files within their own package system or zip, using a custom GFx::FileOpener to load them as necessary.

3. GFxExport incorrectly exports text with special effects (glow, shadow, etc.). Why does this occur?

You should not use "*-fonts*" option if you want to use effects like glow, shadow and blur. Please refer to the "Font and Text Configuration Overview" for more details.

4. I have my own texture format that my engine uses. How can I use GFx files with images in this format?

After the export is finished, you can have your own custom tools on to convert the exported images into your own custom format.

After images are stored in your own texture format, you have to install the GFx::ImageCreator class to load those textures into Scaleform. A GFx::ImageCreator::CreateImage virtual function will get called for every texture file. Here you can load your custom images, wrap them in Render::ImageInfo and return for Scaleform to use.

5. ***How can I make textures to be exported with names I want?***

You can set the name you want GFxExport to use in Linkage Properties of the image.

6. ***Our pack tool relies on images named [filename of .swf]_[ID#], how can we make GFxExport use this name format even for images that are tagged "Export for ActionScript"?***

Use " *–ne*" option.

7. **How do I exclude image from packed texture?**

You can add *–nopack* to Linkage Identifier of the image in the Flash Studio.

8. ***How can we avoid re-exporting textures that we know did not change from the last export?***

You can make these files read-only.

9. ***GFxExport takes too long to process our files. Is there any way to speed it up?***

Default DDS compression quality is "normal". Using "*-qf*" option ("fast" quality setting) will significantly reduce processing time.

10. ***Where can we find more information on GFxExport?***

You can find additional information on GFxExport in the related forum topic on the Developer Section of our website. Also, refer to the "Art and Assets" and "Using Custom Images, Icons and Dynamic Textures" sections of our FAQ.