

Getting Started with GfX 3.3

This document explains how to get up and running quickly with Scaleform GfX 3.3.

Authors: Matthew Doyle
Version: 1.10
Last Edited: June 9, 2011

Copyright Notice

Autodesk® Scaleform® 3

© 2011 Autodesk, Inc. All rights reserved. Except as otherwise permitted by Autodesk, Inc., this publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose.

Certain materials included in this publication are reprinted with the permission of the copyright holder.

The following are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries: 3DEC (design/logo), 3December, 3December.com, 3ds Max, Algor, Alias, Alias (swirl design/logo), AliasStudio, Alias|Wavefront (design/logo), ATC, AUGI, AutoCAD, AutoCAD Learning Assistance, AutoCAD LT, AutoCAD Simulator, AutoCAD SQL Extension, AutoCAD SQL Interface, Autodesk, Autodesk Intent, Autodesk Inventor, Autodesk MapGuide, Autodesk Streamline, AutoLISP, AutoSnap, AutoSketch, AutoTrack, Backburner, Backdraft, Beast, Built with ObjectARX (logo), Burn, Buzzsaw, CAiCE, Civil 3D, Cleaner, Cleaner Central, ClearScale, Colour Warper, Combustion, Communication Specification, Constructware, Content Explorer, Dancing Baby (image), DesignCenter, Design Doctor, Designer's Toolkit, DesignKids, DesignProf, DesignServer, DesignStudio, Design Web Format, Discreet, DWF, DWG, DWG (logo), DWG Extreme, DWG TrueConvert, DWG TrueView, DXF, Ecotect, Exposure, Extending the Design Team, Face Robot, FBX, Fempro, Fire, Flame, Flare, Flint, FMDesktop, Freewheel, GDX Driver, Green Building Studio, Heads-up Design, Heidi, HumanIK, IDEA Server, i-drop, Illuminate Labs AB (design/logo), ImageModeler, iMOUT, Incinerator, Inferno, Inventor, Inventor LT, Kynapse, Kynogon, LandXplorer, LiquidLight, LiquidLight (design/logo), Lustre, MatchMover, Maya, Mechanical Desktop, Moldflow, Moldflow Plastics Advisers, MPI, Moldflow Plastics Insight, Moldflow Plastics Xpert, Moondust, MotionBuilder, Movimento, MPA, MPA (design/logo), MPX, MPX (design/logo), Mudbox, Multi-Master Editing, Navisworks, ObjectARX, ObjectDBX, Opticore, Pipeplus, PolarSnap, PortfolioWall, Powered with Autodesk Technology, Productstream, ProMaterials, RasterDWG, RealDWG, Real-time Roto, Recognize, Render Queue, Retimer, Reveal, Revit, RiverCAD, Robot, Scaleform, Scaleform AMP, Scaleform CLIK, Scaleform GFx, Scaleform IME, Scaleform Video, Showcase, Show Me, ShowMotion, SketchBook, Smoke, Softimage, Softimage|XSI (design/logo), Sparks, SteeringWheels, Stitcher, Stone, StormNET, StudioTools, ToolClip, Topobase, Toxik, TrustedDWG, U-Vis, ViewCube, Visual, Visual LISP, Volo, Vtour, WaterNetworks, Wire, Wiretap, WiretapCentral, XSI.

All other brand names, product names or trademarks belong to their respective holders.

Disclaimer

THIS PUBLICATION AND THE INFORMATION CONTAINED HEREIN IS MADE AVAILABLE BY AUTODESK, INC. "AS IS". AUTODESK, INC. DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE REGARDING THESE MATERIALS.

How to Contact Scaleform:

| | |
|----------|---|
| Document | Getting Started with GFx 3.3 |
| Address | 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 | Welcome..... | 1 |
| 2 | Installation..... | 2 |
| 2.1 | Flash Versions | 2 |
| 2.2 | SDK Installation | 3 |
| 2.2.1 | The SDK Browser | 3 |
| 2.2.2 | Scaleform CLIK Installation | 3 |
| 2.2.3 | Scaleform Video Installation | 4 |
| 2.2.4 | Scaleform IME Installation | 5 |
| 2.2.5 | GFx 3.3 for UE3 Installation..... | 5 |
| 2.2.6 | GFx 3.3 for Gamebryo Installation..... | 5 |
| 2.2.7 | DirectX SDK | 5 |
| 2.2.8 | .Net Framework..... | 5 |
| 2.2.9 | Directory Structure | 5 |
| 2.2.10 | Scaleform SDK for Windows | 8 |
| 2.3 | Console Platform Installations | 10 |
| 2.3.1 | GFx SDK for Xbox 360 | 10 |
| 2.3.2 | GFx SDK for PS3..... | 12 |
| 2.3.3 | GFx SDK for Wii | 18 |
| 2.4 | Installing the Scaleform Launcher Panel for CS3..... | 22 |
| 2.5 | Installing the Scaleform Launcher Panel for CS4..... | 25 |
| 2.6 | Using the Scaleform Launcher Panel..... | 28 |
| 2.7 | Licensing | 36 |
| 2.7.1 | License Problems | 36 |
| 2.8 | Support | 38 |
| 2.9 | Feedback | 38 |
| 3 | What is Included? | 39 |
| 3.1 | Learning Flash | 39 |
| 3.2 | Getting Started Guides..... | 40 |
| 3.3 | Development Documents | 40 |

| | | |
|----------|----------------------------------|-----------|
| 3.4 | Platform-Specific Documents..... | 41 |
| 3.5 | Demos and Examples..... | 42 |
| 3.6 | Online Documents..... | 43 |
| 4 | Where to Begin | 43 |
| 4.1 | User-Specific Information..... | 44 |
| 4.1.1 | UI Programmers | 44 |
| 4.1.2 | UI Designers..... | 44 |
| 4.1.3 | UI Artists..... | 44 |
| 4.2 | Playing Files | 45 |
| 4.3 | Starting a New Project..... | 46 |

1 Welcome

Welcome to Scaleform® GfX™ 3.3. This guide explains what Scaleform GfX 3.3 is, installation of GfX 3.3, licensing and where to look for additional information. GfX 3.0 and higher versions includes several features designed to improve workflow, testing and debugging, and iteration time as well as adding new core features such as the Scaleform Common Lightweight Interface Kit (CLIK™) and video.

Scaleform GfX Features and Add-ons:

- Scaleform CLIK
- Scaleform Video, powered by CRI Movie™
- Scaleform Analyzer for Memory and Performance (AMP™)
- Scaleform Input Method Editor (IME)
- Memory enhancements
- Garbage collection

2 Installation

The Scaleform GfX 3.3 SDK comes with a variety of add-ons such as Scaleform Video and Scaleform IME that can be downloaded and installed in addition to the base package. These additional products and options are available from: <https://developer.scaleform.com/gfx?action=dl>

2.1 Flash Versions

Scaleform GfX 3.3 requires Adobe® Flash® Creative Suite® 3 (CS3) or newer. GfX 3.3 supports a broad range of Flash and ActionScript™ (AS) features. While GfX 3.3 does not support every Flash or AS feature, the rare occurrence of an unsupported feature has little to no impact on development. GfX 3.3 supports ActionScript 2.0 specifically, and not AS 3.0; however, Flash CS3 and CS4 can compile and run AS 2.0 as well as AS 3.0. For a complete list of GfX 3.3 supported Flash and AS 2.0 features, please see the document entitled *Scaleform GfX 3.3 – Flash Support Overview*.

Adobe has done a tremendous amount of work turning the Creative Suite into a unified suite of products which complement one another. This allows the end user to work seamlessly from application to application within the Creative Suite and provides a smoother workflow than was possible before. In some cases however, the newer features and workflow of CS4 may not be needed, and, as such, it may not be a good decision to upgrade from CS3. Obviously, ensuring that each artist and developer are using the same versions of Flash, Photoshop, Illustrator etc will help alleviate workflow strain caused by file version issues.

2.2 SDK Installation

The Scaleform GfX 3.3 SDK on Windows platform will by default be installed to: *C:/Program Files/Scaleform/GfX SDK 3.3/*

If GfX 3.3 is installed to a different location, please make note of the location and take the new location into account when reading through the documentation included with the SDK. The latest DirectX® SDK from Microsoft may need to be installed if it isn't already; however, the DirectX SDK is only necessary if compiling the GfX 3.3 Player. For testing of the prebuilt GfX 3.3 Player, only the DirectX runtime is needed. Once installation is complete, the Windows® Start menu will include links to commonly used SDK features, demos, tutorials and documents. For additional help, see the documentation in the Scaleform->GfX SDK 3.3->Documentation folder in the Windows Start menu, or via Windows Explorer: *C: /Program Files/Scaleform/GfX SDK 3.3/Doc/*

The GfX 3.3 SDK can be extracted to any desired location on Macintosh/ Linux and includes platform-specific read me instructions on compiling and playing the sample players included in the SDK. Please refer to the *readme_make.txt* and the specific *readme_xx.txt* files for more information on the configurations and the compiler options.

2.2.1 The SDK Browser

Scaleform GfX 3.3 includes an SDK Browser, which provides quick access to demos, tutorials and documentation, as well as other important GfX 3.3 items.

On Windows, the SDK Browser can be accessed via the Start Menu: *Programs->Scaleform->GfX SDK 3.3->Scaleform SDK Browser* or the shortcut added on the Desktop. On Mac, the SDK Browser can be accessed by going to *scaleform_gfx_3.3_macos/Bin/Browser/* and double clicking Browser.app.

2.2.2 Scaleform CLIK Installation

GfX 3.0 and higher versions comes with the Scaleform Common Lightweight Interface Kit (CLIK), which provides an optimized component framework for rapid user interface (UI) development. While the necessary files for CLIK are copied onto the computer during the SDK installation, CLIK requires additional installation steps to integrate it into the Flash authoring environment. CLIK installation and use instructions may be found in the *Getting Started with CLIK* guide, accessible in a variety of ways.

1. Via the Windows Start Menu shortcut:
Start->Programs->Scaleform->GfX SDK 3.3->Documentation->Scaleform GfX 3.3 – Getting Started with CLIK.pdf

2. Via Windows Explorer:
C:/Program Files/Scaleform/GFx SDK 3.3/Doc/gfx_3.3_getting_started_with_CLIK.pdf
3. Via the SDK Browser.

The files necessary for CLIK can all be found here in a default GFx SDK Installation:

On Windows:

*C:/Program Files/Scaleform/GFx SDK 3.3/Resources/CLIK/
C:/Program Files/Scaleform/GFx SDK 3.3/Resources/Tools/*

On other platforms:

*scaleform_gfx_3.3_platform/Resources/CLIK/
scaleform_gfx_3.3_platform/Resources/Tools/*

2.2.3 Scaleform Video Installation

Scaleform Video powered by CRI Movie is a premium module (separately purchased from Scaleform) that must be downloaded and installed after the base GFx 3.3 SDK has been installed. Scaleform Video provides highly optimized video support within Flash files. Once Scaleform Video has been downloaded and installed, it will be found in the same directory as the GFx 3.3 SDK installation. The Scaleform Video Encoder icon will be added to both the desktop and the Windows Start menu inside the Scaleform->GFx SDK 3.3->Video folder. For more information on Scaleform Video, please see the document *Getting Started with Video*.

Location of Video Encoder files: *C:/Program Files/Scaleform/GFx SDK 3.3/Bin/VideoEncoder/*

Location of Video Demo files: *C:/Program Files/Scaleform/GFx SDK 3.3/Bin/VideoDemo/*

Location of Video Documentation:

1. *C:/Program Files/Scaleform/GFx SDK 3.3/Doc/gfx_3.3_getting_started_with_video.pdf*
2. Start->Programs->Scaleform->GFx SDK 3.3->Documentation->Scaleform GFx 3.3 – Getting Started with Video.pdf
3. SDK Browser.

2.2.4 Scaleform IME Installation

Scaleform Input Method Editor (IME) is an add-on product that allows for multi-language character input support for Flash files. It must be downloaded and installed separately after the core GFx 3.3 SDK has been installed. It will be installed by default to: *C:/Program Files/Scaleform/GFx SDK 3.3/Bin/IME/*

2.2.5 GFx 3.3 for UE3 Installation

Licensed Unreal® Engine 3 (UE3) customers, who have registered with Scaleform to receive the UE3 engine integration package, can download it from the Scaleform website and install it. It may be necessary to merge some files into UE3 as well. For information on using the Scaleform GFx 3.3 UE3 integration, please see the documents *Unreal Workflow* and *Unreal Engine 3 Integration Overview*.

2.2.6 GFx 3.3 for Gamebryo Installation

Licensed Emergent Gamebryo® customers, who have registered with Scaleform to receive the Gamebryo integration package, can download it from the Scaleform website and install it. For information on using the Scaleform GFx 3.3 Gamebryo integration, please see the document *Gamebryo LightSpeed Integration Overview*.

2.2.7 DirectX SDK

For best results on Windows, we recommend installing the latest DirectX SDK from Microsoft. Download the latest SDK from <http://msdn.microsoft.com/en-us/directx/default.aspx>

2.2.8 .Net Framework

.NET Framework 2.0 SP1 is required to use CRI Movie tools for Scaleform Video. Download and install *.NET Framework 2.0 SP1* from [Microsoft .NET Framework Version 2.0 SP1](#)

2.2.9 Directory Structure

The GFx installation has a number of directories containing various subfolders and files such as examples, documentation, source, code, libraries, binaries, etc. Some directories may be specific to a

particular platform and configuration. Here is a description of the major directories which comprise the GfX installation:

| | |
|----------------|--|
| 3rdParty/ | 3rd party packages |
| Apps/ | |
| Demos | Demo application source files |
| Common | Common source files for most demos. Platform-specific setup and input handling |
| FxPlayer | Our main Flash player. Displays memory usage and performance counters. |
| GfXPlayerTiny | The simplest GfX application; a starting point for development |
| RenderTexture | Samples which render Flash to a texture and embed a user's texture into Flash. |
| Bin/ | Various binary files (executables, samples, etc). |
| [Platform] | Binary executables (GfXPlayer, samples, etc) for a specific platform |
| FxPlayer | GfXPlayer data files |
| RenderTexture | RenderTexture samples files |
| SWFTToTexture | RenderTexture samples files |
| Samples | Sample Flash files |
| VideoDemo | GfX video sample files |
| Doc/ | Documentation |
| Include/ | GfX include files (C++ headers) |
| Lib/ | GfX libraries |
| [Platform] | Libraries for all configurations |
| [Config] | Libraries specific to a particular platform and configuration |
| Obj/ | Intermediate build files |
| [Platform] | Platform- specific object files |
| Projects/ | Build system support files (projects, make files, etc) |
| Common | Common make files |
| [Platform] | Platform specific make files |
| Src/ | GfX source code |
| GRenderer | Source for sample renderers (included with all releases). |
| GSoundRenderer | Source for sample sound renderers (included with all releases). |

GfX platform names and #defines

| GfX Platform | Platform Names | #defines |
|---------------------|-----------------------|------------------------------|
| Win32 | Windows | GFC_OS_WIN32 |
| x64 | Windows for x86_64 | GFC_OS_WIN32 |
| *-linux | Linux for * processor | GFC_OS_LINUX |
| MacOS | MacOS X | GFC_OS_DARWIN, GFC_OS_MAC |
| WinCE | WinCE, WinMo | GFC_OS_WINCE |
| iPhone | iPhone | GFC_OS_IPHONE |
| psp | Playstation Portable | GFC_OS_PSP |
| PS3 | Playstation 3 | GFC_OS_PS3 |
| wii | Wii | GFC_OS_WII |
| Xbox360 | Xbox 360 | GFC_OS_XBOX360 |

Build Tools

| | Standard, often gcc |
|--------|----------------------------|
| Msvc80 | Visual Studio 8.0 (2005) |
| Msvc90 | Visual Studio 9.0 (2008) |
| CW | Codewarrior |
| snc | SN Systems |

[Platform] refers to one of the names in the first column above in the Directory structure, a slash, and an appropriate build tool; for example Win32/Msvc80. If the “standard” build tools are used, there is no suffix; for example MacOS. Most platforms support only one of those build tools. When project files are supplied, they are always placed in a build tool directory even if the “standard” tools use the same underlying compiler.

2.2.10 Scaleform SDK for Windows

The Scaleform GFx library is supported on Windows® XP (32 bit, SP3), Vista (32/64 bit, SP2) and Windows 7 (32/64 bit). It is fully compatible with Visual Studio 2005 (8.0) and Visual Studio 2008 (9.0) (Visual Studio 2010 support coming soon).

2.2.10.1 Scaleform Lib Descriptions

Linking your product against the correct Scaleform GFx precompiled libs ensures smooth integration with the latest version of Scaleform GFx.

Scaleform GFx SDK precompiled libs are located by default in:

\Program Files\Scaleform\GFx SDK 3.3\Lib\[Platform]\[MSVC Version]\

The four sub folders under this directory describe the runtime library the libs were compiled against.

| | |
|--------------------------|---|
| Debug | Scaleform GFx Multi-threaded Debug libs |
| Runtime Library | Multi-threaded Debug (/MTd) |
| Debug Information Format | C7 Compatible (/Z7) |
| | Non optimized debug code |

| | |
|--------------------------|---|
| DebugOpt | Scaleform GFx Multi-threaded Optimized Debug libs |
| Runtime Library | Multi-threaded (/MT) |
| Debug Information Format | C7 Compatible (/Z7) |
| Optimization | Full Optimization (/Ox) |
| | Optimized debug code |

| | |
|-----------------|---|
| Release | Scaleform GFx Multi-threaded Release libs |
| Runtime Library | Multi-threaded (/MT) |
| Optimization | Full Optimization (/Ox) |
| | Optimized release code |

| | |
|-----------------|--|
| Shipping | Scaleform GFx Multi-threaded Shipping libs |
| Runtime Library | Multi-threaded (/MT) |
| Optimization | Full Optimization (/Ox) |
| | Optimized release code with no logging, stats, etc |

2.2.10.2 MSVC Project Settings

To properly execute the Scaleform GfX sample demos (such as **GfXPlayerSWFToTexture**, and **GfXPlayerTextureInSWF**) you must change the "Working Directory" of the project to the corresponding data directory.

To set the working directory, do the following:

1. Click on **Project** in the main menu
2. Click on **Properties** (last option)
3. Change **Configuration** combo box to All Configurations
4. Click on **Debugging** item
5. Paste in the appropriate **Working Directory**

Example:

GfXPlayerSWFToTexture working directory is:

C:\Program Files (x86)\Scaleform\GFx SDK 3.3\Bin\SWFToTexture

GfXPlayerTextureInSWF working directory is:

C:\Program Files (x86)\Scaleform\GFx SDK 3.3\Bin\RenderTexture

2.3 Console Platform Installations

Licensed console developers who have registered with Scaleform may also download and install the console extensions for any of the consoles they are registered to use. Simply download the additional files for each console, and follow the installation instructions. Installing the console package will add additional directories into the default GFx 3.3 SDK installation directory. These folders will contain the necessary source code, examples and documentation needs for each platform. See the section in this document describing the GFx directory structure for more information.

Specific installation procedures for Xbox360®, Wii™ and PS3™ are provided in the following subsections.

2.3.1 GFx SDK for Xbox 360

Xbox 360 users may download the source and lib packages of GFx SDK from the Developer Section of the website. This section explains how to install and run the sample GFx players on Xbox 360.

2.3.1.1 Requirements

This software needs to be installed on your development computer for the proper operation of GFx.

- OS Windows XP/Vista
- VS2005 or VS2008 (2008 recommended)
- XDK versions, November 2008 or March 2009 or June 2009

2.3.1.2 Installation

First install the GFx SDK for Windows followed by the GFx SDK for Xbox 360. Listed below are the packages for Windows and Xbox360 that need to be installed.

For evaluation purposes, install the latest versions of these eval packages:

- gfx_3.3_eval_{msvc80 or 90}.exe
- gfx_3.3_eval_xbox360_{msvc80 or 90}.exe

Licensed source users should install these src packages:

- gfx_3.3_src_{msvc80 or 90}.exe
- gfx_3.3_src_xbox360_{msvc80 or 90}.exe

Licensed library (non-source) users should install these packages:

- gfx_3.3_lib_{msvc80 or 90}.exe
- gfx_3.3_lib_xbox360_{msvc80 or 90}.exe

Optional packages (eval or library) for IME and Video add-ons:

- IME: gfx_3.3_{eval or lib}_ime_{msvc80 or 90}.exe
- Video:
 - gfx_3.3_{eval or lib}_video_{msvc80 or 90}.exe
 - gfx_3.3_{eval or lib}_video_xbox360_{msvc80 or 90}.exe

For instructions on where to place the license key, see the section in this document on Licensing.

2.3.1.3 Building and Running the Players/Samples

After installing the GFx Xbox 360 SDK, open the GFx 3.3 Xbox 360 Demos Visual Studio solution, accessible via the Start Menu and the GFx SDK Browser. The solution is available for both Visual Studio 2005 (Msvc80) and Visual Studio 2008 (Msvc90).

Please ensure that your Xbox 360 Development Kit or Test Kit is powered on and connected to your network before building the demos. Within Visual Studio, select the configuration you'd like to build from the drop down menu on the toolbar at the top of the screen. Available configurations include Debug, DebugOpt, and Release.

You are now ready to build the GFx demo projects. To do so, select "Build" from Visual Studio's top menu and select "Build Solution". This will initiate a build of all the projects within the solution. When compiling and linking of the executables completes, by default the solution will deploy the executables and all necessary content (particularly GFX and SWF files) to your Xbox Development Kit.

If your Xbox is a Development Kit, you can run the demos directly from Visual Studio. To define which project to launch on the Xbox, right-click the project and select "Set as StartUp Project". You can now launch the executable by selecting "Start Debugging" from the "Debug" drop down on the Visual Studio menu (you can also start debugging by clicking the green arrow beside where you defined your configuration or by pressing F5). You should now see the demo running on your Xbox. Please note that in the GFx 3.3 Xbox 360 Demos solution, only GFxPlayer, GFxPlayerSWFTToTexture, GFxPlayerTextureInSWF, and GFxPlayerTiny can be set as start up projects.

If your Xbox is a Test Kit, you must launch the demos from the Xbox. The Xbox Launcher should now include those projects you built in its master list. Simply select the project you wish to launch using an Xbox controller and press the controller's A button to launch the demo. Again, please note that you will only see GFxPlayer, GFxPlayerSWFTToTexture, GFxPlayerTextureInSWF, and GFxPlayerTiny

displayed in the Xbox Launcher because the other projects included in the solution (GFx_Xbox360, GFxExpat, and GSound) are not executables.

2.3.1.4 Building the Source

If you have the GFx source installation (for both Win32 and Xbox 360), you can build the GFx SDK itself. Here's how to build the GFx library for Xbox360 from the source code:

1. Build the third party solution. This step should only be performed once before the first build of GFx library. Open {GFX_SDK}/Projects/Xbox360/{msvc80 or 90}/SDK/GFx 3.3 Xbox 360 3rd Party.sln, build the solution (selecting the appropriate configuration such as Release) and then close it.
2. Open {GFX_SDK}/Projects/Xbox360/{Msvc80|Msvc90}/SDK/GFx 3.3 Xbox 360 SDK.sln solution.
3. Select the desired configuration and choose Build Solution from the build menu.

Build Settings:

The following compiler preprocessor defines are set for Xbox 360 builds:

- `_XBOX` is defined for all configurations
- `_DEBUG` is defined for Debug and DebugOpt configurations
- `NODEBUG` is defined for Release configuration.

Compiler options for RTTI (Run Time Type Information) and C++ exception handling option are disabled for Xbox360 builds.

2.3.2 GFx SDK for PS3

Before installing the GFx SDK on PS3, it is recommended that users read the readme text file provided with the SDK.

Cell SDK 250 or newer is required for sound support with FMOD. GNU make version 3.81 is required (unless using Visual Studio for building).

Visual Studio projects are also supplied, for use with ProDG VSI. The PS3 SNC release does not support Visual Studio at this time.

The renderer libs `librenderer_gl.a` and `librenderer_ps3.a` are supplied prebuilt and contain the compiled shaders. If you want to manually rebuild the renderer, or are using Visual Studio, then the compiled shaders are provided as extra libs. If the renderer libs are built with Visual Studio, they will not include the compiled shaders.

Compiled shaders for the libgcm renderer are stored in *Lib/PS3/Shaders.a*.

Compiled shaders for the PSGL renderer are stored in *Lib/PS3/GLShaders.a*.

The compiled shaders can only be rebuilt with make.

2.3.2.1 Installation

The Windows packages for GfX (and optional add-ons GfX IME and GfX Video) should be first installed by using the supplied installers. After installation is done for Windows, PS3 src and lib packages should be installed in the same location in the file system where the Windows packages were installed.

If installing an evaluation version of GfX SDK, please refer to the section in this document on Licensing, for details on where to place the license key.

After installing an add-on such as IME or Video, it will automatically be enabled in demo applications after they are rebuilt with make. The demo applications in GfX SDK include: GfXPlayer, GfX PlayerSWFTToTexture, etc. These prebuilt demo applications included in the base (without add-ons) release were built with the add-ons enabled, but if you build them again, the resulting executable will no longer have the functionality provided by the add-ons.

If using Visual Studio, the projects for the demo applications will need to be modified to use the add-on. Each add-on will require preprocessor symbols to be defined and a library to be linked against:

| Preprocessor symbol(s) | Library |
|------------------------|---|
| GFC_USE_IME | -lgfx_ime |
| GFC_USE_IME_PS3 | \$(SCE_PS3_ROOT)\target\ppu\lib\libsutil_imejp_stub.a |
| GFC_USE_VIDEO | -lgfx_video |

2.3.2.2 Renderer libs

Unlike older versions, prebuilt renderer libs are provided and are named as `Lib/<platform>/libgrenderer_<renderer>.a`

These can usually be linked into an application and used without being rebuilt.

2.3.2.3 Building GFx using make

Before building GFx with make, you should have a Linux tool set installed, including the make command itself. We recommend **cygwin** which can be downloaded and installed as follows:

1. Go to <http://www.cygwin.com/>, download setup.exe and run it
2. Choose 'Install from Internet'
3. Set the root directory to whatever you want or use default (c:/cygwin)
4. Set the local package directory to whatever you want or use default (c:/cygwin)
5. Select internet connection or use default 'Direct Connection'
6. Choose any server download site, and wait for download to complete
7. Under 'Select Packages', expand 'Devel', then check the boxes next to 'binutils' and 'make'
8. Wait for download to complete and you are DONE

After that you can start the "Cygwin Bash Shell" from the Windows start menu and check for "ld" and "make".

```
$ make -v
GNU Make 3.81
Copyright (C) 2006 Free Software Foundation, Inc.
...

This program built for i686-pc-cygwin

$ ld -v
GNU ld (GNU Binutils) 2.18.50.20080625

===
```

Once the make utility is installed, you are now ready to build GFx.

Below are the steps for using make:

1. Create a Makeconfig file in the top level GFx directory (or edit the example one that is provided there). The file should contain:

```
export CELL_SDK := <path to cell sdk>
EPATH := $(PATH)
export PATH = $(CELL_SDK)/host-win32/ppu/bin:$(CELL_SDK)/host-
               win32/Cg/bin:$(CELL_SDK)/host-win32/bin:$(EPATH)
```

The paths in Makeconfig must be unix-format paths, with forward slashes and no drive letters with colons. Use */<drive letter>/...* for msys and */cygdrive/<drive letter>/...* for Cygwin. For example, if CELL_SDK is installed to C:\ps3\cell:

```
"export CELL_SDK := /cygdrive/c/ps3/cell".
```

2. Build GFx libs and samples.

```
$ make P=PS3
```

To build using SNC:

```
$ make P=PS3+snc
```

When using a Lib or Eval release, the "normal" PS3 release cannot build with snc, and the snc release cannot build without snc (make options must include P=PS3+snc).

3. Copy a SWF file to *\$(CELL_SDK)/FxPlayer/flash.swf*. Other SWF and GFX files in that directory can be viewed in FxPlayer; use the L1 and R1 buttons.
4. If this is an evaluation version, place the license key in *"\$(CELL_SDK)/FxPlayer/gfxlicense.txt"* (or wherever your cell folder is). Your application must load the RTC PRX module:

```
cellSysmoduleLoadModule(CELL_SYSMODULE_RTC);
```

5. Now run the demos with the ProDG target manager or debugger. Configuring the home directory is not needed as the absolute path *\$(CELL_SDK)/FxPlayer* is used for loading Flash files.
6. To run the SWFToTexture and RenderTexture demos, copy the SWF files in *Bin/SWFToTexture* and *Bin/RenderTexture* to *\$(CELL_SDK)/FxPlayer*.

2.3.2.4 Building GfX using Visual Studio

As an alternative to using Make files, Visual Studio solutions and projects are provided to build the GfX demos (and libraries if you have the source installation for Win32 and PS3). The only disadvantage to using Visual Studio is that shaders will not be recompiled, but if you just want to rebuild the code and are not changing shaders, Visual Studio will work fine.

To build the GfX source, you need to run the source installers for PS3 and Win32. Open up the Visual Studio solution called GfX 3.3 PS3 SDK.sln which is located in the *Projects/PS3/Msvc80/SDK* folder. Next, select the appropriate configuration and choose Build Solution from the Build menu.

To build the demos, you need to follow a similar procedure. Open up the solution called GfX 3.3 PS3 Demos.sln which is located in the *Projects/PS3/Msvc80/Demos* folder. Next, select the appropriate configuration and choose Build Solution from the Build menu.

2.3.2.5 PS3 Renderer Support

Currently, support for both GCM and GL rendering is supported in the PS3 version.

libgcm:

Source: *Src/GRenderer/GRendererPS3Impl.cpp*

This renderer uses shaders linked into the executable; no runtime file loading.

Link *Lib/PS3/Shaders.a* with your executable; to use our build system just for this:

```
$ make P=PS3 Lib/PS3/Shaders.a
```

PSGL:

Source: *Src/GRenderer/GRendererOGLImpl.cpp*

Also see *Src/GRenderer/GRendererOGLImplPS3.cpp*

This renderer uses shaders linked into the executable; no runtime file loading.

Link *Lib/PS3/GLShaders.a* with your executable; to use our build system just for this:

```
$ make P=PS3 Lib/PS3/GLShaders.a
```

GLShaders.a does not currently support the SN linker. Use "`ppu-lv2-g++ -mno-sn-ld`" to use the GNU linker.

The following shows the various GfX players that utilize libgcm and PSGl renderers.

With libgcm renderer:

- | | |
|---------------------------|---------------------------------------|
| • FxPlayerGCM | Basic SWF player |
| • FxPlayerTiny | Tiny SWF player |
| • FxPlayerSWFToTextureGCM | Demo of rendering SWF to texture |
| • FxPlayerTextureInSWFGCM | Demo of embedding user texture in SWF |

With PSGL renderer:

- | | |
|--------------------------|---------------------------------------|
| • FxPlayerGL | Basic SWF player |
| • FxPlayerTinyGL | Tiny SWF player |
| • FxPlayerSWFToTextureGL | Demo of rendering SWF to texture |
| • FxPlayerTextureInSWFGL | Demo of embedding user texture in SWF |

All samples support PSGL and libgcm renderers using a common sample framework for video mode setup, buffer allocation, etc.

PSGL uses *Apps/Demos/Common/OpenGLPS3App.**

libgcm uses *Apps/Demos/Common/GcmGLPS3App.**

FxPlayerSWFToTexture/FxPlayerTextureInSWF also require *Apps/Demos/Common/MathLib.**

Code in the samples themselves that is renderer specific uses the following conditional compilation:

```
#ifdef FXPLAYER_RENDER_GCM
    <gcm specific>
#else
    <psgl specific>
#endif
```

2.3.2.6 Build Configurations

Libs and executables have the configuration names appended (except for Release alone).

- | | |
|-------------|---|
| • Release | Optimized |
| • Debug | With debug info |
| • DebugOpt | Optimized with debug info |
| • NoRTTI | Disable RTTI & exceptions |
| • NoThreads | Disable GFx thread support (GFC_NO_THREADSUPPORT) |

Specify the configuration to build with C=<config> on the make command line. Separate multiple configuration names with "+". For example, optimized with no thread support is "make

C=Release+NoThreads". The flags used for each configuration are near the top of the root Make file, or at the top of a platform-specific make file in a subdirectory of Projects.

2.3.2.7 Integration Notes

GFx requires the following PRX Modules: resc, fs, and gcm:

```
cellSysmoduleLoadModule(CELL_SYSMODULE_RESC);  
cellSysmoduleLoadModule(CELL_SYSMODULE_FS);  
cellSysmoduleLoadModule(CELL_SYSMODULE_GCM);
```

ActionScript "Date" (and evaluation versions) require rtc also:

```
cellSysmoduleLoadModule(CELL_SYSMODULE_RTC);
```

2.3.2.8 Thread Support

Thread support is enabled by default if the system supports it. To build without thread support, define `GFC_NO_THREADSSUPPORT` when compiling all files. `GThreads_pthread.cpp` does not have to be built.

2.3.3 GFx SDK for Wii

The GFx SDK for Wii comes with specific read me text files – *readme_wii.txt*, *readme_make.txt* – that provide compiler instructions. It is recommended that users read these text files apart from this documentation while performing the installation.

2.3.3.1 Requirements

Please make sure that you have this software installed and functioning on your development computer:

- Windows XP 32-bit (other versions of Windows may work for building only)
- Cygwin, including packages:
 - binutils
 - make 3.81
- Revolution SDK 2.4 or later

2.3.3.2 Installation

Install GfX SDK for Windows, followed by GfX SDK for Wii. All .exe installers place the files in the same location, selected during the first install. The installer sets the environment variable GFXSDK to point to the top level SDK directory. You may install GfX packages distributed in .tar.gz or .zip format in this location also, as long as the GfX version is the same in all cases.

2.3.3.3 Running the Player/Samples

The executables in Bin/wii can be run with Codewarrior or ndrun. You can drag an executable to Codewarrior and a project for debugging will be created automatically. Press (F5) or click the green arrow to run (make sure to copy the appropriate content first).

The sample applications access data through the DVD emulation. Place the Flash files and any resources they require in *RVL_SDK/dvddata/FxPlayer*. If this is an evaluation version, place the license key in *dvddata/FxPlayer/gfxlicense.txt*.

The sample applications load these files:

| | |
|----------------------|---|
| FxPlayer | flash.swf (use any SWF or GfX file) |
| FxPlayerTiny | Window.swf (use any SWF or GFX file) |
| FxPlayerSWFToTexture | Copy all files in <i>Bin/SWFToTexture</i> to <i>dvddata/FxPlayer</i> . |
| FxPlayerTextureInSWF | Copy all files in <i>Bin/RenderTexture</i> to <i>dvddata/FxPlayer</i> . |

FxPlayer controls:

| | |
|---------|---|
| A | Enter key and mouse click |
| B | Escape key |
| 1 | Open/close HUD |
| 2 | Toggle HUD focus |
| Home | Pause |
| - and + | Change Flash file (other SWF and GFX files in <i>dvddata/FxPlayer</i>) |
| C | Toggle fast forward |
| Z | Change anti-aliasing mode |

2.3.3.4 Building the Samples

Both make files and Codewarrior projects can be used for building the sample applications. Some prebuilt executables are also included.

GFx uses several build configurations, which are a combination of features and compiler options. Each build configuration is one of:

| | |
|----------|--------------------------------------|
| Release | Optimized without checks |
| Debug | With debug info and checks |
| DebugOpt | Optimized with debug info and checks |

Plus some optional suffixes:

| | |
|------------|---|
| +NoRTTI | Disable RTTI & exceptions |
| +Sdata0 | No small data section (-sdata 0 -sdata2 0) |
| +NoThreads | Disable GFx thread support (GFC_NO_THREADSUPPORT; never shipped prebuilt) |

For evaluation and lib packages, only the configurations that were supplied prebuilt will build correctly. Wii packages only include configurations with +NoRTTI. You may build with RTTI from a source package. The + becomes _ (underscore) in output filenames.

To build with Codewarrior, first enable XML projects in your IDE by checking "Use text-based projects" in the IDE Extras group of the Preferences window. Note that this will cause other projects you may be using to be converted. If you do not want this, convert our projects and then disable the option. The executables built with Codewarrior have shorter filenames than those built with make or included in the packages because the linker cannot use long filenames.

The projects do not use the video add-on if installed. You can add video support to those projects by adding libgfx_video.a to the appropriate targets (be sure the configurations match), and adding "#define GFC_USE_VIDEO" to the preprocessor text in Target Settings, C/C++ Preprocessor group.

Projects for rebuilding GFx libs or the 3rdParty libs are not included, to rebuild those you must use make.

To build with make, you need to have a linux-style toolset installed – we recommend Cygwin. Please see the PS3 section 2.3.2.3 on 'Building GFx using make' that specifies how to download and install Cygwin.

Next, create a Makeconfig file (or edit the one that is provided) in the top level SDK directory containing the paths to parts of the Wii SDK (these paths may be different on your system).

```
WII_CWDIR := "/cygdrive/c/Program Files/Freescale/CW for Wii v1.0"
WII_SDKDIR := /cygdrive/c/RVL_SDK
WII_NDEVDIR := /cygdrive/c/NDEV
```

The paths in Makeconfig must be unix-format paths, with forward slashes and no drive letters. Use */cygdrive/<drive letter>/...* at the start of a path. Use quotes if the path contains spaces or other special characters.

Then run `"make P=wii C=DebugOpt+NoRTTI"` from a cygwin shell. This will build the DebugOpt+NoRTTI configuration, which provides reasonable performance and some capability in the debugger. The value of C in the command line can be replaced by any supported build configuration. The environment settings come from the Makeconfig file and so it is not necessary to use a RVL_NDEV shell or other specific environment for building.

2.4 Installing the Scaleform Launcher Panel for CS3

The Scaleform Launcher panel provides a quick and accessible method for publishing a SWF file directly to the Scaleform GFx player from the Flash authoring environment. In order to install them follow the steps outlined below.

1. Launch the Adobe Extension Manager, which can be found on Windows in Windows®>Start Menu>Programs>Adobe Extension Manager CS3. The default installation location is *C:/Program Files/Adobe/Adobe Extension Manager*.

On Mac, the Adobe Extension Manager can be launched from Finder>Applications>Adobe Extension Manager CS3>Extension Manager.

2. In the Extension Manager window, click the *Install New Extension* button, located in the top left of the window. And on Mac, click the *Install* button, located in the top left of the window.
3. Browse to the following directory:
On Windows: *C:/Program Files/Scaleform/GFx SDK 3.3/Resources/Tools/*
On Mac: *scaleform_gfx_3.3_macos/Resources/Tools/*
4. Double click the MXP file in this directory: *Scaleform Extensions.mxp*.

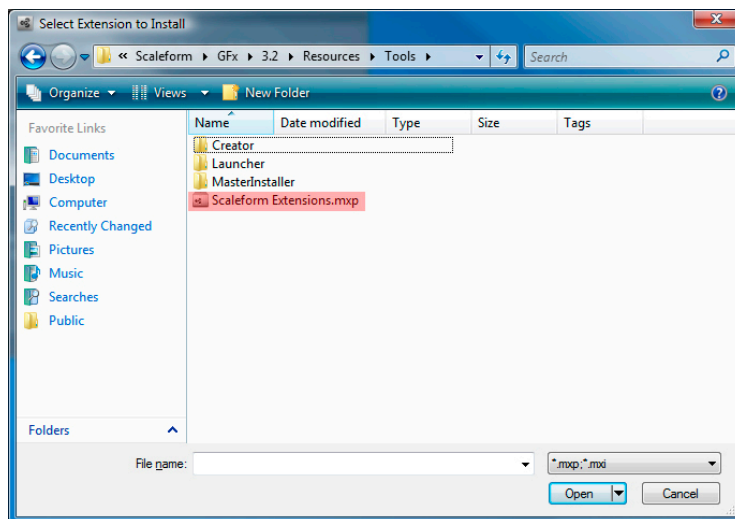


Figure 1: Scaleform Extensions.mxp file location on Windows.

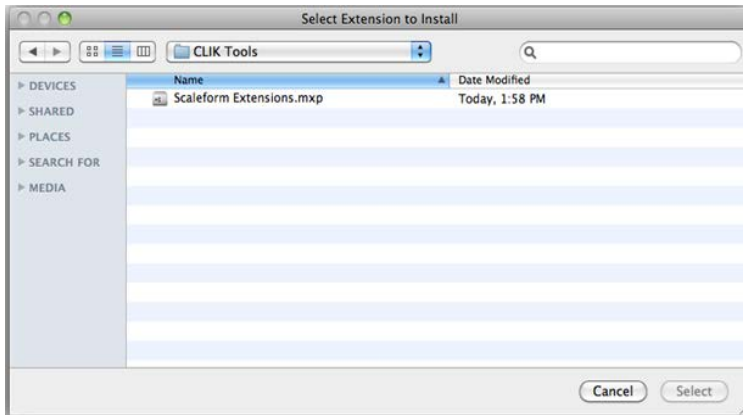


Figure 2: Scaleform Extensions.mxp file location on Mac.

5. Press *Accept* on the license window.
6. Press *OK* in the dialogue telling you the extension was successfully installed.

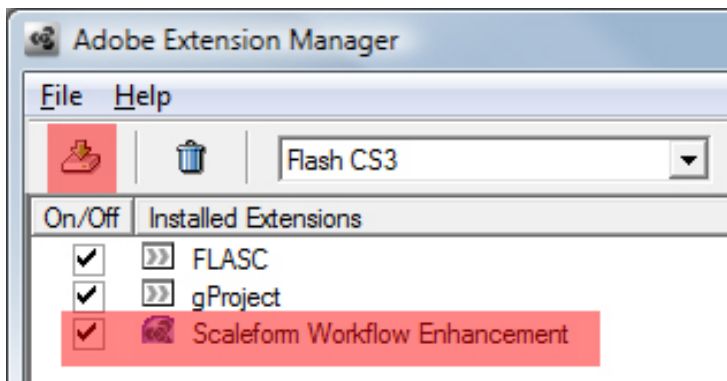


Figure 3: Adobe Extension Manager in CS3 on Windows.

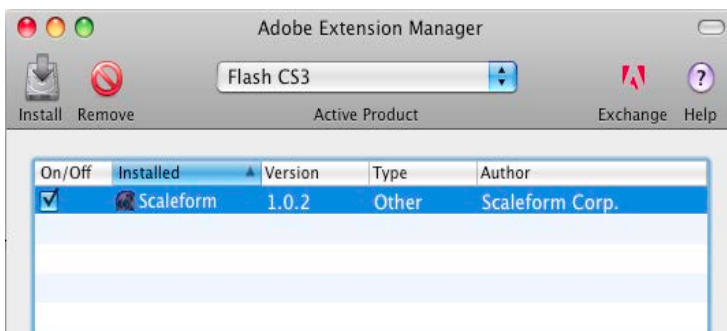


Figure 4: Adobe Extension Manager in CS3 on Mac.

7. Exit the Adobe Extension Manager.
8. Restart Flash and create a new Flash File (ActionScript 2.0).

9. In Flash, the *Scaleform Launcher* panel is now accessible via the *Window* dropdown menu, under *Other Panels*. Open the panel at this time by selecting it.

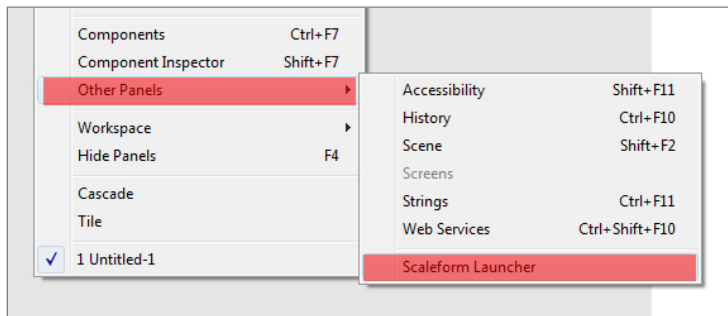


Figure 5: Add the Scaleform Launcher via Other Panels on Windows.

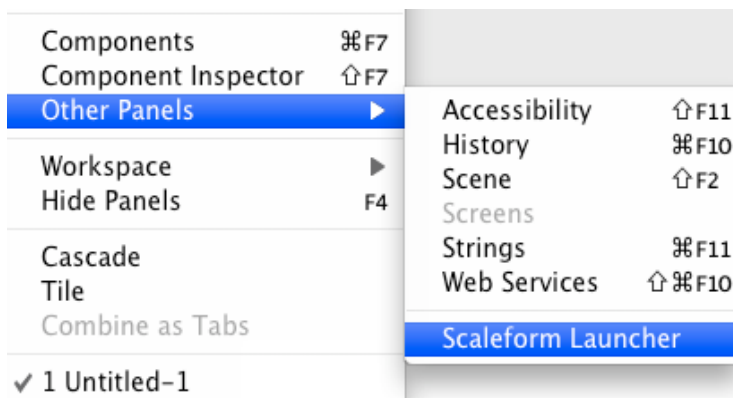


Figure 6: Add the Scaleform Launcher via Other Panels on Mac.

2.5 Installing the Scaleform Launcher Panel for CS4

The Scaleform Launcher panel provides a quick and accessible method for publishing SWF files directly to the Scaleform GfX player from the Flash authoring environment. In order to install them follow the steps outlined below.

1. Launch the Adobe Extension Manager from CS4 by clicking *Help* in the top Flash menu, and then select *Manage Extensions* from the dropdown. **IMPORTANT:** If the Extension Manager is not launched from within Flash, there is a known Adobe issue in which the manager may install the extension into the wrong directory if the computer has more than one language installed.
2. In the Extension Manager window click the *Install* button, located in the top-middle of the window.
3. Browse to the following directory:
On Windows: *C:/Program Files/Scaleform/GfX SDK 3.3/Resources/Tools/*
On Mac: *scaleform_gfx_3.3_macos/Resources/Tools/*
4. Double click the MXP file in this directory: *Scaleform Extensions.mxp*.

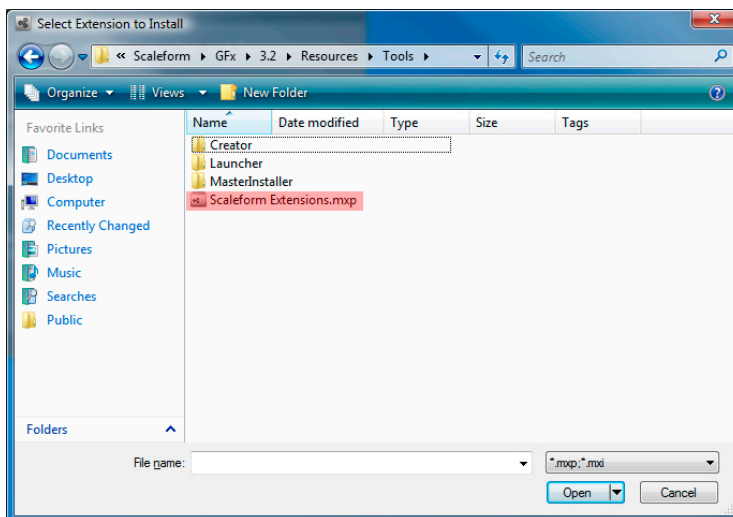


Figure 7: Scaleform Extensions.mxp file location on Windows.

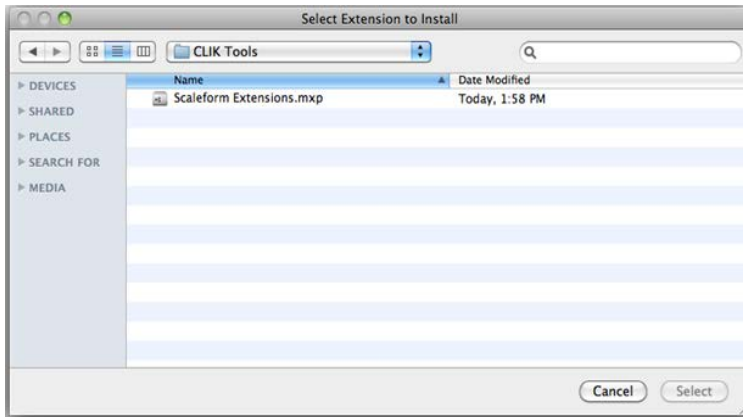


Figure 8: Scaleform Extensions.mxp file location on Mac.

5. Press *Accept* on the license window.
6. Press *OK* if a dialogue pops up informing you to close and restart Flash CS4.

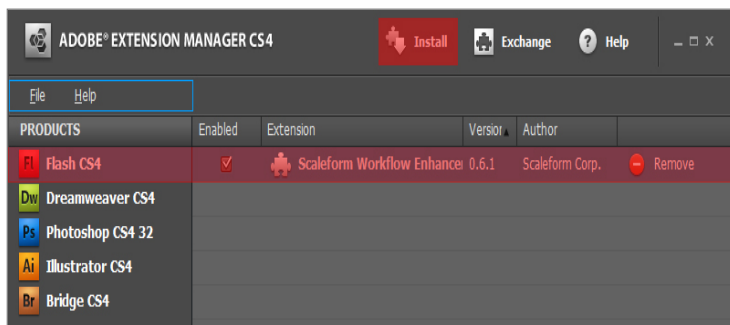


Figure 9: Adobe Extension Manager in CS4.

7. Exit the Adobe Extension Manager.
8. Restart Flash.
9. In Flash, the *Scaleform Launcher* panel is now accessible via the *Window* dropdown menu, under *Other Panels*. Open the panel at this time by selecting it.

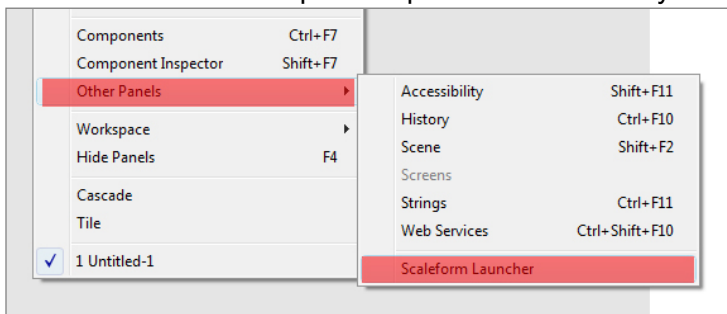


Figure 10: Displaying the Scaleform Launcher panel on Windows.

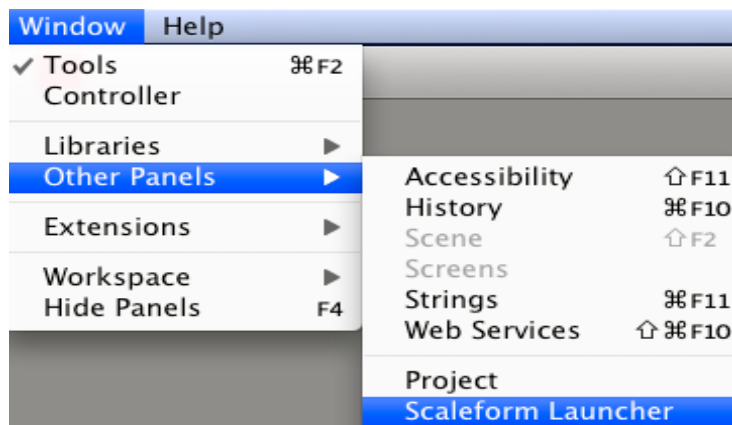


Figure 11: Displaying the Scaleform Launcher panel on Mac.

2.6 Using the Scaleform Launcher Panel

The *Scaleform Launcher* panel allows the creation of Scaleform GfX player profiles, which can be used to publish and test SWF files directly to the Scaleform GfX player. This is the preferred method for publishing and testing SWFs designed to be used by Scaleform GfX, as the standard Adobe player does not support CLIK and other GfX specific extensions.

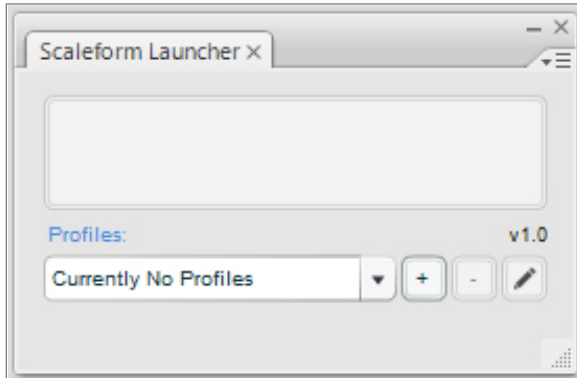


Figure 12: The Scaleform Launcher Panel

Adding a new profile opens the *Add New Profile* dialogue — see *Adding Players/Profiles* below for instructions on how to add a new profile. Inside the *Add New Profile* dialogue, the *player EXE* combo box maintains a list of all the player executables, which can be managed by pressing the *Plus* [+] and *Minus* [-] buttons. Scaleform GfX for Windows comes with executables for AMP (Scaleform's Analyzer for Memory and Performance), Debug and Release, 64 and 32-bit builds of the FxPlayer. Scaleform GfX for Mac comes with executables for FxPlayer_DebugOpt (AMP), Debug and Release, Intel and PowerPC builds of the FxPlayer. Custom launcher executables on both Windows and Mac system can also be made to publish and preview SWF content on consoles or directly inside of a game.

The *player EXE* dropdown allows the selection of the executable that will be used in the current profile. The same EXE can be reused in several profiles, allowing them to receive different command line parameters.

Adding Players/Profiles

1. When the panel is first installed, it is necessary to add a GfX player for testing SWF movies. Press the *Plus* [+] button located to the right of the *Profiles:* dropdown to bring up the *Add New Profile* dialogue.

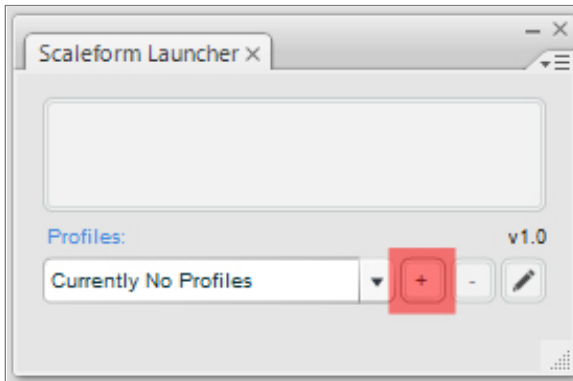


Figure 13: Add a new profile by pressing the plus button

NOTE: If a window indicating the script is causing Flash to run slowly pops up, select *No* to continue normal operation. This is a known issue that will be corrected in future releases.

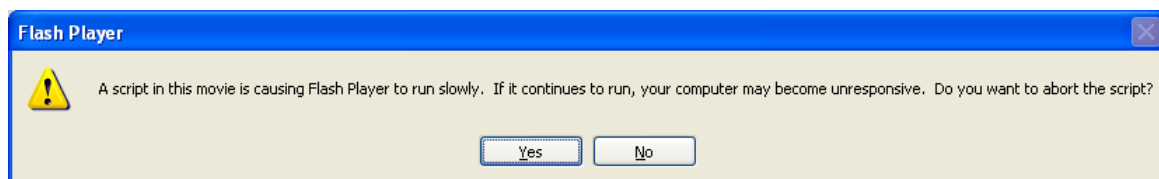


Figure 14: Abort script dialogue.

2. Inside the dialogue, click the *Plus* [+] button to browse to and select the desired Scaleform GfX player executable.

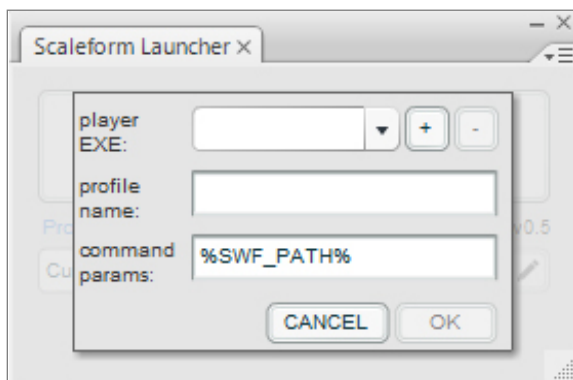


Figure 15: Add new profile dialogue.

3. We recommend choosing the *FxMediaPlayer* executable on Windows platform because it provides performance monitoring statistics in the player window; however, any Gfx player executable may be used. The *FxMediaPlayer* file is located in *C:/Program Files/Scaleform/GFx SDK 3.3/Bin*

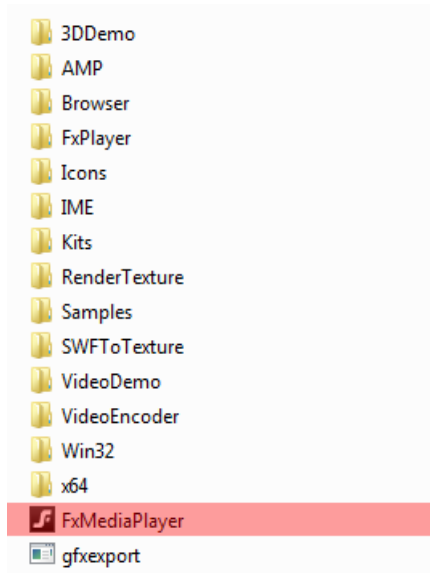


Figure 16: Browse to the *FxMediaPlayer* executable on Windows.

On Mac, choose the *FxPlayer_DebugOpt* executable located in *scaleform_gfx_3.3_macos/Bin/MacOS/FxPlayer_DebugOpt.app/Contents/MacOS*

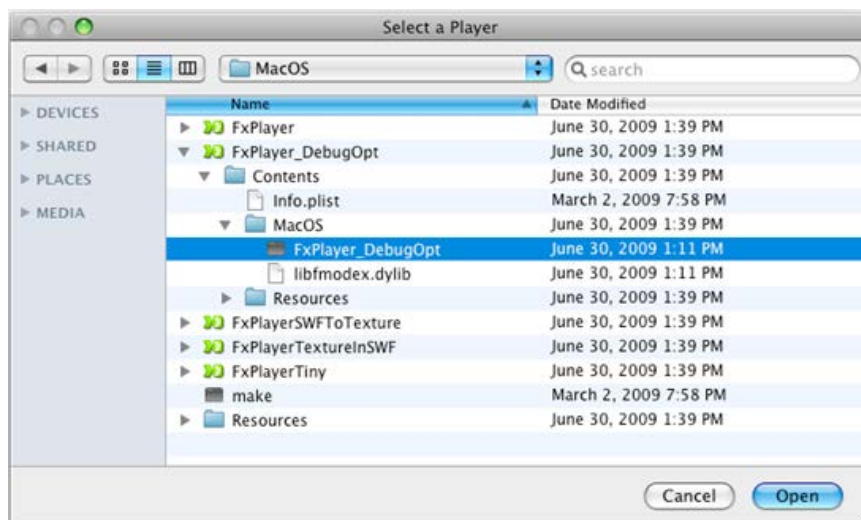


Figure 17: Browse to the *FxPlayer_DebugOpt* executable on Mac.

4. A prompt will pop up. Enter a name for the player in the prompt under *Player Name*.

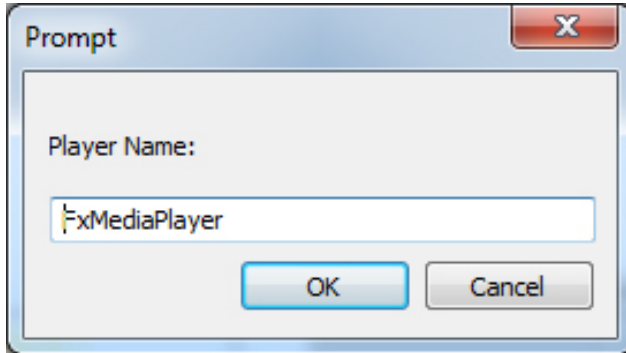


Figure 18: Player name prompt on Windows.



Figure 19: Player name prompt on Mac.

5. Press *OK* to close this dialogue.
6. Return to the *Add Profile* dialogue, and type a new profile name in the *profile name* field. For instance, type 'AMP' on Windows system and 'DebugOpt' on Mac.
7. The *command params* field is used to add custom launch parameters that Scaleform GfX player recognizes. Be sure to leave '%SWF_PATH%' as the first entry in this field. All command line parameters should follow this initial entry:

Example: '%SWF_PATH% -nh'

This entry will load the SWF into Scaleform GfX player with no HUD.

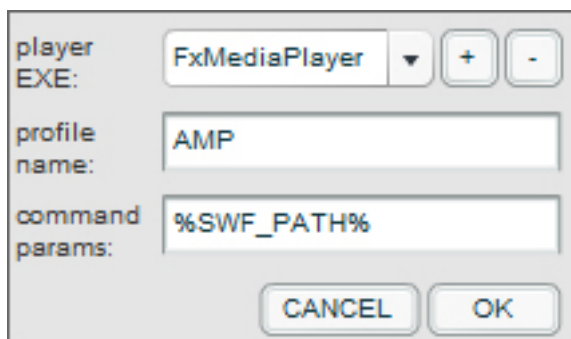


Figure 20: AMP Profile with default command params on Windows.



Figure 21: FxPlayer DebugOpt with default command params on Mac

8. Press *OK*.
9. The *new* profile will now be available in the *Profiles* dropdown inside the *Scaleform Launcher* panel. Other user-defined profiles will be located in this dropdown as well.

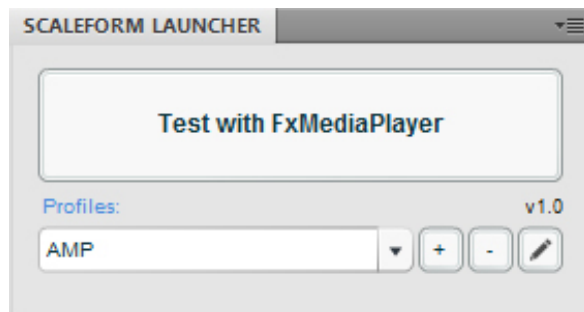


Figure 22: Scaleform Launcher panel with AMP profile visible on Windows.

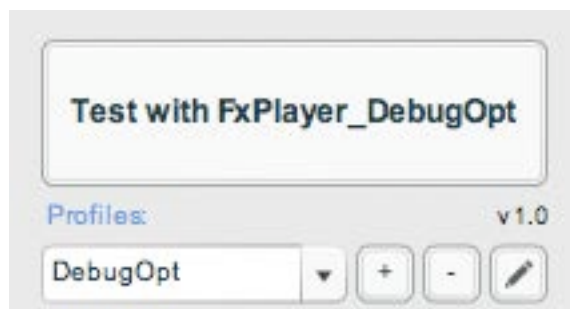


Figure 23: Scaleform Launcher panel with Debug_Opt profile visible on Mac.

Editing Profiles

1. Select a profile from the *Profiles* dropdown in the *Scaleform Launcher* panel.
2. Press the *Edit* button, which is denoted by a pencil icon.

3. Choose a different player from the *player EXE* dropdown, change the profile name, or alter the command parameters as necessary.
4. Press *OK*.

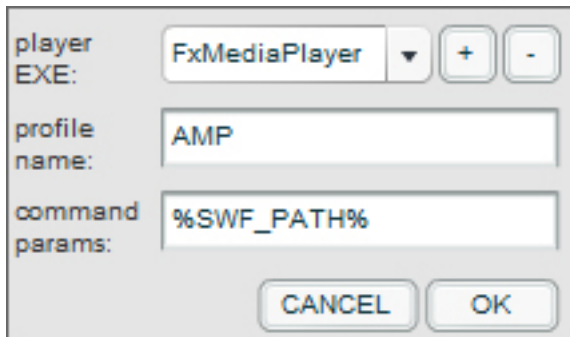


Figure 24: Edit and Delete Profiles dialogue on Windows.

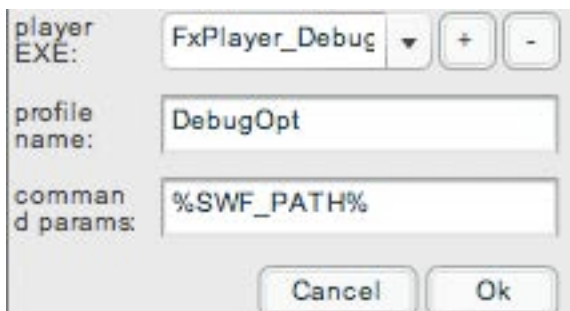


Figure 25: Edit and Delete Profiles dialogue on Mac.

Deleting Profiles

1. Select a profile from the dropdown in the *Scaleform Launcher* panel.
2. Press the *Minus [-]* button to delete it.
3. When asked if you are sure you want to remove the selected profile, press *OK* to do so, or press *Cancel* to keep the profile.

Testing SWFs

1. In order to test a movie, create and save a sample SWF file.
2. Choose a profile from the *Profiles* dropdown menu in the *Scaleform Launcher* panel.

3. Press the *Test with [Executable Name]* button.
4. Scaleform GfX player will be launched, and the SWF will be published and played.

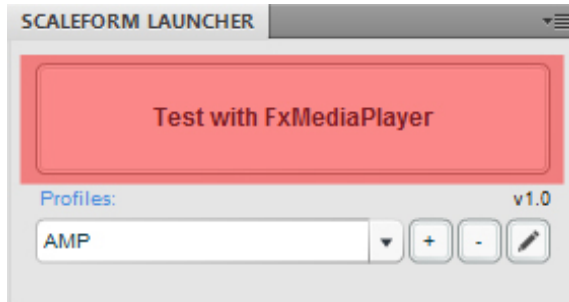


Figure 16: Press Test with FxMediaPlayer to publish and test the SWF on Windows.

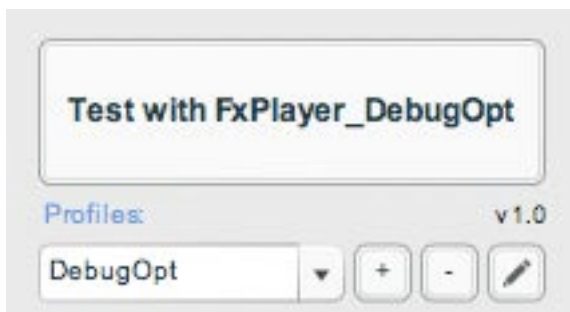


Figure 27: Press Test with FxPlayer_DebugOpt to publish and test the SWF on Mac.

Launching the Player via Keyboard Shortcut

You can setup Flash to launch the GfX Player via keyboard shortcut as well. Simply follow the steps below to set it up.

1. Select *Edit* in the top menu.
2. Choose *Keyboard Shortcuts* in the dropdown menu.
3. Ensure Adobe Standard is the Current set: in the top dropdown of the Keyboard Shortcuts window.
4. Press the *Duplicate Set* button.

5. (Optional) Rename the set.
6. Press *OK*.
7. Ensure "Drawing Menu Commands" is selected in the *Commands* dropdown.
8. Press the plus sign (+) next to *Commands* in the list to expand it.
9. Select Publish in GfX Player.
10. Add a new Shortcut using the plus (+) button.
11. Set the shortcut to Ctrl+Enter by typing (Ctrl+Enter) while the Press key field is highlighted. This will override the default Flash behavior. Alternatively, you may set this to whatever key combination you prefer.
12. Press the *Change* button to accept.
13. When asked to reassign, click the *Reassign* button.
14. Next, open the Control list under *Commands*.
15. Select Test Movie. This is the normal Flash publish shortcut.
16. Add a new Shortcut using the plus (+) button.
17. Type (Ctrl+\ (backslash)) while the Press key field is highlighted to set the shortcut. Alternatively, you may set this to whatever key combination you prefer.
18. Press the *Change* button to accept.
19. Press *OK* to accept all changes and close the Keyboard Shortcuts window.

2.7 Licensing

2.7.1 License Problems

To run any application that is linked with GfX 3.3 Eval libs, please ensure the file *gfxlicense.txt* is in one of the proper directories for the desired platform. The file *gfxlicense.txt* is a text file that contains the key in the following format—all lower case with no spaces:

```
aaaaa-bbbbbb-cccc--ddddd
```

If an early version of the game or application is released, the *gfxlicense.txt* file may be distributed in the released package. Please note that GfX 3.3 searches the directories in the order they are listed below. Special consideration must be taken for an SDK installation with multiple platforms. If a valid *gfxlicense.txt* is found in directory path two, but an expired *gfxlicense.txt* is found before that in directory path one, the application will exit with the assumption that the evaluation license is expired.

When an evaluation expires, any GfX 3.3 application that finds the expired *gfxlicense.txt* file will exit, displaying the message:

```
The GfX trial period expired on 1(m) 1(d) 2008, please contact Scaleform for an extension.  
Error: 106
```

If the application cannot find a *gfxlicense.txt* in any of the locations listed below, the application will exit, displaying the message:

```
Failed to open the GfX license file, please make sure gfxlicense.txt exists in the working  
directory. Error: 101
```

The directories checked and the order they are checked for each platform are:

Windows / Mac® / Linux®

1. Current directory.
2. Directory specified by the GFXSDK environment variable.
3. If no GFXSDK environment variable is set, then the directory specified by the HOME environment variable.

Xbox 360®

1. XDK Debug Drive - D:/\$ (SolutionName)/gfxlicense.txt

Note that the license key file should be put in the same directory where an executable file is deployed to.

“PLAYSTATION®2”(PS2™)

1. On the host machine:

Linux - /usr/local/sce/FxPlayer/gfxlicense.txt

Windows - C:/usr/local/sce/FxPlayer/gfxlicense.txt

“PSP®”(PlayStation®Portable)”

1. On the host machine:

Linux - /usr/local/psp/FxPlayer/gfxlicense.txt

Windows - C:/usr/local/psp/FxPlayer/gfxlicense.txt

“PLAYSTATION®3”(PS3™)

1. /app_home/gfxlicense.txt
2. /host_root/usr/local/cell/FxPlayer/gfxlicense.txt
3. /dev_bdvd/PS3_GAME/USRDIR/gfxlicense.txt

Wii™

1. dvddata/FxPlayer/gfxlicense.txt

2.8 Support

There are a wide variety of support options available to you as a Scaleform customer.

Website: Each customer has a registered name and project on the Scaleform website at www.scaleform.com. If this is not the case, please register. Once registered, access to <https://developer.scaleform.com/gfx> will become available. This is the developer section of the Scaleform website, where there is a variety of documents, whitepapers, tutorials, sample files, downloads, forums and an FAQ.

Email: Email Scaleform support at support@scaleform.com with any problems or questions that might arise. Please search the documentation and website support forums for more info on topics before contacting support, however, as there have been many issues that have already been solved and posted in the developer section of the website. When emailing support, please include: name, company name, detailed description of the problem, steps to reproduce, platform, screenshots of the problem if available. Please be sure to attach any relative code or files Scaleform will need to view in order to help diagnose the issue.

Forum: We encourage all Scaleform licensees to use the developer forums to exchange information with other developers whenever possible. Because every project has different needs, it is possible that another studio with a similar project has already tackled the issue. A fellow programmer, designer or artist may be able to provide advice tailored to the issue even better than one of Scaleform's technical support team members. The Scaleform engineers also regularly post important information in the forums, so keep an eye out for these critical posts.

If your company currently has a support contract with us, we also provide additional email and phone support. Please see your master service agreement for full details on technical support options.

2.9 Feedback

At Scaleform we want to hear anything you have to say about us or our products. We have set up a special email address at feedback@scaleform.com. This email address goes directly to our senior production team. We would love to hear ideas for new features, suggestions for changes to our core architecture and anything else that you think we should hear about. Your feedback and suggestions are essential to help us continue making Scaleform GfX a great product.

3 What is Included?

Scaleform GfX 3.3 comes with a variety of different documents and sample files. These documents provide information related to the core features of GfX 3.3, integrating GfX 3.3 into a game engine, how to use GfX 3.3 and its various components, how GfX 3.3 works with Flash and AS 2.0 and much more. These documents serve as a guide on how to get started, but by no means cover every possible option, as there are many different solutions to most problems, especially when considering different game engines, game genres, use cases, platforms, etc.

Assuming GfX 3.3 was installed to the default directory, most documents will be found under the *Doc* folder.

Documents are also available in two other locations:

1. Via the SDK Browser.
2. Via the Windows Start menu: Start->Programs->Scaleform->GfX SDK 3.3->Documentation

3.1 Learning Flash

Some documents assume a certain level of knowledge of Flash and AS2. For Scaleform GfX users new to Flash, we recommend visiting the IGDA (International Game Developers Association) Flash SIG (Special Interest Group) website at http://www.igda.org/wiki/Flash_SIG. This website has a list of various Flash resources from beginner level and up; we also encourage interested individuals to consider joining the IGDA and the SIG, if you have not already done so.

Some other excellent Flash and ActionScript resources include:

- [Lynda.com](http://www.lynda.com)
- [Essential ActionScript 2.0 by Collin Moock](#)
- [ActionScript: The Definitive Guide by Collin Moock](#)

3.2 Getting Started Guides

Scaleform GfX 3.3 comes with a number of getting started guides. Designed for users of all skill levels, these guides include short, easy-to-follow examples to help get users familiarized with GfX 3.3 as quickly as possible.

- *Getting Started with CLIK*: Provides an overview of CLIK and walks users through building their first front-end interface with Flash, CLIK, Illustrator® and Photoshop®.
- *Getting Started with CLIK Buttons*: Provides an overview of how the button components are built and used in CLIK. New users should first complete the *Getting Started with CLIK* document before starting this one.
- *Getting Started with Video*: Walks a user through video creation in After Effects® or Adobe Premiere®, adding alpha channels, exporting the video to USM format for use in GfX 3.3, and how to set up a Flash file to play the final video. Note: This document is only included with Scaleform Video.
- *Getting Started with IME*: Provides a quick overview of the Input Method Editor and how to quickly integrate it and get it up and running in a game engine.

3.3 Development Documents

These documents detail more technical information relating to GfX 3.3 and Flash. They are much more in depth than the *Getting Started* guides and are written to aid in GfX 3.3 integration with an application or game engine.

- *Integration Tutorial*: This document introduces basic GfX 3.3 usage and 3D engine integration using a DirectX 9 example application.
- *Reference Documentation*: Available online and as CHM, the Reference Documentation describes the GfX 3.3 library including the general framework and detailed function information.
- *Scaleform CLIK User Guide*: This document describes the CLIK framework and the components with detailed implementation instructions.
- *Memory System Overview*: This document covers new features in GfX 3.3 including: AMP Performance Analysis, ActionScript Garbage Collection and Technical Performance Metrics information.

- *Font Overview Configuration*: This document describes the font and text rendering system used in Scaleform GfX 3.3, providing details on how to configure both the art assets and GfX C++ APIs for internationalization.
- *Scale9Grid*: This document describes the details of using Scale9Grid functionality in Scaleform GfX 3.3 to create undistorted, resizable windows, panels, buttons, and other images.
- *ActionScript Extensions*: This document describes ActionScript extensions available in Scaleform GfX 3.3.
- *IME Configuration*: This document covers how to integrate and use IME with GfX 3.3. It also includes information on the configuration and use of the IME Language Bar.
- *DrawText API*: This document describes the GfXDrawText API introduced in GfX 3.3 for C++ text driven rendering and formatting outside of ActionScript sandbox.
- *XML Overview*: This document describes configuring XML support available in Scaleform GfX.
- *Flash Support Overview*: This document provides the most up-to-date listing of Flash functionality supported by GfX 3.3.
- *Best Practices Guide*: This *document* describes the best practices for efficiently implementing and effectively leveraging GfX 3.3.
- *GfXExport Reference Guide*: This document describes the GfXExport utility tool which preprocesses SWF files into an optimized GfX format for use with Scaleform GfX.
- *GfX LITE Customization Overview*: This document details the Scaleform GfX LITE configuration libraries and describes how to customize GfX LITE in your own application.
- *Frequently Asked Questions*: This document covers many commonly asked questions on a variety of topics.

3.4 Platform-Specific Documents

Registered developers for Microsoft, Sony, Nintendo, or UE3 may have access to additional documentation if platform-specific code has been licensed and approved. If platform-specific SDKs are not accessible, but you are a registered developer for these platforms, please contact support@scaleform.com.

- *IME Xbox 360 Configuration*: Covers IME integration for Xbox 360.
- *IME PS3 Configuration*: Covers IME integration for PS3.
- *Unreal Engine 3 Workflow*: This document explains the basics of importing interactive Flash content into Unreal Engine 3.
- *Unreal Engine 3 Integration*: This document covers how to fully integrate GFx 3.3 into UE3. It describes how Scaleform GFx 3.3 can be integrated with UE3, including Kismet, UnrealScript and the Unreal Editor.
- *Gamebryo LightSpeed Integration Overview*: This document presents the Gamebryo LightSpeed integration with GFx 3.0 and higher versions.

3.5 Demos and Examples

Scaleform GFx 3.3 also comes with several demos and examples. More demos and examples are available on the Scaleform website, and can be downloaded and installed separately. They are found in the “Downloads” Section, after clicking the *Demos* button. Alternatively, point the web browser to <https://developer.scaleform.com/gfx?action=demo>.

The demos include videos, interactive content, Flash files and a wide assortment of other content designed to aid licensees in learning GFx 3.3 quickly.

- *GFx Player Tiny*: Code which demonstrates how to create a small GFx player, on various platforms. This is the best example to start with when learning about how to start using the GFx SDK.

Source (Multi-Platform): *C:/Program Files (x86)/Scaleform/GFx SDK 3.3/Apps/Demos/GFxPlayerTiny*

- *GFx Player SWF to Texture*: Demonstrates how a Flash-based form can be placed upon a 3D surface. Forms are fully interactive, regardless of the manner in which the 3D object is being manipulated. Illustrates potential in-game applications (e.g., placing a form on an in-game element).

Executable (PC): *C:/Program Files/Scaleform/GFx SDK 3.3/Bin/Win32/Msvc90/GFxPlayerSWFToTexture/FxPlayerSWFToTexture_D3D9_Release_Static.exe*

Source (Multi-Platform): *C:/Program Files (x86)/Scaleform/GFx SDK 3.3/Apps/Demos/RenderTexture*

- *GfX 3D Demo*: Demonstrates selective rendering of Flash to specific textures inside of a 3D model. Flash is being simultaneously rendered underneath, on and above a 3D model. Also illustrates dynamic loading of movies.

Executable (PC): *C:/Program Files/Scaleform/GFx SDK 3.3/Bin/Win32/Msvc90/GfXPlayer3DDemo/FxPlayer3DDemo_D3D9_Release_Static.exe*

Source (PC): *C:/Program Files (x86)/Scaleform/GFx SDK 3.3/Apps/Demos/3DDemo*

- *GfX Player Textures in SWF*: Shows true 3D being rendered inside a movable Flash window. Illustrates Flash effects, such as tweens, on a render target for 3D.

Executable (PC): *C:/Program Files/Scaleform/GFx SDK 3.3/Bin/Win32/Msvc90/GfXPlayerTextureInSWF/FxPlayerTextureInSWF_D3D9_Release_Static.exe*

Source (Multi-Platform): *C:/Program Files (x86)/Scaleform/GFx SDK 3.3/Apps/Demos/RenderTexture*

The above demos can be found in *Bin/<platform>/* directory where the GFx SDK is extracted on Macintosh and Linux platforms.

3.6 Online Documents

In addition to the documents and tutorials that come with the GFx 3.3 SDK, there are additional documents and examples available on the Scaleform website:

<https://developer.scaleform.com/gfx?action=doc>

4 Where to Begin

Where and how to begin with GFx 3.3 depends heavily on how it will be used, what engine and technologies will be used with it, the skill of the team and the time available. It is recommended to first learn how to play and view files in the GFx player, then determine what is needed to integrate GFx 3.3 in the game engine, or learn how to use Flash. Regardless of whether you are an artist, designer or programmer, it is important to acquire a general overview of the entire pipeline for building assets for use with GFx 3.3.

A good early step is to complete a series of performance analysis with your specific implementations in order to verify and ensure optimal functionality. As with other aspects of game development, when

implementing GfX 3.3, common sense and good testing procedures are critical. While integrating GfX 3.3 is not difficult, content creation guides should be established to ensure the UI artist/developer is not creating assets that could degrade performance. In the same way that a developer would not create a character model for an in game engine composed of 10 million polygons, a UI developer would not create a Flash file that has 10,000 animated layers with transparency and excessive animation tweens. Use common sense and past experience when building Flash assets. It is very easy to create assets that run slow if they are too complex or utilize certain features. Remember, just because something works doesn't mean it isn't broken (e.g., poorly implemented). For a FAQ on best practices, please visit the Scaleform website: <https://developer.scaleform.com/gfx?action=faq>

4.1 User-Specific Information

4.1.1 UI Programmers

Programmers must usually first worry about how to integrate GfX into the game engine and how to connect the game systems and databases to the game. In order to get a front end to correctly play in a game, GfX 3.3 must first be integrated into the engine, which will allow the engine to draw 2D Flash graphics. Next, the engineer will need to connect the game databases to the Flash systems (via ActionScript). This will allow the game to send messages to a Flash file and vice versa. This will be necessary to send information from the game client that the interface needs to dynamically know about (e.g., player names, scores, health). The engine will likewise need to communicate changes in the interface back to the game client; for example, if the player changes an option, the game will know about it and be able to take the necessary steps. The more complex the interface or interactive content, the more work which may need to go into making sure the connections are working properly.

4.1.2 UI Designers

User Interface Designers must first make sure they know how to use Flash, and get up to speed on AS2. Designers will want to make sure they know how to use the CLIK components, create interface components from scratch and know how to connect their interactive content to the game or any external sources that may be needed. Early on, designers will probably want to rapidly prototype their projects, which will help them understand how to build functional content while simultaneously analyzing performance.

4.1.3 UI Artists

Artists must make sure they are familiar with CS3 or CS4 and Flash. They will need to make sure they understand how to skin interface components, especially the CLIK components, along with how to create any other graphics that might be needed. Artists and UI Designers should work together to

understand the performance implications of their choices and spend some time experimenting to determine the limits of artwork complexity to avoid negative performance impact. For example, the content for an in-game interface will have significantly different requirements than those for a front-end menu. Understanding how complex everything is will be an important step in the early development process. Once the performance restrictions are understood, then artists can begin to prototype the visual look of the project.

The most important thing to remember when using GfX 3.3 Flash: artists are only limited by their imagination and the performance restrictions placed on them by the game engine and platform.

4.2 Playing Files

Scaleform GfX 3.3 comes with a variety of players for Windows and each platform. The Scaleform GfX 3.3 Player supports playing native SWF files from Flash or Scaleform's proprietary GFX file format. To play a file, simply run the GfX 3.3 player of choice and then drag and drop a SWF or GFX file onto the player window.

By default on Windows, during installation, the Scaleform GfX 3.3 SDK installation places a "Scaleform Fx Media Player" icon on the desktop. Additional players may be accessed from the start menu: Start->Programs->Scaleform ->GfX SDK 3.3>GfX Players. These players may be located via Windows Explorer as well:

C:/Program Files/Scaleform/GfX SDK 3.3/Bin/Win32/Msvc80/GfXPlayer
C:/Program Files/Scaleform/GfX SDK 3.3/Bin/Win32/Msvc90/GfXPlayer

To test, double click the "Scaleform Fx Media Player" icon on the desktop. This will open up a blank player window. Next, use Windows Explorer in Windows to browse to: *C:/Program Files/Scaleform/GfX SDK 3.3/Bin/3DDemo*. There are a number of SWF files in this *3DDemo* folder. To play any of these, just drag and drop them onto the player window.

On other platforms such as Mac or Linux, the players are located in *Bin/<platform>/* in the GfX SDK 3.3 directory. These players can be run from either the command line or the Finder (on Mac) depending on the platform.

To play a sample SWF file, choose any SWF from the *3DDemo* found in *Bin* folder of the GfX SDK 3.3 directory. Drag and drop the sample onto the player window.

Closing the player is not necessary to play a different SWF. Drag and drop another file onto the player at any time to play the new SWF.

To run the player from the command line:

On Linux - `Bin/i386-linux/FxPlayer <swf file>`

On MacOS - `Bin/MacOs/FxPlayer.app/Contents/MacOs/FxPlayer <swf file>`

Note that starting the player from the command line will size the window to the Stage from the Flash file.

Additional samples are provided in: `/Bin/Samples` in the GfX SDK 3.3 installation.

4.3 Starting a New Project

Scaleform GfX 3.3 support the use of AS 2.0. Flash CS3 and CS4 may be used to author content; however, it is important that when starting a new project, the file type chosen is a new AS 2.0 Flash File. This can be done from the initial start page of Flash, or from the top menu via File->New->Flash File (ActionScript 2.0). Ensure that the Flash Publish Settings have Flash Player 8 chosen as the Flash Player type. These steps will help to ensure maximum compatibility.

It is important to note that Scaleform GfX 3.3 support most of the commands in AS 2.0, but not all of them. It is important to reference the *gfx_3.3_flash_support.pdf* file, which contains a full list of all supported AS 2.0 commands.