



Tapwave® Developer's Overview

Version 1.1a



Tapwave Developer's Overview

Copyright

© Copyright 2003-2004 Tapwave, Inc. All Rights Reserved. Tapwave is a registered trademark of Tapwave, Inc. The Palm logo, HotSync, Palm OS, Palm, Palm Powered, and the Palm Powered logo are registered trademarks of PalmSource, Inc., and its affiliates. X-Forge is a trademark of Fathommer, Ltd. All other brands are trademarks or registered trademarks of their respective owners.

1. Overview

Welcome to the Tapwave platform!

Tapwave provides a next-generation gaming platform that combines the best features of the Palm OS® platform with advanced gaming features, including:

- All-inclusive gaming environment with the Fathammer X-Forge™ 3D Game Engine
- 2D and 3D graphics with the Fathammer X-Forge™ Core Libraries
- High-speed, direct-access graphics API
- Digitized and synthesized sound
- Built-in collaborative networking
- Advanced input event queue model
- Large 480x320 pixel screen, with portrait or landscape mode display

The Tapwave platform is a multi-layered system. This document provides an overview of some of the important APIs provided by these layers. You must choose the most appropriate set of APIs for your application, taking into account the following tradeoffs:

- Execution speed vs. development complexity
- Increased functionality vs. code/storage size
- Portability of application code to other platforms

The following figure shows the system architecture for the Tapwave platform:

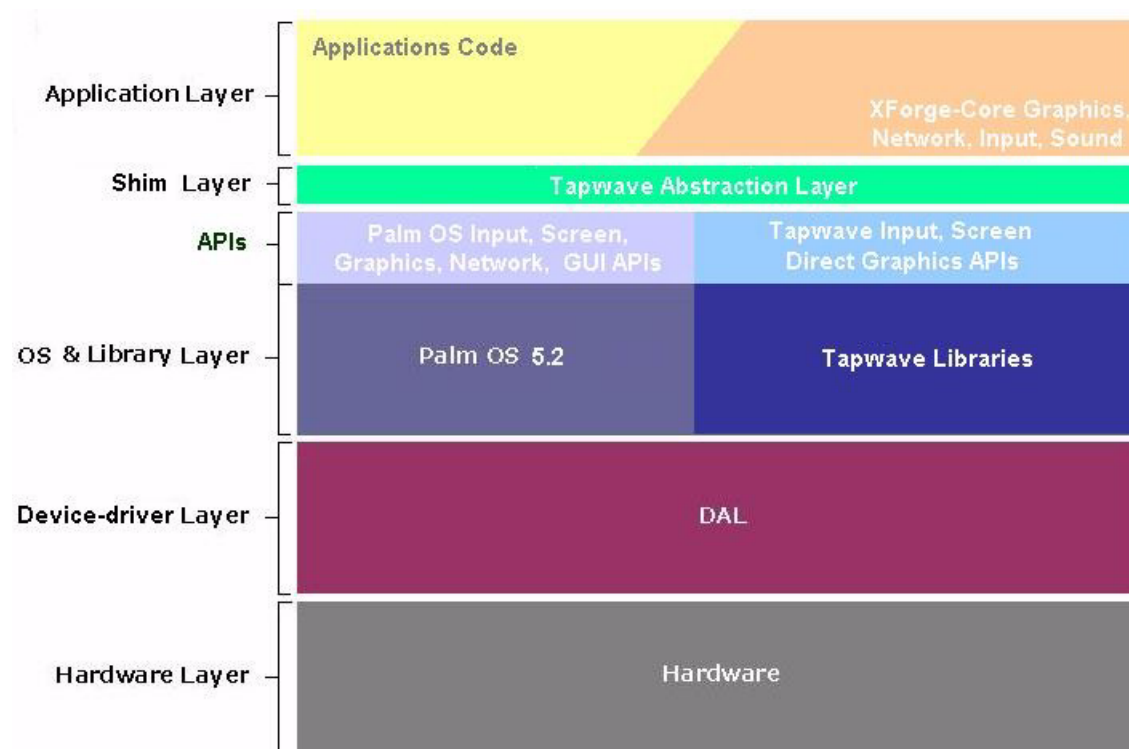


Figure 1. Tapwave Platform System Architecture

2.Fathammer X-Forge 3D Game Engine

The Fathammer X-Forge 3D Game Engine provides an all-inclusive environment that supplies most necessary application features. It is a portable framework designed to make 3D game development easier. Among other things, it provides:

- Physics simulations, including gravity and collisions
- Camera angles
- Sound
- Collaborative networking

If your application fits the X-Forge Game Engine model, you should use it. It provides a structure for your application, and it eases development by providing various built-in managers to govern physics, collisions, communications, visibility, and other aspects of your game. Of course, using the X-Forge Game Engine will increase your application's memory and storage usage.

For an overview, see ["Chapter 11, Game Engine Overview"](#) in the *X-Forge 1.0.2 Guide*.

The [X-Forge 1.0.2 Guide](#) for the X-Forge Game Engine, provides extensive and specific information on working with the X-Forge Game Engine, including chapters on 2D and 3D graphics, audio, and networking. Several examples are also included.

3. Fathammer X-Forge Core

The X-Forge Core APIs provide the underlying infrastructure for the X-Forge Game Engine, including:

- Graphics
- Audio
- File I/O
- Memory management
- Resource management
- Networking

While not providing an all-inclusive game engine, the X-Forge Core does provide an application framework, that you can use to structure an application. You should use this framework if it fits your application. If not, you can use the Palm OS model, along with additional features provided by the X-Forge Core libraries.

4. Graphics

The Tapwave platform includes several graphics APIs. These API's allow the developer to make tradeoffs between ease of development, performance, and cross-device portability. All of the API's described here will be portable between Tapwave devices.

The X-Forge Game Engine provides a general framework for game development that frees you to focus on your algorithms and the differentiating factors that make your game unique. Its graphics code is designed with portability in mind, and it provides good graphics performance for most applications.

For some extremely demanding applications, the X-Forge Game Engine may not provide the best possible graphics performance. If you decide not to utilize the X-Forge Game Engine, you may want to leverage the extensive functionality provided by the X-Forge Core libraries. The X-Forge Core APIs provide more direct control over display graphics, and offer very good graphics performance.

For developers who prefer not to use the X-Forge libraries, Tapwave provides a set of advanced TwGfx APIs. The TwGfx APIs provide easy access to graphics operations which are directly accelerated by the graphics hardware and that are still portable and compatible with future Tapwave devices. See the [Tapwave TwGfx Graphics API Reference](#) for more information.

Graphics State Note: We do not recommend interspersing calls to X-Forge Core graphics APIs with calls to other graphics API's. The X-Forge Core graphics routines may save some state internally. This could result in a misdrawn or garbled display.

4.1.2D Graphics

2D graphics support is provided by the following APIs:

- X-Forge Core 2D API
- Palm OS Window Manager
- Tapwave Graphics (TwGfx) API

4.1.1.X-Forge Core 2D API

The X-Forge Core 2D graphics API provides a portable, thin wrapper around the raw pixel data. It supports:

- Blitting with scaling
- Clipping
- Blending
- Rectangle fill
- Line drawing

For more information, see ["Chapter 8, 2D Graphics"](#) in the *X-Forge 1.0.2 Guide*.

4.1.2.Palm OS Window Manager

For simpler graphics applications, such as a board game, the Palm OS APIs provide a window management system, including clipping and point transformation.

For more information, see ["Drawing on the Palm Powered Handheld"](#) in the *Palm OS Programmer's Companion - Volume I*.

4.1.3.Tapwave Graphics (TwGfx) API

The TwGfx API provides the maximum graphics performance for 2D rendering on the Tapwave platform. This API exposes the graphics accelerator and video memory present in the Tapwave device in a forward compatible fashion. Key features include hardware accelerated "blt"s, line rendering, rectangle filling, alpha blending and transparency. The video memory can be easily used to perform high performance scene composition, and "flicker free" video updates. Note that Tapwave graphics API calls and Palm OS graphics calls can be safely interspersed.

4.2.3D Graphics

3D graphics on the Tapwave platform are provided through the X-Forge Core graphics library.

For more information, see ["Chapter 9, 3D Graphics"](#) in the *X-Forge 1.0.2 Guide*.

5.Sound

As is the case with graphics, you have several sound APIs available, including:

- X-Forge Core Audio API, for sound effects and digitized audio
- Tapwave Advanced Sound API, for volume and mute control
- Palm OS Sound API, for simple sounds and digitized audio

5.1.X-Forge Core Audio APIs

The X-Forge Core Audio APIs were written to make firing one-off sound effects easy, while at the same time supporting more advanced, dynamic control of playback. These APIs provide support for digitized sound, special effects such as Doppler shift, and playback of RIFF Wave files (.wav) and extended Module (.XM) audio sources.

Because digitized sound is more realistic than synthesized sound, it consumes more memory and storage space, and can take longer to play out.

For a platform-independent overview, see [“Chapter 3, Platform Independent - Audio”](#) in the *X-Forge 1.0.2 Guide*.

For an overview of the audio capabilities of the X-Forge Core libraries, see [“Chapter 5, Overview - Audio”](#) in the *X-Forge 1.0.2 Guide*.

For specific source code examples, see [“Chapter 10, Audio”](#) in the *X-Forge 1.0.2 Guide*.

5.2.Palm OS Sound APIs

Palm OS Sound APIs support simple sounds such as system beeps and alarms. If you are already using these simple sound routines, Palm OS also supports MIDI digitized sound through ADPCM-sampling.

For more information, see [“Chapter 10, Palm System Support - Sound”](#) in the *Palm OS Programmer's Companion - Volume I*.

For additional overview information, see [“Advanced Sound Support”](#) in the *Tapwave Programmer's Reference*.

For API descriptions, see the [Tapwave Advanced Sound API Reference](#) document.

6. Networking

Networking features are provided by the:

- X-Forge Game Engine
- X-Forge Core libraries
- Palm OS

Easy collaborative networking is built into the Tapwave platform, simplifying development of multiplayer games. Transport-independence means you only need to code to a single API to enable shared gaming across the room via Bluetooth, or across the Internet via TCP/IP!

For additional overview information specific to the Tapwave platform, see [“Bluetooth Collaborative Networking Support”](#) in the *Tapwave Programmer's Reference*.

For API descriptions, see [“Part IV, Libraries”](#) in the *Palm OS Programmer's API Reference*.

6.1. X-Forge Game Engine

The X-Forge Game Engine provides a robust, high-level network gaming model. Individual game objects on a device can directly interact with game objects on other devices. It provides a multiplayer game model, where the game state between the various devices participating in the game is automatically synchronized, and also includes a method for authenticating and keeping track of the users who are allowed to take part in the game.

Additionally, the X-Forge Game Engine provides transport-independence and portability. This means that your application can run across the Internet or a Bluetooth PAN, without any additional coding, and can be ported to other platforms that support the Fathammer libraries.

For more information, see [“Chapter 20, Multiplayer Games”](#) in the *X-Forge 1.0.2 Guide*.

6.2. X-Forge Core

The X-Forge Core provides transport-independent networking, using a sockets API. The underlying transport, currently Bluetooth and Internet, is abstracted away.

Four types of packet communications are provided:

- **Guaranteed** - provides automatic retransmission and sequencing
- **Non-guaranteed** - datagram, send-once model
- **Quick-guaranteed** - provides prioritized packets

Tapwave Developer's Overview

- Recent-state - send at pre-defined intervals

You should use this API if you want to leverage system-provided authentication, and use a socket-oriented machine-to-machine model.

For more information, see ["Chapter 5, Overview - Network"](#) in the *X-Forge 1.0.2 Guide*.

6.3.Palm OS

Palm OS contains extensive support for both IP networking and Bluetooth. It provides access to the various protocol layers of the Bluetooth stack, including:

- L2CAP - packet-oriented
- RFCOMM - data-stream/serial-oriented
- OBEX - object-level transfer-oriented

The Palm OS APIs allow for direct control over the stack, including:

- Stack initialization
- Discovering other Bluetooth devices
- Establishing device-to-device links
- Device authentication
- Control of Piconets (8 Bluetooth devices communicating together)

Palm OS also provides access to the IP protocols, TCP and UDP, through a sockets-oriented API. It includes requisite utility functions such as *gethostbyname*.

Medium-level constructs such as Object Exchange and Serial Port Emulation are also provided. These can be run over phone/modem, IR, or Bluetooth, to a phone or an Internet Access point.

These routines have no knowledge of gaming objects and are lower-level than those provided by the X-Forge Core API. Using these routines requires a great deal of work, as well as understanding the underlying network protocols. The potential for higher performance must be balanced against the additional development time, code complexity, maintenance, and reduction of portability. If you have a pre-existing implementation of a mechanism to serialize/deserialize gaming objects across a network, you may choose to port your existing code using these routines.

For comprehensive information on Bluetooth, see ["Chapter 6, Bluetooth"](#) in the *Palm OS Programmer's Companion, Volume II, Communications*.

For full information on TCP/IP networking, as well code samples, see ["Chapter 7, Network Communication"](#) in the *Palm OS Programmer's Companion, Volume II, Communications*.

7.Input

For a thorough description of the Tapwave platform's innovative input event model, see ["Advanced Input Support"](#) in the *Tapwave Programmer's Reference*.

For API descriptions, see ["Tapwave Input API"](#) in the *Tapwave Programmer's Reference*.

7.1.X-Forge Core

For information on input and controls, see the following sections in the *X-Forge 1.0.2 Guide*:

- ["Chapter 3, Platform Independent - Controls"](#)
- ["Chapter 5, Overview - Controls"](#)
- ["Chapter 6, Application Framework - Controller Input Class"](#)

7.2.Palm OS User Input

The Palm OS user input routines can be used for emulation of a simple, button-oriented game. For this type of application, the navigator can be treated as a four-way or eight-way D Pad (see [Tap_](#) later in this document).

Palm OS user input routines are based on an event queue model. A *GetNextEvent* loop is used to respond to user input events.

Coding with the Palm OS user input routines provides the best portability to simpler Palm OS-based devices, such as those that do not have navigators. Slower input, limited by event loop processing, as well as diminished accuracy, is the price paid for portability in this case.

For a general description of the user interface support in Palm OS, see ["Chapter 4, User Interface"](#) in the *Palm OS Programmer's Companion - Volume I*.

7.3.Tapwave Platform Navigator

Flexible navigator support is built into the Tapwave platform. The navigator can be treated as either a simple 4- or 8-way button cluster or an analog device.

For finer control and faster interactivity, you can read the navigator as an analog device. This is a polling model, in which you check the navigator state as often as required by your application. This provides the fastest possible user feedback.

When used as an analog device, the navigator position takes into account the display orientation (landscape versus portrait). For more information, see ["Tapwave Screen APIs"](#) in the *Tapwave Programmer's Reference*.

8. Gaming Support

Tapwave has developed a *High Score API*, which gives you an easy way to manage your user's high scores. This is a special feature of Tapwave devices that enables you to build community between your game users by sponsoring high score contests, etc... A *High Score Conduit*, included with the Tapwave desktop, uploads these scores to Tapwave.com. It is important that you use this High Score API if you want your users to be able to participate in this service of the Tapwave website.

For more information, see ["Gaming Support"](#) in the *Tapwave Programmer's Reference*.

9. Tapwave Platform API Cross-Reference

Tapwave has developed several API sets that take advantage of the unique features of Tapwave devices. The table below is a helpful cross-reference, associating desired game features with corresponding sections in the *Software Development Kit* documentation.

Feature	Document	Section(s)
Screen size and layout	<i>Tapwave Programmer's Reference</i>	"Large Screen and Landscape Support" "Tapwave Screen API"
Input and event queue management	<i>Tapwave Programmer's Reference</i>	"Advanced Input Support" "Tapwave Input API"
High Scores	<i>Tapwave Programmer's Reference</i>	"Gaming Support" "Gaming APIs"
Rumbler	<i>Tapwave Programmer's Reference</i>	"Vibration Support" "Device API"
Graffiti®/Pen Input Area	<i>Tapwave Programmer's Reference</i>	"Tapwave Screen API"
Graphics Accelerator	Tapwave TwGfx Graphics API Reference	Entire document

Tapwave Developer's Overview

Volume and Mute Control	<u>Tapwave Advance Sound API Reference</u>	Entire document
-------------------------	--	-----------------