Autodesk®
Scaleform®

Getting Started with Scaleform 4.3

This document explains how to get up and running quickly with Scaleform 4.3.

Authors: Matthew Doyle, Mustafa Thamer

Version: 2. 12
Last Edited: December 2, 2013

Autodesk®
GAMEWARE

Copyright Notice

Autodesk® Scaleform® 4.3

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 Autodesk Scaleform:

Document	Getting Started with Scaleform 4.3
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 Welcome

Welcome to Autodesk® Scaleform® 4.3. This guide describes what Scaleform 4.33 is, how to use and install the SDK, and where to look for additional information. Scaleform 4.3 and higher versions include several features designed to improve workflow, testing, debugging, and iteration time, as well as an entirely new multi-threaded core architecture and renderer.

Scaleform 4.3 Features and Add-ons:

- Scaleform CLIK - a set of readymade widget components.
- Scaleform Analyzer for Memory and Performance (AMP™) - a remote profiling and debugging tool.
- Support for ActionScript 3 as well as legacy support for ActionScript 2.
- Highly optimized, multi-threaded 2.5D rendering engine.
- Scaleform UI Kits - fully featured demos (code and data provided) that implement common UI use-cases.
- GFxExport - command line data processing tool.
- Extensive documentation and samples.
- Scaleform Video add-on, powered by CRI Movie™.
- Scaleform Input Method Editor (IME) add-on.
- Cross-platform compatibility with all major engines and platforms, including mobile devices.

2 Installation and Usage

The Scaleform 4.3 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://gameware.autodesk.com/scaleform/developer/?action=dl

2.1 Flash Versions

Scaleform 4.3 requires Adobe® Flash® Creative Suite® 4 (CS4) or newer. Scaleform 4.3 supports a broad range of Flash Player 10.1 and ActionScript™ (AS) features. While Scaleform 4.3 does not support every Flash or AS feature, the rare occurrence of an unsupported feature has little to no impact on development. Scaleform 4.3 supports both AS 2.0 and AS 3.0. For a complete list of Scaleform 4.3 supported Flash 10.1 and AS 2.0 and 3.0 features, please see the document entitled *Scaleform 4.3 – 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 CS5.5 may not be needed, and, as such, developers may decide to stay with CS4. 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 4.3 SDK on Windows platform will by default be installed to: *C:\Program Files (x86)\Scaleform\GFx SDK 4.3*

If Scaleform 4.3 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 Scaleform 4.3 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.

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 Scaleform 4.3 Player. For testing of the prebuilt Scaleform 4.3 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 4.3->Documentation folder in the Windows Start menu, or via Windows Explorer: *C:\Program Files (x86)\Scaleform\GFx SDK 4.3\Doc*

2.2.1 Installing Packages

Scaleform packages are provided in three different configurations, depending on the licensing arrangement:

1. Full source
2. Licensed Binary Libs
3. Evaluation Binary Libs

The evaluation packages require the use of a license key, which must be placed in the correct location for the runtime to find it. For more instructions on using the eval license key, please see the subsequent section on Licensing.

Scaleform provides a core SDK, as well as optional add-on packages, which must be purchased separately. Current add-ons include Scaleform IME and Scaleform Video (see the sections below for more details on add-ons).

Install Scaleform packages (and optional add-ons) by using the supplied installers. Installations for all platforms can go in the same directory. Although they share common code which may overlap, platform-specific code, projects, libs and content are organized separately. All installers which are executables set the environment variable GFXSDK to point to the top level SDK directory. You may install Scaleform packages distributed in .tar.gz or .zip format in this location also, as long as the Scaleform version is the same in all cases.

After installing an add-on such as Scaleform Video, it will automatically be enabled in demo applications after they are rebuilt with make. The demo applications in Scaleform SDK include FxPlayer, PlayerSWFToTexture, etc. These prebuilt demo applications included in the base release (without add-ons) 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 the add-ons are not installed.

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**
`GFX_ENABLE_VIDEO`	`Libgfxvideo`
`GFX_ENABLE_IME`	`Libgfxime`

Scaleform uses several build configurations, which are a combination of features and compiler options. In most cases, at least four configurations are available:

Shipping	Optimized without checks
Release	Optimized without checks
Debug	With debug info and checks
DebugOpt	Optimized with debug info and checks

2.2.1.1 Package Extraction for Windows OS

Some of our platforms may be packaged using tar.bz2. To extract those packages on Windows we recommend using a non-commercial package extractor which handles that format. For example, a tool such as 7-zip can extract tar.bz2 files.

2.2.2 The SDK Browser

Scaleform 4.3 includes an SDK Browser, which provides quick access to demos, tutorials and documentation, as well as other important Scaleform 4.3 items.
On Windows, the SDK Browser can be accessed via the Start Menu: *Programs->Scaleform->GFx SDK 4.3->Scaleform SDK Browser* or the shortcut added on the Desktop. On Mac, the SDK Browser can be accessed by going to *scaleform_gfx_4.3_macos/Bin/Browser/* and double clicking Browser.app.

2.2.3 Scaleform CLIK Installation

Scaleform 4 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 4.3->Documentation->Scaleform 4.3 – Getting Started with CLIK.pdf

2. Via Windows Explorer:
   C:/Program Files/Scaleform/GFx SDK 4.3/Doc/sf_4.3_getting_started_with_CLIK.pdf

3. Via the SDK Browser.

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

C:/Program Files/Scaleform/GFx SDK 4.3/Resources/AS2/CLIK/
C:/Program Files/Scaleform/GFx SDK 4.3/Resources/AS3/CLIK/
C:/Program Files/Scaleform/GFx SDK 4.3/Resources/Tools/

On other platforms:

scaleform_gfx_4.3_platform*/Resources/AS2/CLIK/*
scaleform_gfx_4.3_platform*/Resources/AS3/CLIK/*
scaleform_gfx_4.3_platform*/Resources/Tools/*

2.2.4 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 Scaleform 4.3 SDK has been installed. Scaleform Video provides highly optimized video support within Flash files and is tightly integrated with the UI and Flash development environment. Scaleform Video is currently supported on "PlayStation® 3" (PS3™), Xbox® 360, Wii™, Mac and PC.

Once Scaleform Video has been downloaded and installed, it will be found in the same directory as the Scaleform 4.3 SDK installation. The Scaleform Video Encoder icon will be added to both the desktop and the

Windows Start menu inside the Scaleform->GFx SDK 4.3->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 4.3/Bin/Tools/VideoEncoder/*

Location of Video Demo files: *C:/Program Files/Scaleform/GFx SDK 4.3/Bin/Data/AS2/Video/*

Location of Video Documentation:

1. *C:/Program Files/Scaleform/GFx SDK 4.3/Doc/sf_4.3_getting_started_with_video.pdf*

2. Start->Programs->Scaleform->GFx SDK 4.3->Documentation->Scaleform 4.3. – Getting Started with Video.pdf

3. SDK Browser.

2.2.5 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 is currently supported on the PC.

Scaleform IME must be downloaded and installed separately after the core Scaleform 4.3 SDK has been installed. It will be installed by default to:

C:/Program Files/Scaleform/GFx SDK 4.3/Bin/Data/AS2/IME/
C:/Program Files/Scaleform/GFx SDK 4.3/Bin/Data/AS3/IME/

2.2.6 Platform SDKs

For best results using the Scaleform code, we recommend installing the correct SDKs for each platform. For information on which SDKs are required for the Scaleform version you download, please see the Build History in the Scaleform Developer's center.

2.2.7 .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

2.2.8 Directory Structure

The Scaleform 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 Scaleform installation:

3rdParty/	3rd party packages
Apps/	
Samples	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 Scaleform application; a starting point for development
DrawText	Source code for the DrawText sample which demonstrates the DrawText API
ImageDelegate	Sample demonstrating image substitution using the ImageDelegate class.
RenderTexture	Source files for SWFToTexture and TextureInSWF samples.
Video	Sample Video demos
Kits	Source file for Kits, which are functional use-cases of Scaleform
Tutorial	Source code and projects for the Scaleform tutorial
Bin/	Various binary files (executables, samples, etc).
[Platform]	Binary executables (Scaleform Player, samples, etc) for a specific platform
Data/AS2/Samples	Sample ActionScript2 Flash files
FxPlayer	Scaleform Player data files
RenderTexture	RenderTexture samples files
SWFToTexture	RenderTexture samples files
ImageDelegate	ImageDelegate sample files
Data/AS2/Video	Scaleform video sample files
Data/AS2/IME	Scaleform IME sample files
Data/AS2/Kits	Sample data for ActionScript2 Kits
Data/AS3/Samples	Sample ActionScript3 Flash files
Data/AS3/IME	Sample IME ActionScript 3 files
Data/AS3/Kits	Sample data for ActionScript3 Kits
Doc/	Documentation
Include/	Scaleform include files (C++ convenience headers)
Lib/	Scaleform 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/	Scaleform source code
GFx	Scaleform core SDK source code
Render	Source for sample renderers (included with all releases).
Sound	Source for sample sound renderers (included with all releases).
Platform	Source for platform specific application class.

Scaleform platform names and #defines

Scaleform Platform	**Platform Names**	**#defines**
Win32	Windows	SF_OS_WIN32
x64	Windows for x86_64	SF_OS_WIN32
*-linux	Linux for * processor	SF_OS_LINUX
MacOS	MacOS X	SF_OS_DARWIN, SF_OS_MAC
Android	Android	SF_OS_ANDROID
iPhone	iPhone	SF_OS_IPHONE
PS3	Playstation 3	SF_OS_PS3
PSVita	PSVita	SF_OS_PSVITA
3DS	3DS	SF_OS_3DS
Wii	Wii	SF_OS_WII
Wii U	Wii U	SF_OS_WIIU
Xbox360	Xbox 360	SF_OS_XBOX360

Build Tools

	Standard, often gcc
Msvc80	Visual Studio 8.0 (2005)
Msvc90	Visual Studio 9.0 (2008)
Msvc10	Visual Studio 10.0 (2010)
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.

2.2.9 Building Scaleform

Scaleform libraries and executables can be built with Visual Studio projects, most commonly, or with make, which supports platforms that don't use the Visual Studio IDE. Projects and Solutions can be found in the 'Projects' directory and are organized by platform.

2.2.9.1 Using make

Many platforms can be built using 'make', rather than using an IDE, such as Visual Studio. The Scaleform SDK comes with a readme file, *readme_make.txt*, which provides compiler instructions for make-based platforms. This applies to the following platforms:

- Linux
- MacOS
- Android™
- iPhone®
- PS3™
- Nintendo 3DS™
- Wii™
- Wii U™

Before building Scaleform 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 Scaleform.

Below are the steps for using make:

1. Create a Makeconfig file in the top level Scaleform directory (or edit the example one that is provided there). Here’s what to do using the PS3 as an example. The file should contain:

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

   d. 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:

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

2. To build a particular platform, or a configuration for a platform, using the following command line syntax, which specifies platform and configuration:

   ```
   a. $ make P=PS3
   b. $ make P=PS3 C=Debug
   ```

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

C=Release+NoRTTI". 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.

2.2.9.2 Example 'make' usage:

Common usage:

```
make P=Android C=Debug
make P=3DS C=Debug+NoRTTI
```

Build for PS3 using the SNC compiler

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

Build for PS3 using gcc

```
make P=PS3
```

Build for iOS, using non-SCU lumping (see next section on SCU builds)

```
make P=iPhone/armv7 SCU=0
```

Build with verbose output:

```
make P=wii C=Shipping+NoRTTI VERBOSE=1
```

Build a Release, x64 lib with RTTI for Mac:

```
make P=MacOS/x86_64 C=Release
```

Build for the iPhone simulator or armv7:

```
make P=iPhone/sim
make P=iPhone/armv7 C=Release+NoRTTI
```

Build a single target (lib or executable):

```
make P=MacOS C=Release+NoRTTI Lib/MacOS-x86_64/Release_NoRTTI/libgfx.a
```

Compile 4 files in parallel (-j4):

```
make P=PS3 C=Shipping+NoRTTI -j4 Bin/PS3/FxPlayer_Shipping_NoRTTI.elf
```

2.2.9.3 SCU Builds

For some platforms and libraries, Scaleform builds using groups of files combined into single compilation units (SCUs). This technique is also called Lumping or Unity builds. SCU builds have the benefits of reduced compilation time and smaller library sizes. The Scaleform SDK generally provides debug symbols in all libraries so that developers can have more information when debugging. If debug symbols are not desired in an executable, they can be easily stripped at link time.

Without SCU builds, library sizes can grow significantly, since redundant debug symbol information is placed in the library by multiple files. Please note that even though the library size may increase or decrease depending on whether SCU builds are being used, the size is being affected by the amount of debugging symbols and not by Scaleform library code.

SCU builds combine a logical group of files into a single compilation unit, using a grouping file which ends in _All.cpp. The grouping file includes the other files using a series of #include statements. For example, here is the SCU grouping file for the AS3 Obj Accessibility directory:

```
Src/GFx/AS3/Obj/Accessibility/AS3_Obj_Accessibility_All.cpp
…
#include "AS3_Obj_Accessibility_ISearchableText.cpp"
#include "AS3_Obj_Accessibility_ISimpleTextSelection.cpp"
#include "AS3_Obj_Accessibility_Accessibility.cpp"
#include "AS3_Obj_Accessibility_AccessibilityImplementation.cpp"
#include "AS3_Obj_Accessibility_AccessibilityProperties.cpp"
```

This all happens behind the scenes, so developers shouldn't need to worry about it. However, it is possible to build without the SCU technique as well. In Visual Studio, when SCU builds are available, you will see two sets of projects, for example:

SCU Versions:

- AS2_SCU
- AS3_SCU
- AIR_SCU

Non-SCU Versions:

- AS2
- AS3
- AIR

You can manually build the non-SCU projects if you would like.

When building with make, SCU builds are also automatically used for some platforms. In this case, if SCU builds are not desired, simply use the option 'SCU=0' as part of the make command line, for example:

```
make P=PS3+snc  C=Release+NoRTTI  SCU=0
```

2.3 Scaleform SDK for Windows

The Scaleform library is supported on Windows® XP (32 bit, SP3), Vista (32/64 bit, SP2) and Windows 7 (32/64 bit). Support for Windows 8, include Metro style apps is available in Scaleform 4.3. Scaleform is fully compatible with Visual Studio 2008 (9.0) and higher versions.

2.3.1.1 Scaleform Library Configurations

Linking your product against the correct Scaleform precompiled libs ensures smooth integration with the latest version of Scaleform.
Scaleform SDK precompiled libs are located by default in \Program Files\Scaleform\GFx SDK 4.3\Lib\[Platform]\[MSVC Version]\
The four sub folders under this directory describe the runtime library the libs were compiled against.

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

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

Shipping	Scaleform Multi-threaded Shipping libs
Runtime Library	Multi-threaded (/MT)
Debug Information Format	C7 Compatible (/Z7)
Optimization	Full Optimization (/Ox)
	Optimized release code with no logging, stats, etc

2.3.1.2 MSVC Project Settings

To properly execute the Scaleform sample demos (such as **SWFToTexture**, and **TextureInSWF)** you must change the "Working Directory" of the project to the data directory: C:\Program Files (x86)\Scaleform\GFx SDK 4.3\Bin\Data

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**

2.4 Scaleform SDK for Mac

Scaleform SDK packages include Scaleform libs, and a sample player for MacOS using Cocoa and GL.

2.4.1.1 Requirements

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

- MacOS Snow Leopard (10.6)/Lion (10.7)
- Xcode (versions 3.2.6 or higher) developer toolset for Mac/iPhone/iPad

Please note: PPC is not supported

2.4.1.2 Installation

Install Scaleform SDK for MacOS. Listed below are the packages for MacOS that need to be installed.
For evaluation purposes, install the latest versions of these eval packages:

- sf_4.3_eval_macos_i686.tar.bz2 or sf_4.3_eval_macos_x86_64.tar.bz2

Licensed source users should install these src packages:

- sf_4.3_src_macos_i686.tar.bz2 or sf_4.3_src_macos_x86_64.tar.bz2

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

- sf_4.3_lib_macos_i686.tar.bz2 or sf_4.3_lib_macos_x86_64.tar.bz2

Optional packages for Video add-ons

- Video
    - sf_4.3_{eval/lib/src}_video_macos_i686.tar.bz2

2.4.1.3 Building for MacOS X (10.6 and 10.7)

After installing the Scaleform MacOs SDK, open terminal.app and go to the root of the Scaleform directory.
To build Scaleform (and samples) for both x86 and x64:

```
 $ make P=MacOS
```

Executables are placed in Bin/MacOS.

To build Scaleform (and samples) for a single cpu:

```
   $ make P=MacOS/i686
```

or

```
   $ make P=MacOS/x86_64
```

Please note that zlib is expected to be provided by the system.

2.4.1.4 Running Executables

Executables are placed in *Bin/MacOS-i686* or *Bin/MacOS-x86_64*.

The included sample player can be run from the Finder or the command line.

From the command line: *$ Bin/MacOS-i686/FxPlayer.app/Contents/MacOS/FxPlayer <swf file>*

From the Finder, drag a SWF file to the Dock icon, the executable, or the player's window. The AS2 and AS3 versions of the HUD and Menu kits can be run from the Finder (*Bin\Data\AS3\Kits\Menu or Bin\Data\AS3\Kits\HUD*) or from the command line.

From the command line (Menu Kit): *$Bin/MacOS-i686/MenuDemo.app/Contents/MacOS/MenuDemo*

2.5 Scaleform SDK for Consoles

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 files for each console, and follow the installation instructions. Installing the console package will add additional directories into the default Scaleform 4.3 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 Scaleform directory structure for more information.

Specific installation procedures for Xbox360®, Wii™, PS3™, PS Vita™ and 3DS™ are provided in the following subsections.

Each console installation may require a particular platform SDK version. Please see the build history notes in the Scaleform Developer Center for more details.

2.5.1 Scaleform SDK for Xbox 360

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

2.5.1.1 Building and Running the Players/Samples

After installing the Scaleform Xbox 360 SDK, open the Scaleform 4.3 Xbox 360 Demos.sln Visual Studio solution, accessible via the Start Menu and the Scaleform SDK Browser.

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, Release and Shipping.

You are now ready to build the Scaleform 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 Scaleform 4.3 Xbox 360 Demos solution, only Player, SWFToTexture, TextureInSWF, and PlayerTiny 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 Player, SWFToTexture, TextureInSWF, and PlayerTiny displayed in the Xbox Launcher because the other projects included in the solution (GFx_Xbox360, GFxExpat, and Sound) are not executables.

When running the FxPlayer on Xbox 360® you can use the following gamepad commands:

- Use the D Pad to move focus (Arrow keys)
- Press A to select what has focus (Enter)
- Press Y to toggle Wireframe
- Press X to toggle Playback Info
- Press B to Enable Fast Forward
- Press RB to play to the next movie
- Press LB to play to the previous movie
- Press Start to Pause/Play
- Press Back to toggle gamepad control pass-thru mode

When the <back> Key is pressed, gamepad events are no longer intercepted by the FxPlayer and instead are mapped to Flash keyboard events as below. This allows developers to test how their content responds to controller input using a typical mapping scheme.

- A : Enter,
- B : Escape,
- Start : ~,
- LT : PgUp,
- RT : PgDn,
- LB: Home,
- RB : End,
- LTH : Insert
- RTH : Delete

Optionally, you can specify a new Flash (SWF) file by changing the FXPLAYER_FILENAME define in FxPlayerXbox360.cpp:
Example:

```
#define FXPLAYER_FILENAME "Window.swf"
```

The default "Window.swf" demonstrates the use of the arrow pad buttons to change focus and the A button for execution.

2.5.1.2 Building the Source

If you have the Scaleform source installation for Xbox 360, you can build the Scaleform SDK itself. Here’s how to build the Scaleform 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 Scaleform library. Open {GFX_SDK}/Projects/Xbox360/Msvc90/SDK/GFx 4.3 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/Msvc90/SDK/GFx 4.3 Xbox 360 SDK.sln solution.

3. Select the desired configuration and choose Build Solution from the build menu.

Build Settings:

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

2.5.2 Scaleform SDK for PS3

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

2.5.2.1 System Requirements

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 renderer libs libgfxrender_ps3.a is 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.

2.5.2.2 Renderer libs

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

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

2.5.2.3 Using make

Before building Scaleform with make, you should have a Linux tool set installed, including the make command itself.

1. See the earlier section, Sec 2.2.9.1 on Building Scaleform using Make

2. Build Scaleform libs and samples.

   a. `$ make P=PS3`

   b. To build using SNC:

      `$ make P=PS3+snc`

   c. 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 Scaleform Player; use the L1 and R1 buttons.

4. If this is an evaluation version, place the license key in "*$(CELL_SDK)/FxPlayer/sf_license.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/Data/AS2/Samples/SWFToTexture* and *Bin/Data/AS2/Samples/RenderTexture* to *$(CELL_SDK)/*F*xPlayer*.

2.5.2.4 Using Visual Studio

As an alternative to using Make files, Visual Studio solutions and projects are provided to build the Scaleform demos (and libraries if you have the source installation for 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 Scaleform source, you need to run the source installers for PS3. Open up the Visual Studio solution called Scaleform 4.3 PS3 SDK.sln which is located in the *Projects/PS3/Msvc90/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 Scaleform 4.3 PS3 Demos.sln which is located in the *Projects/PS3/Msvc90/Samples* folder. Next, select the appropriate configuration and choose *Build Solution* from the Build menu.

2.5.2.5 Player Controls

When running the Scaleform Player on "PS3" you can use the following gamepad commands:

- Use the D Pad to move focus (Arrow keys)
- Press Cross to select what has focus (Enter)
- Press Triangle to toggle wireframe
- Press Square to toggle HUD info
- Press Circle to enable fast forward mode
- Press R1, L1 to play to the next, previous movie
- Press L2 to toggle anti-aliasing
- Press R2 to enable mouse; show/hide cursor
- Press Start to pause/play
- Press Select to toggle gamepad control pass-thru mode

When the <select> Key is pressed, gamepad events are no longer intercepted by the FxPlayer and instead are mapped to common Flash keyboard events as defined in the FxPlayerAppBase ctor (PadKeyCommandMap section). This allows developers to test how their content responds to controller input using a typical mapping scheme.

NOTE: The "Window.swf" demonstrates the use of the D Pad arrow buttons to change focus and the Cross (X) button for execution. You must copy the "Window.swf" to */usr/local/cell/FxPlayer/flash.swf* in order for the FxPlayer to start properly. Additional SWF/GFX files can be copied to the directory and cycled through using the L1/L2 buttons.

2.5.2.6 PS3 Renderer Support

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

libgcm:

Source: *Src/Render/PS3/* 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:

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 Scaleform players that utilize libcgm and PSGL renderers.

With libgcm renderer:

• PlayerGCM	Basic SWF player
• PlayerTiny	Tiny SWF player
• SWFToTextureGCM	Demo of rendering SWF to texture
• TextureInSWFGCM	Demo of embedding user texture in SWF

With PSGL renderer:

• PlayerGL	Basic SWF player
• PlayerTinyGL	Tiny SWF player
• SWFToTextureGL	Demo of rendering SWF to texture
• PlayerTextureInSWFGL	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/Samples/Common/OpenGLPS3App.**

SWFToTexture/TextureInSWF also require *Apps/Samples/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
```

2.5.2.7 Build Configurations

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

Shipping	Optimized
Release	Optimized
Debug	Non-optimized, with full debugging
DebugOpt	Optimized, with full debugging
NoRTTI	Disable RTTI & exceptions
NoThreads	Disable Scaleform thread support (SF_ENABLE_THREADS)

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.

2.5.2.8 Integration Notes

Scaleform 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);
```

2.5.2.9 Thread Support

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

2.5.3 Scaleform SDK for PS Vita

2.5.3.1 Requirements

Please make sure that you have the correct Vita SDK and tools installed and functioning on your development computer.

2.5.3.2 Installation

Install the Scaleform SDK for PS Vita by unzipping the package to a directory on your system. Listed below are the packages for PS Vita that need to be installed.
For evaluation purposes, install the latest versions of these eval packages:

- sf_4.3_psvita_vc90_eval.exe

Licensed source users should install these src packages:

- sf_4.3_psvita_vc90_src.exe

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

- sf_4.3_psvita_vc90_lib.exe

2.5.3.3 Building the Players/Samples

After installing the Scaleform PS Vita SDK, open the Scaleform 4.3 PS Vita Demos Visual Studio solution located here: *"Projects/PSVita/Msvc90/Demos/GFx 4.3 PSVita Demos.sln"*
From this solution, the Release, Debug, DebugOpt and Shipping configurations of FxPlayerMobile and FxPlayerTiny can be build and run. If you have the Scaleform source installation for the PSVita, you can build the Scaleform SDK itself. Here's how to build the Scaleform library for the PSVita from the source code:

1. Open "Projects/PSVita/Msvc90/SDK/GFx 4.3 PSVita SDK.sln"

2. Select the desired configuration and choose Build Solution from the build menu

You can also build the Vita code using make. Please see section 2.2.9 above on building Scaleform with make, for more details.

2.5.3.4 Running Players and Samples

The following sample players are included in both the Demo and Source solution:

FxPlayerTiny	A simple reference for beginning an integration with Scaleform
FxPlayerMobile	A more advanced player that includes a drop-down HUD

Place your content (SWFs, etc.) in a folder named "FxPlayer" in the root of the Vita's "File Serving Directory". Make sure that one of your SWFs is named "flash.swf", since this is the default startup file that FxPlayerMobile looks for.

2.5.3.5 Evaluation Versions

See section 2.10 below, for details on placement of the license key.

2.5.4 Scaleform SDK for Nintendo Wii

2.5.4.1 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/sf_license.txt*.

The sample applications load these files:

Player	flash.swf (use any SWF or GFx file)
PlayerTiny	Window.swf (use any SWF or GFX file)
SWFToTexture	Copy all files in *Bin/Data/AS2/Samples/SWFToTexture* to *dvddata/FxPlayer*.
TextureInSWF	Copy all files in *Bin/Data/AS2/Samples/RenderTexture* to *dvddata/FxPlayer*.

FxPlayer controls:

A	Enter key and mouse click
B	Escape key
Open/close HUD	
Toggle HUD focus	
Home	Pause
- and +	Change Flash file (other SWF and GFX files in *dvddata/FxPlayer*)
C	Toggle fast forward
Z	Toggle gamepad control pass-thru mode

When the <Z> Key is pressed, gamepad events are no longer intercepted by the FxPlayer and instead are mapped to common Flash keyboard events as defined in the FxPlayerAppBase ctor (PadKeyCommandMap section). This allows developers to test how their content responds to controller input using a typical input mapping.

2.5.4.2 Building the Samples

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

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

Shipping	Optimized without checks
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 Scaleform thread support (SF_ENABLE_THREADS; 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 GFX_ENABLE_VIDEO`" to the preprocessor text in Target Settings, C/C++ Preprocessor group.

Projects for rebuilding Scaleform 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 Scaleform 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.
Scaleform 4.3

2.5.5 Scaleform SDK for Nintendo 3DS

Scaleform SDK for Nintendo 3DS

2.5.5.1 Requirements

Please make sure that you have the correct CTR SDK and tools installed and functioning on your development computer.

2.5.5.2 Installation

Install the Scaleform SDK for 3DS by unzipping the package to a directory on your system. Listed below are the packages for the 3DS that need to be installed.
For evaluation purposes, install the latest versions of these eval packages:

- sf_4.3_eval_3ds.tar.bz2

Licensed source users should install these src packages:

- sf_4.3_src_3ds.tar.bz2

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

sf_4.3_lib_3ds.tar.bz2

2.5.5.3 Building the code

After installing the Scaleform 3DS SDK, open Cygwin and go to the root of the Scaleform directory.
To build Scaleform (and samples):

```
$ make P=3DS
```

Executables are placed in Bin/3DS.

Please see section 2.2.9 for more details on building Scaleform with make.

2.5.5.4 Running Players and Samples

To run FxPlayerMobile and/or FxPlayerTiny, first select the Flash files to use by editing Projects/3DS/Makefile, STARTUP_FLASH and FxPlayer_CONTENT. These files are stored in the rom image during build. You can also add more SWF files to Obj/3DS/rom_FxPlayerMobile/FxPlayer and rebuild the cci.

Once your content has been selected and the player has been built, open up the CTR debugger to load and run the executable.

2.5.5.5 Evaluation Versions

See section 2.10 for details on placement of the license key.

2.6 Scaleform SDK for Mobiles

Developers who have registered with Scaleform may also download and install the mobile packages for any of the mobile platforms they are registered to use. Simply download the stand-alone package for each platform, and follow the installation instructions. The package will contain the necessary source code, examples and documentation needed for each platform. See the section in this document describing the Scaleform directory structure for more information.

Specific installation procedures for Android and iOS are provided in the following subsections.

2.6.1 Scaleform SDK for Android

2.6.1.1 Requirements

This software needs to be installed on your development computer for the proper operation of Scaleform:

- Latest Android SDK for your operating system
- Latest Android NDK for your operating system
- The appropriate Android Platform API(s) for the platform(s) you are targeting

2.6.1.2 Installation

First install the Scaleform SDK for Android. Listed below are the packages for Android that need to be installed.

For evaluation purposes, install the latest versions of this eval package:

- sf_4.3_eval_android.tar.gz

Licensed source users should install this src package:

- sf_4.3_src_android.tar.gz

Licensed library (non-source) users should install this package:

- sf_4.3_lib_android.tar.gz

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

2.6.1.3 Building and Running the Players/Samples

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

```
export JAVA_HOME := <path to JDK>
ANDROID_SDK := <path to SDK>
ANDROID_NDK := <path to NDK>
```

```
export PATH := $(JAVA_HOME)/bin:$(ANDROID_SDK)/tools:$(ANDROID_SDK)/platform-tools
```

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 the Android_SDK is installed to C:\Android/SDK:
"`export ANDROID_SDK := /cygdrive/c/Android/SDK`".

2. Build the Scaleform libs and samples:

   ```
   $ make P=Android
   ```

3. Copy the appropriate APK file found in Bin/Android to the device. Navigate to the location on the device you placed the APK and install it to the device.

4. Copy a SWF file to /sdcard/*flash.swf*. Other SWF and GFx files in that directory can be viewed in Scaleform Player via the arrows in the pop-up HUD accessed using the Menu Key.

5. If this is an evaluation version, place the license key in "/sdcard/*sf_license.txt*".

6. Now run the demo application from your application launcher and it will load the flash.swf copied to "/sdcard".

2.6.1.4 Building the Source

If you have the Scaleform source installation, you can build the Scaleform SDK itself. The instructions for building the Scaleform library for Android from the source code are the same as above. Make will detect the presence of the full source tree and recompile it accordingly.

2.6.2 Scaleform SDK for iOS

2.6.2.1 Requirements

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

- Mac OS X Snow Leopard
- Xcode 3.2.5, Xcode 4
- Recent SDK version

2.6.2.2 Installation

First install the Scaleform SDK for iOS. Listed below are the packages for iOS that need to be installed.

For evaluation purposes, install the latest version of this eval package:

- sf_4.3_eval_ios.tar.bz2

Licensed source users should install this src package:

- sf_4.3_src_ios.tar.bz2

Licensed library (non-source) users should install this package:

- sf_4.3_lib_ios.tar.bz2

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

2.6.2.3 Building and Running the Players/Samples

After extracting the Scaleform iOS package, open the Scaleform 4.3 iPhone Xcode solution, located in "*Projects\iPhone\Xcode3\GFx 4.3 iPhone SDK*\".

Please ensure that your iOS device is powered on and connected to your system before building the demos. Within Xcode, select the configuration you'd like to build from the drop down menu on the toolbar at the top left of the screen. Available configurations include Debug, DebugOpt, Release and Shipping.

You are now ready to build the Scaleform demo project. To do so, select "Build and Run" from Xcode's menu bar. This will initiate a build of the project within the solution. When compiling and linking of the executable completes, by default, the solution will deploy the executable and all necessary content (particularly GFX and SWF files) to your iOS device.

Please note that the iOS demo app looks for a single SWF named "flash.swf" during startup. If that SWF doesn't exist, then the player will start up with a blank screen. Content can also be loaded into the player manually via iTunes. First, plug the device in and launch iTunes. Once launched, head over to the "File Sharing" section in the Apps tab, select FxPlayerMobile in the Apps window and then either drag a SWF to the "FxPlayerMobile Documents" window or click on the "Add…" button to add the SWFs. The content will sync on the fly.

2.6.2.4 Building the Source

If you have the Scaleform source installation, you can build the Scaleform SDK itself. Here's how to build the Scaleform library for iOS from the source code:

1. Open up a terminal window and go the root of your Scaleform iOS package. Once there, build the lipo version of the libs (choosing the appropriate configuration such as Release) via make (for example:

```
make P=iPhone/armv7 C=Release
```

2.6.3 Scaleform SDK for Windows Phone

2.6.3.1 Requirements

The following must be installed to use the Autodesk Scaleform Mobile SDK for Windows Store:

- Windows 8
- Windows Phone 8 SDK
- Visual Studio 12

2.6.3.2 Installation

For evaluation purposes, install the latest version of this eval package:

- sf_4.3.25_wphone9_vc11_eval.exe

Licensed source users should install this src package:

- sf_4.3.25_wphone9_vc11_src.exe

Licensed library (non-source) users should install this package:

- sf_4.3.25_wphone9_vc11_lib.exe

2.6.3.3 Building and Running the Players/Samples

After extracting the Windows Phone package, open the Scaleform 4.3 Visual Studio solution located in your Projects directory.

Select the configuration you’d like and build your desired project.

This will initiate a build of the project within the solution. When compiling and linking of the executable completes, by default, the solution will deploy the executable and all necessary content (particularly GFX and SWF files) to your device.

Please note that our player looks for a single SWF named “flash.swf” during startup. If that SWF doesn’t exist, then the player will start up with a blank screen.

2.6.4 FAQs Related to Lifecycle Management in Mobile Player

This section contains few of the questions that developers may have when using the Scaleform SDK for Mobiles.

1. How should Scaleform operations change after receiving an onPause notification?

In the main thread, the Advance function of GFx::Movie instances should not be called when the device is suspended. The GFx::Movie buffers changes between Advance and NextCapture calls, so if the device is suspended for a long time, this can result in a large amount of unnecessary memory usage. Also, doing so will cause playback to appear to continue while the device was suspended, which is generally undesirable. The rendering thread will be automatically suspended by the system, so no changes there are required.

2. Is it necessary to unload Scaleform on suspend and reload Scaleform on resume?

It shouldn't be necessary to have any kind of teardown/bring-up during the standard suspend/resume cycle. As long as all pending drawing is completed, and no attempts to advance or display the movie are made while the device is suspended, everything should work correctly.

3. How do I inform the render HAL that a new GL context has been created after the device resumes?

Simply call Render::GL::HAL::ResetContext. The HAL should flush out any active handles to textures/shaders/buffers and re-create them with the new GL context the next time they're used.

4. Is it possible to unload Scaleform while preserving the state of loaded movies?

Unfortunately, it is not. Fortunately, it should not be necessary to restart Scaleform after a suspend/resume cycle as long as you're careful with your call to ResetContext and as long as you don't Advance/Capture/NextCapture/Display while the application is suspended.

2.7 *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 player from the Flash authoring environment. In order to install them follow the steps outlined below.

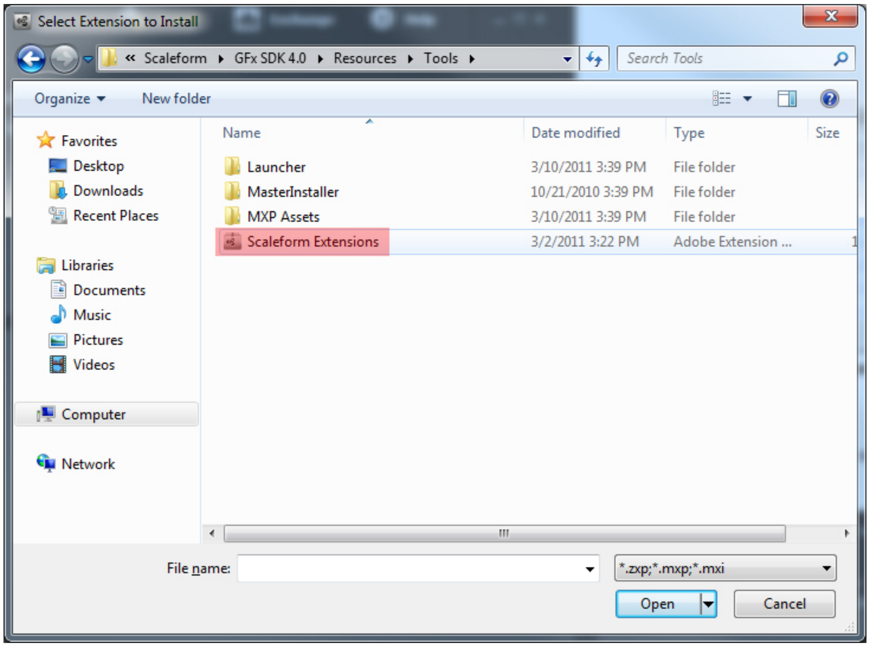
1. Launch the Adobe Extension Manager, which can be found on Windows in Windows®>Start Menu>Programs>Adobe Extension Manager CS3. The default installation location is *C:/Program Files/Adobe/Adobe Extension Manager.*

   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.

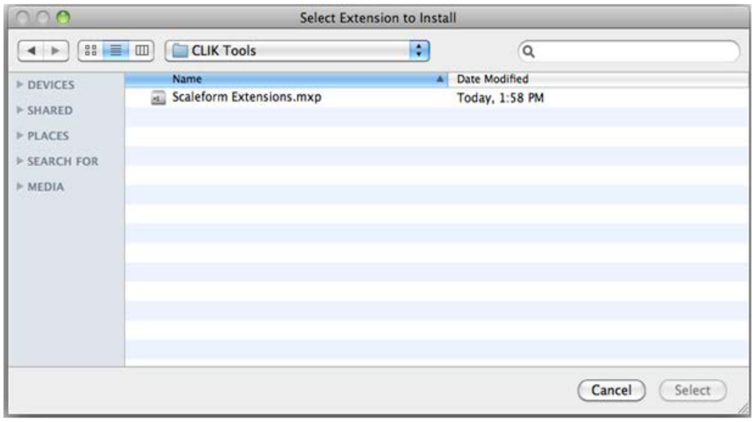
3. Browse to the following directory:
   On Windows: *C:/Program Files/Scaleform/GFx SDK 4.3/Resources/Tools/*
   On Mac: *scaleform_gfx_4.3_macos/Resources/Tools/*

4. Double click the MXP file in this directory: *Scaleform Extensions.mxp*.



Name	Date modified	Type	Size
Launcher	3/10/2011 3:39 PM	File folder	
MasterInstaller	10/21/2010 3:39 PM	File folder	
MXP Assets	3/10/2011 3:39 PM	File folder	
Scaleform Extensions	3/2/2011 3:22 PM	Adobe Extension ...	1

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

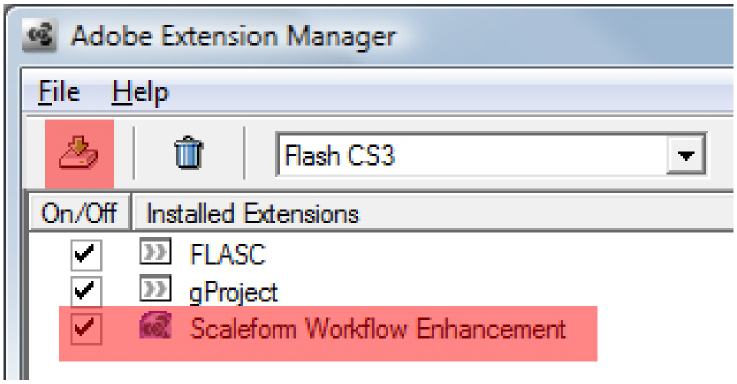


Name	Date Modified
Scaleform Extensions.mxp	Today, 1:58 PM

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

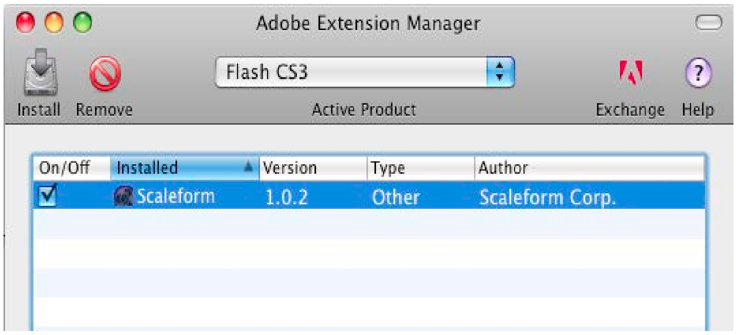
5. Press *Accept* on the license window.

6. Press *OK* in the dialogue telling you the extension was successfully installed.



On/Off	Installed Extensions
☑	FLASC
☑	gProject
☑	Scaleform Workflow Enhancement

Figure 3: Adobe Extension Manager in CS3 on Windows.



On/Off	Installed	Version	Type	Author
☑	Scaleform	1.0.2	Other	Scaleform Corp.

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 or 3.0).

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.

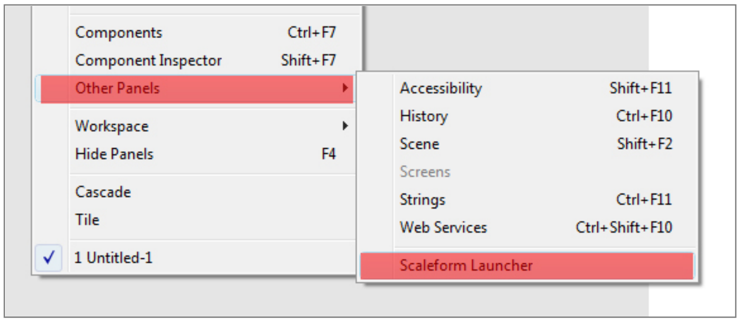


Figure 5: Add the Scaleform Launcher via Other Panels on Windows.

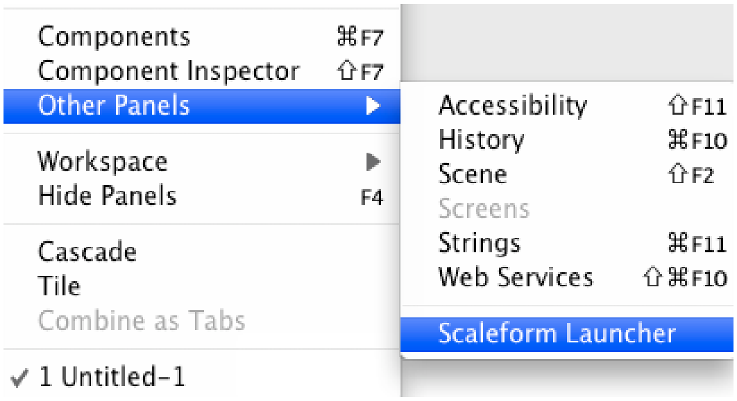


Figure 6: Add the Scaleform Launcher via Other Panels on Mac.

2.8 Installing the Scaleform Launcher Panel for CS4 & CS5

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

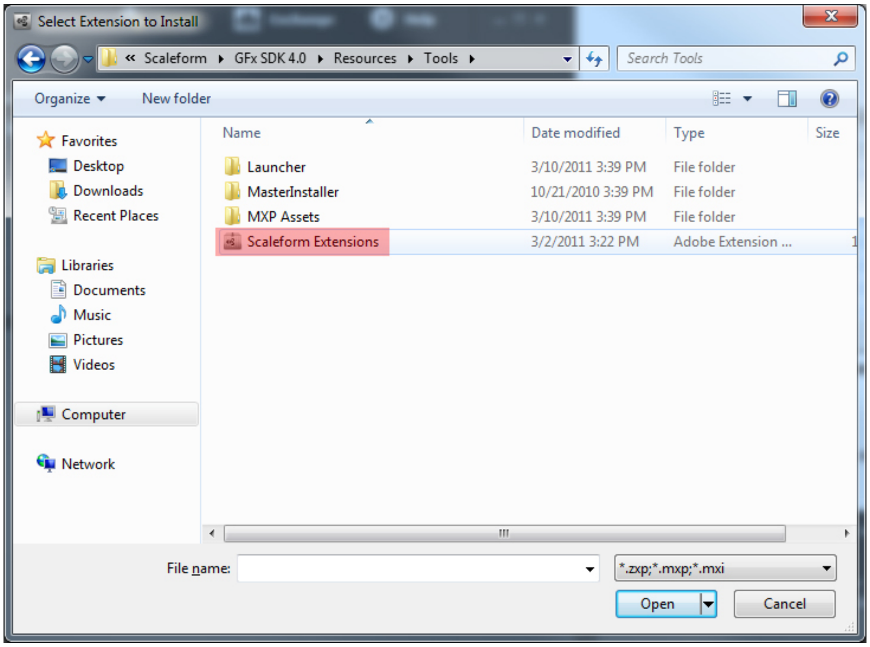
1. Launch the Adobe Extension Manager from CS4 or CS5 by clicking *Help* in the top Flash menu, and then select *Manage Extensions* from the dropdown. **IMPORTANT:** If the Extension Manager is not

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 4.3/Resources/Tools/*
On Mac: *scaleform_gfx_4.3_macos/Resources/Tools/*

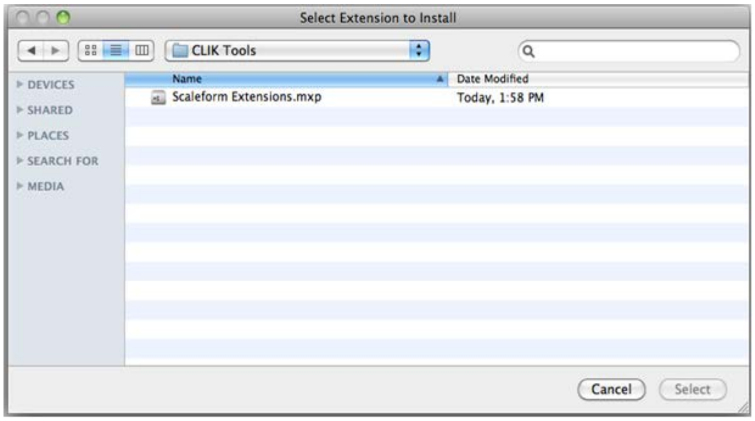
4. Double click the MXP file in this directory: *Scaleform Extensions.mxp*.



Name	Date modified	Type	Size
Launcher	3/10/2011 3:39 PM	File folder	
MasterInstaller	10/21/2010 3:39 PM	File folder	
MXP Assets	3/10/2011 3:39 PM	File folder	
Scaleform Extensions	3/2/2011 3:22 PM	Adobe Extension ...	1

.zxp;.mxp;*.mxi

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

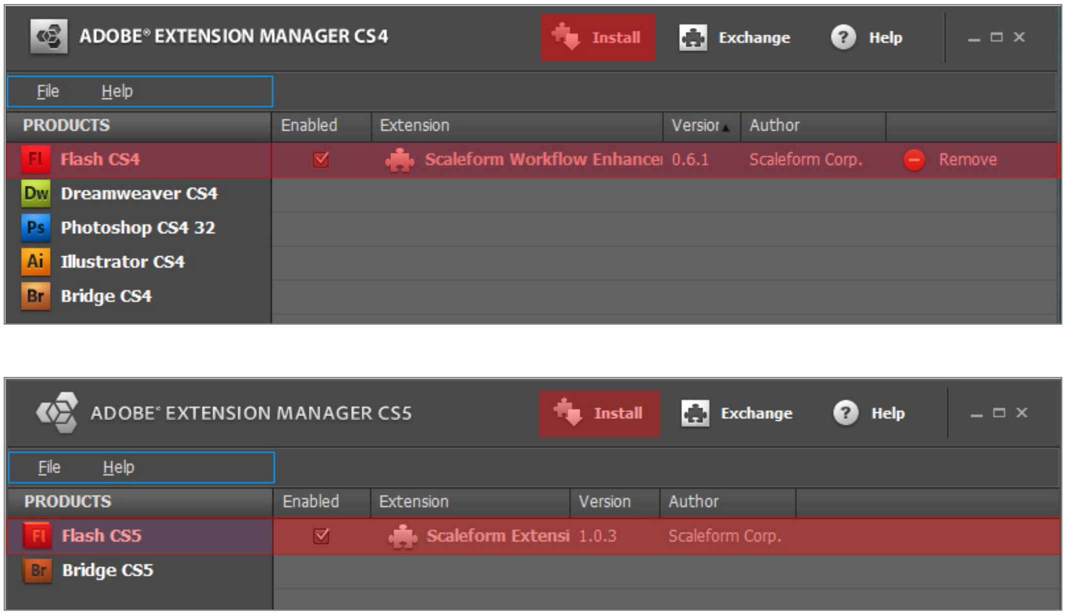


Name	Date Modified
Scaleform Extensions.mxp	Today, 1:58 PM

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 or CS5.



Enabled	Extension	Versior	Author	
☑	Scaleform Workflow Enhancei	0.6.1	Scaleform Corp.	Remove

Enabled	Extension	Version	Author
☑	Scaleform Extensi	1.0.3	Scaleform Corp.

Figure 9: Adobe Extension Manager in CS4 (top) & CS5 (bottom).

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.

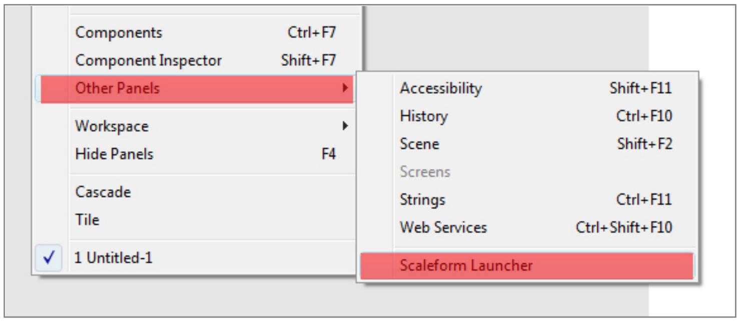


Figure 10: Displaying the Scaleform Launcher panel on Windows.

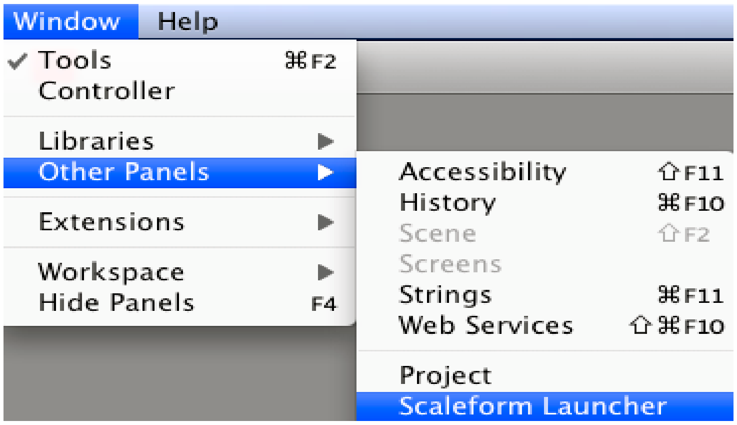


Figure 11: Displaying the Scaleform Launcher panel on Mac.

2.9 Using the Scaleform Launcher Panel

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

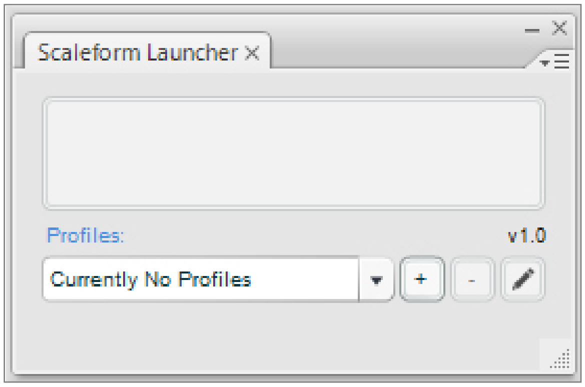


Figure 12: The Scaleform Launcher Panel

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 for Windows comes with executables for AMP (Scaleform's Analyzer for Memory and Performance), Debug and Release, 64 and 32-bit builds of the Scaleform Player. Scaleform for Mac comes with executables for FxPlayer_DebugOpt (AMP), Debug and Release, Intel and PowerPC builds of the Scaleform Player. 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 Scaleform 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.

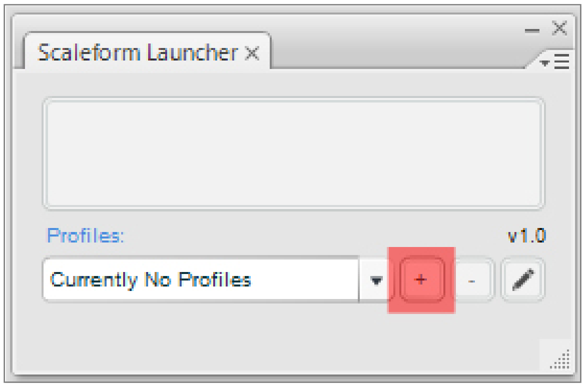
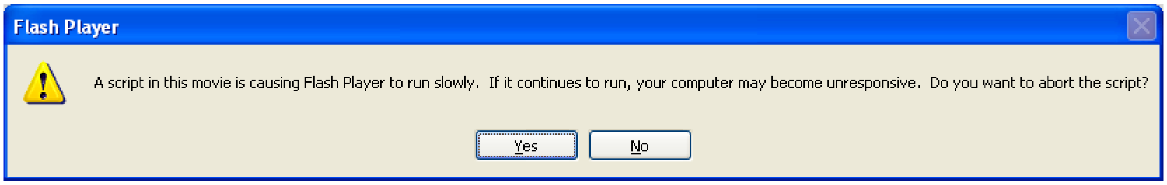


Figure 13: Add a new profile by pressing the plus button

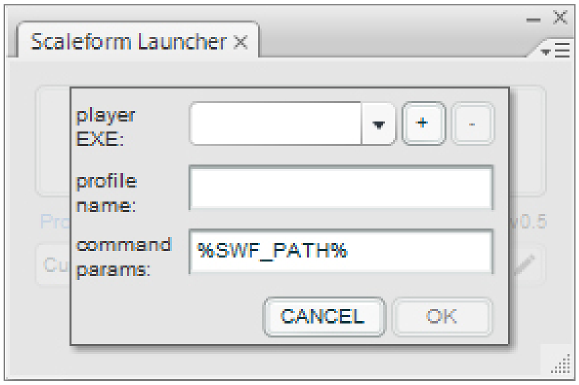
NOTE: If a window indicating the script is causing Flash to run slowly pops up, select *No* to continue normal operation. This is a known issue that will be corrected in future releases.



A script in this movie is causing Flash Player to run slowly. If it continues to run, your computer may become unresponsive. Do you want to abort the script?

Figure 14: Abort script dialogue.

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



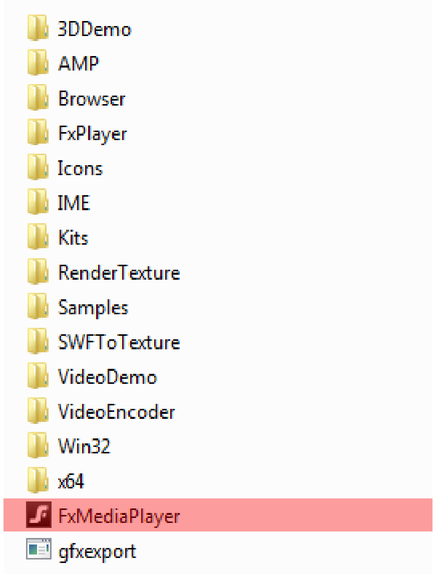
player EXE:

profile name:

command params: %SWF_PATH%

Figure 15: Add new profile dialogue.

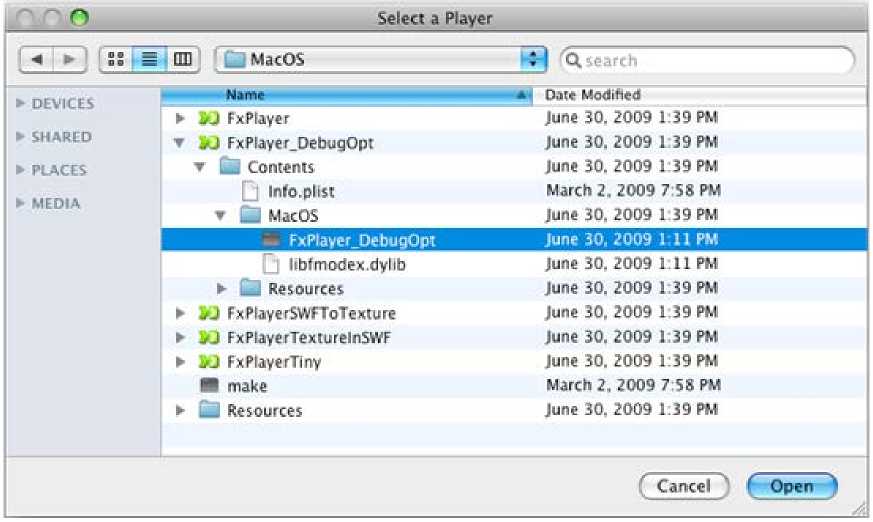
3. We recommend choosing the *GFxMediaPlayer* executable on Windows platform because it provides performance monitoring statistics in the player window; however, any Scaleform player executable may be used. The *GFxMediaPlayer* file is located in *C:/Program Files/Scaleform/GFx SDK 4.3/Bin*



3DDemo
AMP
Browser
FxPlayer
Icons
IME
Kits
RenderTexture
Samples
SWFToTexture
VideoDemo
VideoEncoder
Win32
x64
FxMediaPlayer
gfxexport

Figure 16: Browse to the *GFxMediaPlayer* executableon Windows.

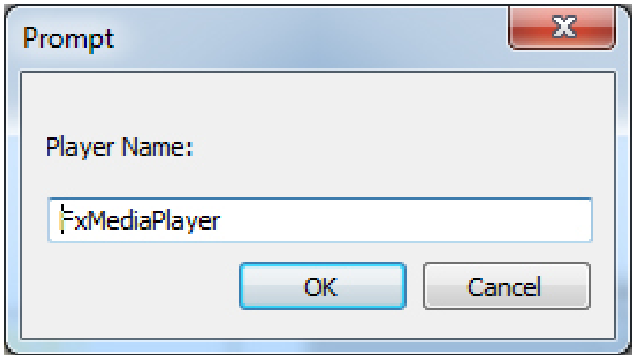
On Mac, choose the *GFxPlayer_DebugOpt* executable located in *scaleform_gfx_4.3_macos/Bin/MacOS/FxPlayer_DebugOpt.app/Contents/MacOS*



Name	Date Modified
▶ FxPlayer	June 30, 2009 1:39 PM
▼ FxPlayer_DebugOpt	June 30, 2009 1:39 PM
▼ Contents	June 30, 2009 1:39 PM
Info.plist	March 2, 2009 7:58 PM
▼ MacOS	June 30, 2009 1:39 PM
FxPlayer_DebugOpt	June 30, 2009 1:11 PM
libfmodex.dylib	June 30, 2009 1:11 PM
▶ Resources	June 30, 2009 1:39 PM
▶ FxPlayerSWFToTexture	June 30, 2009 1:39 PM
▶ FxPlayerTextureInSWF	June 30, 2009 1:39 PM
▶ FxPlayerTiny	June 30, 2009 1:39 PM
make	March 2, 2009 7:58 PM
▶ Resources	June 30, 2009 1:39 PM

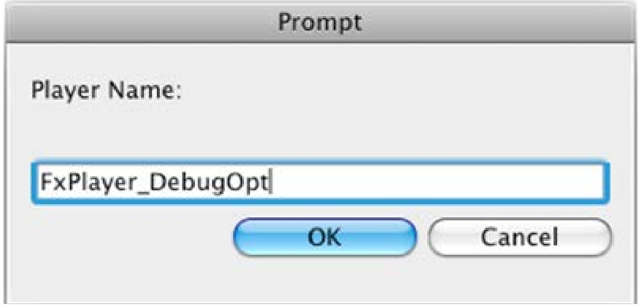
Figure 17: Browse to the *GFxPlayer_DebugOpt executable* on Mac.

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



Player Name: FxMediaPlayer

Figure 18: Player name prompt on Windows.



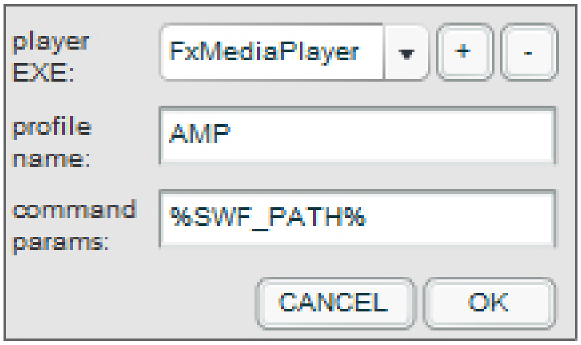
Player Name: FxPlayer_DebugOpt

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 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 player with no HUD.

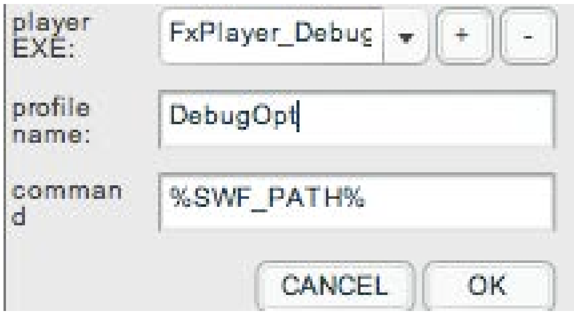


player EXE: FxMediaPlayer

profile name: AMP

command params: %SWF_PATH%

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



player EXE: FxPlayer_Debug

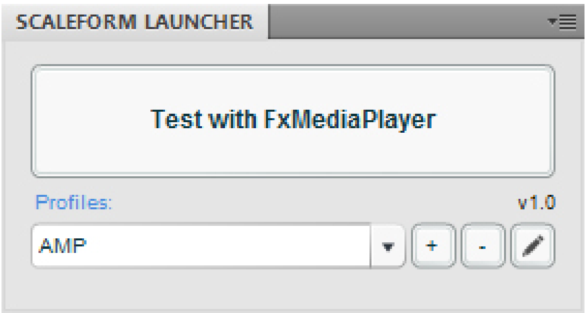
profile name: DebugOpt

command: %SWF_PATH%

Figure 21: GFxPlayer DebugOpt with default command params on Mac

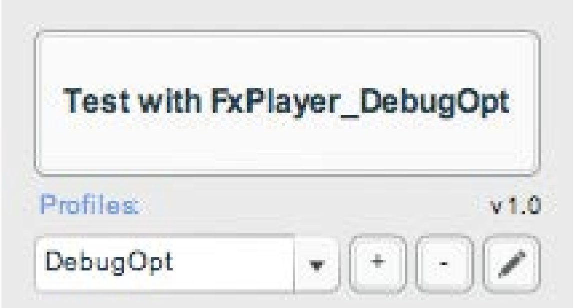
8. Press *OK*.

9. The *n*ew 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.



Profiles: AMP

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



Profiles: DebugOpt

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

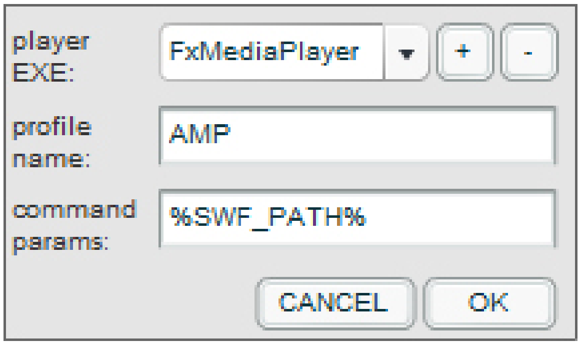
Editing Profiles

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.*

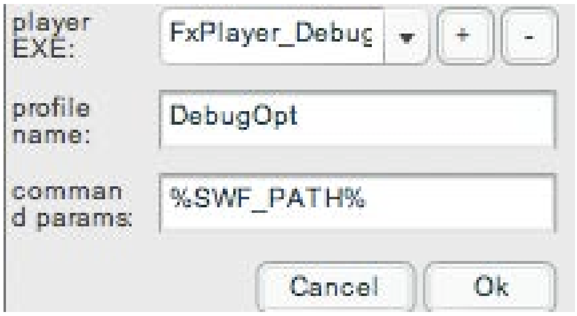


player EXE: FxMediaPlayer

profile name: AMP

command params: %SWF_PATH%

Figure 24: Edit and Delete Profiles dialogue on Windows.



player EXE: FxPlayer_Debug

profile name: DebugOpt

command params: %SWF_PATH%

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 player will be launched, and the SWF will be published and played.

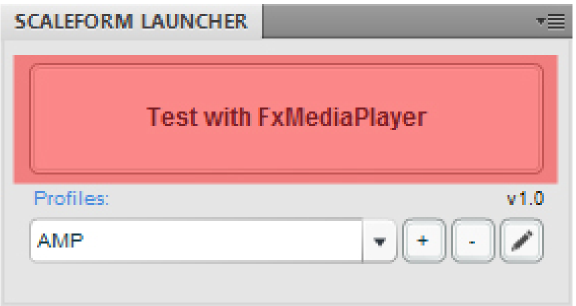


Figure 16: Press Test with GFxMediaPlayer to publish and test the SWF on Windows.

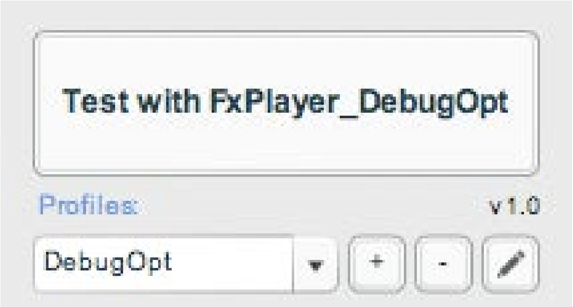


Figure 27: Press Test with GFxPlayer_DebugOpt to publish and test the SWF on Mac.

Launching the Player via Keyboard Shortcut

You can setup Flash to launch the Scaleform 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 Scaleform 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.

2.10 Licensing

2.10.1 License Problems

To run any application that is linked with Scaleform 4.3 Eval libs, please ensure the file sf_license.txt is in one of the proper directories for the desired platform. The file *sf_license.txt* is a text file that contains the key in the following format—all uppercase alpha-numeric characters with no spaces:

3AAAA1BB23B8ZCCCC4CCEOFSJCZ08DDDDDEF

If an early version of the game or application is released, the *sf_license.txt* file may be distributed in the released package. Please note that Scaleform 4.3 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 *sf_license.txt* is found in directory path two, but an expired *sf_license.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 Scaleform 4.3 application that finds the expired *sf_license.txt* file will exit, displaying the message:

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

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

> Failed to open the Scaleform license file, please make sure sf_license.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. Root of the installation 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)/sf_license.txt
   Note that the license key file should be put in the same directory where an executable file is deployed to.

“PLAYSTATION®3”(PS3™)

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

“PLAYSTATION®Vita”(PS Vita™)

1. Copy sf_license.txt to the "FxPlayer" directory located either in the root of your executable directory or the root of your file serving directory.

Wii™

1. dvddata/FxPlayer/ sf_license.txt

3DS™

1. Copy sf_license.txt file in Lib/3DS/target

iOS

1. Copy sf_license.txt to the root of the Xcode project and make sure that Xcode adds it to your target product (Scaleform.app for example).

Android

1. Copy sf_license.txt to the location “/sdcard” on the device using whatever file explorer/manager is available (note that it may default to that location and not allow access to the root of the file system).

Windows Phone

1. Copy “sf_license.txt” in to your Visual Studio project, and make that the text file is set to be added to the applications “Deploy” step.

Windows Store

1. Copy “sf_license.txt” in to your Visual Studio project, and make that the text file is set to be added to the applications “Deploy” step.

All Platforms

The license key can also be used programmatically by using the “GFx_SetEvalKey” function. This function must be called before the Scaleform Loader is initialized.

Note for PS4: In order to read the license.txt file on PlayStation 4, you should add your license key programmatically to the application code. This is done using the GFx_SetEvalKey function in Platform_PS4.cpp. The function needs to be called before the Scaleform loader is initialized.

For example, in Platform_PS4.cpp:

```
void    GFx_SetEvalKey(const char* key);

//--------------------------------------
int main(int argc, char** argv)
{
    GFx_SetEvalKey("LICENSE_KEY"); // replace LICENSE_KEY with the actual license
key string found in the license.txt file

        ...

}
```

2.11 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 https://gameware.autodesk.com/. If this is not the case, please register. Once registered, access to https://gameware.autodesk.com/scaleform/developer/ 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.

Ticketing System: Open a ticket in the Scaleform Developer Center 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 contacting 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.

2.12 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 gameware@autodesk.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 a great product.

3 What is Included?

Scaleform 4.3 comes with a variety of different documents and sample files. These documents provide information related to the core features of Scaleform 4.3, the integration of Scaleform 4.3 into a game engine, how to use Scaleform 4.3 and its various components, how Scaleform 4.3 works with Flash and AS 2.0, 3.0 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 Scaleform 4.3 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 4.3->Documentation

3.1 Learning Flash

Some documents assume a certain level of knowledge of Flash and AS2/AS3. For Scaleform 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
- Essential ActionScript 2.0 by Collin Moock
- ActionScript: The Definitive Guide by Collin Moock

3.2 Getting Started Guides

Scaleform 4.3 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 Scaleform 4.3 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 Scaleform 4.3, 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 Scaleform 4.3 and Flash. They are much more in depth than the *Getting Started* guides and are written to aid in Scaleform 4.3 integration with an application or game engine.

- *Integration Tutorial*: This document introduces basic Scaleform 4.3 usage and 3D engine integration using a DirectX 9 example application.

- *Reference Documentation*: Available online and as CHM, the Reference Documentation describes the Scaleform 4.3 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 the memory management in Scaleform 4.3.

- *Font Overview Configuration*: This document describes the font and text rendering system used in Scaleform 4.3, providing details on how to configure both the art assets and Scaleform C++ APIs for internationalization.

- *Scale9Grid*: This document describes the details of using Scale9Grid functionality in Scaleform 4.3 to create undistorted, resizable windows, panels, buttons, and other images.

- *ActionScript Extensions*: This document describes ActionScript extensions available in Scaleform 4.3.

- *IME Configuration*: This document covers how to integrate and use IME with Scaleform 4.3. It also includes information on the configuration and use of the IME Language Bar.

- *DrawText API*: This document describes the DrawText API introduced in Scaleform 4.3 for C++ text driven rendering and formatting outside of ActionScript sandbox.

- *XML Overview*: This document describes configuring XML support available in Scaleform.

- *Flash Support Overview*: This document provides the most up-to-date listing of Flash functionality supported by Scaleform 4.3.

- *Best Practices Guide:* This *document* describes the best practices for efficiently implementing and effectively leveraging Scaleform 4.3.

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

- *GFx LITE Customization Overview*: This document details the Scaleform LITE configuration libraries and describes how to customize Scaleform LITE in your own application.

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 gameware@autodesk.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: T*his document covers how to fully integrate Scaleform 4.3 into UE3. It describes how Scaleform 4.3 can be integrated with UE3, including Kismet, UnrealScript and the Unreal Editor.

- Platform README files. Some platforms and updates may include a readme file which provides additional platform-specific notes. This file will be located in the root of the SDK installation directory if it exists.

3.5 Demos and Examples

Scaleform 4.3 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 http://gameware.autodesk.com/scaleform/developer/?action=demo.

The demos include videos, interactive content, Flash files and a wide assortment of other content designed to aid licensees in learning Scaleform 4.3 quickly.

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

  Source (Multi-Platform)*: C:/Program Files (x86)/Scaleform/GFx SDK 4.3/Apps/Samples/GFxPlayerTiny*

- *Scaleform 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 4.2/Bin/Win32/Msvc90/ SWFToTexture/SWFToTexture_D3D9_Release_Static.exe*
  Source (Multi-Platform)*: C:/Program Files (x86)/Scaleform/GFx SDK 4.3/Apps/Samples/RenderTexture*

- *Scaleform 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 4.2/*Bin/Win32/Msvc90/
  TextureInSWF/TextureInSWF_D3D9_Release_Static.exe
  Source (Multi-Platform)*: C:/Program Files (x86)/Scaleform/GFx SDK 4.3/Apps/Samples/RenderTexture*

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

3.6 Online Documents and Videos

In addition to the documents and tutorials that come with the Scaleform 4.3 SDK, there are additional documents and examples available on the Scaleform website:
http://gameware.autodesk.com/scaleform/developer/?action=doc

The Scaleform developer's center also includes many video tutorials which explain various parts of the Scale SDK, including a 'Getting Started' series. Here is the link to the Video Tutorials web page:
http://gameware.autodesk.com/scaleform/developer/?action=vid

4 Where to Begin

Where and how to begin with Scaleform 4.3 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 Scaleform player, then determine what is needed to integrate Scaleform 4.3 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 Scaleform 4.3.

For more details on typical Scaleform Integration and Content development, please see the following post in our forums: Scaleform: Sample Integration and Content Development Timeline.

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 Scaleform 4.3, common sense and good testing procedures are critical. While integrating Scaleform 4.3 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://gameware.autodesk.com/developer/forums/

4.1 User-Specific Information

4.1.1 UI Programmers

Programmers must usually first worry about how to integrate Scaleform 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, Scaleform 4.3 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, CS4 or CS5 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 Scaleform 4.3 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 4.3 comes with a variety of players for Windows and each platform. The Scaleform 4.3 Player supports playing native SWF files from Flash or Scaleform's proprietary GFX file format. To play a file, simply run the Scaleform 4.3 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 4.3 SDK installation places a "Scaleform SDK Browser" shortcut on the desktop. The Scaleform Player can be found by launching the SDK Browser and choosing the 'Programs' tab on the left.
Scaleform 4.3Additional players may be accessed from the start menu: Start->Programs->Scaleform ->GFx SDK 4.3>GFx Players. These players may be located via Windows Explorer as well:

C:/Program Files/Scaleform/GFx SDK 4.3/Bin/Win32/Msvc80/GFxPlayer
C:/Program Files/Scaleform/GFx SDK 4.3/Bin/Win32/Msvc90/GFxPlayer

To test, run the Scaleform FxPlayer from the SDK Browser.This will open up a blank player window. Next, use Windows Explorer in Windows to browse to: *C:/Program Files/Scaleform/GFx SDK*

4.3/Bin/Data/AS2/Samples/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 Scaleform SDK 4.3 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 Scaleform SDK 4.3 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/Data/{AS2 or AS3}/Samples* in the Scaleform SDK 4.3 installation.

4.3 Starting a New Project

Scaleform 4.3 supports the use of AS 2.0 and 3.0. Flash CS3, CS4, and CS5 may be used to author content; however, it is important that when starting a new project that you choose either an ActionScript 2 or ActionScript 3 file type, depending upon which version you wish to use. This can be done from the initial start page of Flash, or from the top menu via File->New->Flash File (ActionScript 2.0 or ActionScript 3.0). Ensure that the Flash Publish Settings have Flash Player 10 chosen as the Flash Player type. These steps will help to ensure maximum compatibility.

It is important to note that Scaleform 4.3 support most of the commands in AS 2.0 and AS 3.0, but not all of them. It is important to reference the *sf_4.3_flash_support.pdf* file, which contains a full list of all supported AS 2.0 and 3.0 commands.