CALGARY / SCIENTIFIC

PureWeb® STK 4.0

Quick Start Guide: iOS Client

About Calgary Scientific

Calgary Scientific Inc. is dedicated to providing advanced visualization, web enablement, and mobility enhancement solutions to industries looking for secure access and use of their data or graphics-intensive applications while using their existing systems. Visit www.calgaryscientific.com for more information.

Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Calgary Scientific Inc. cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System's Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Calgary Scientific products.

Trademarks

Released by

Calgary Scientific Inc. www.calgaryscientific.com.

Document Version: PW4.0_iOSClient_QSG_07-2013_v1.000.00

Table of Contents

Preface

Welcome to the *PureWeb STK 4.0 Quick Start Guide: iOS Client*, a document that provides instructions to install the PureWeb STK, and to build, run and troubleshoot sample PureWeb iOS client applications.

Intended Audience

This document is primarily intended for software developers who plan to install and use the STK to PureWeb enable an iOS client.

Making Comments on This Document

If you especially like or dislike anything about this document, feel free to e-mail your comments to techpubs@calgaryscientific.com.

Contacting Calgary Scientific Support

Use one of the methods in the table below to contact Calgary Scientific support.

Web Site	E-Mail
support.getpureweb.com	support@getpureweb.com

Chapter

1　Introduction

PureWeb is a platform that enables the rapid transformation of enterprise software into cloud-ready, web and mobile applications.

The PureWeb STK (software transformation kit) is a set of tools that provides developers everything they need to implement the PureWeb solution in their own applications, in particular service and client APIs, documentation and API reference material, and sample applications in each supported programming language.

The sample applications illustrate key concepts of PureWeb enablement; they can be used as a starting point for developers to create their own PureWeb-enabled applications.

The PureWeb *Quick Start Guide* series is intended to get developers up and running as quickly as possible, using these sample applications as models.

As its name indicates, the current document, *PureWeb STK Quick Start Guide: iOS Client*, focuses on the iOS client. The guide provides instructions to install the PureWeb STK, and to build, run and troubleshoot a sample client.

Two iOS sample clients are available:

* Scribble: a simple canvas that allows users to draw lines, change line color, and erase lines.

* Asteroids: a PureWeb-enabled implementation of the classic video game with a twist -- it supports both single-player and two-player modes; each player independently controls his or her own ship.

Before working with the sample iOS client, developers must build a sample service as described in the *Quick Start Guide* for the service development platform.

Recommended Reading

This *Quick Start Guide* is intended as an introduction and does not provide in-depth information; it should be used in conjunction with the other guides of the PureWeb STK documentation suite, in particular:

• *Installation Guide*

• *Developer's Guide*

• *Server Administration Guide*

• ObjectiveC API reference material

The first three references are installed with the server.

On a Mac operating system, to access the ObjectiveC reference material within Xcode:

• Go to **Window | Organizer**.

• Click the **Documentation** icon.

2 Installing PureWeb

This chapter provides prerequisite information as well as instructions to install the PureWeb STK, and build the sample iOS clients.

Prerequisites

This document assumes that you have already installed the PureWeb server on either Linux or Windows, as described in the *PureWeb Installation Guide*.

To install the iOS client STK on a Mac, you will need:

- Mac OS 10.7 (Lion) or OS 10.8 (Mountain Lion)
- Xcode 4.6

To build and run the iOS sample client applications, you will need:

- iOS 5 or 6
- iOS SDK 5.0 or higher, including the Xcode IDE, iOS Simulator and the suite of tools for Apple iPhone/iPad development. You will be required to register as a developer with Apple.

We also recommend that you review the Apple documentation, videos and sample code for getting started with Apple iPhone/iPad development at: developer.apple.com/devcenter/ios/index.action

Installing the STK

The PureWeb STK components on a Mac are packaged in a .dmg file.

Note: Xcode must be installed first before installing the PureWeb STK.

1. Open the .dmg file to display its content. You will see the license agreement (EULA), the *Installation Guide*, this *Quick Start Guide*, and the installer (.pkg file).
2. Double-click on the `PureWebSetup-[version].pkg` file and follow the steps in the installation wizard.
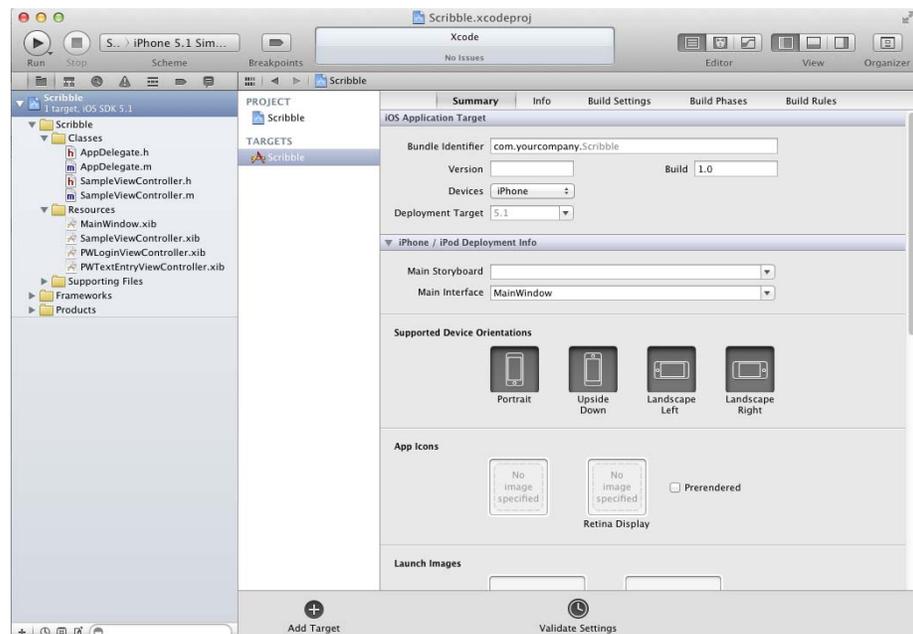
Building the Applications

The section builds both the Scribble and Asteroids sample applications. This is a three-step process which involves creating a project in Xcode, running the build script, and configuring the application on the device or simulator.

Note: You must build the service application before following the procedures in this section to build the iOS client on a Mac.

For instructions, refer to the *Quick Start Guide* for the service that you want to build.

Creating a Project in XCode

1. Open Xcode and select **Create a new Xcode project** from the Welcome dialog.

2. Click on the **PureWeb STK** section to display the sample templates; there is one for Scribble and one for Asteroids.

3. Select the sample template of your choice and click **Next**.

4. Type a name for your project in the **Product Name** field, and type your company name in the **Company Identifier** field. Click **Next**.

5. Choose a location to store your project. We recommend that you check the Source Control option.

6. Click **Create** to automatically create the project with all the appropriate source and header files.
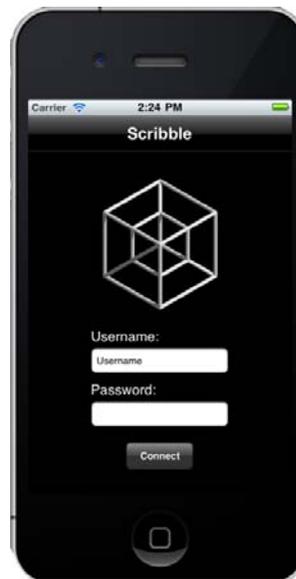
Running the Build Script

Click the **Run** arrow button in the top left corner to build the client code. Once the application has been built, it will be automatically installed into the Apple iPhone simulator and begin to run.

Configuring the Application on the Device

When the application is first running, it displays a login page, as illustrated below.

In order for this login to work, you must first configure the application on the simulator or device to work with your server and credentials.

1. Close the application and navigate to the device's main page (in the simulator, select **Hardware | Home**).

2. Tap the **Settings** icon.

3. Tap the application you built.

4. Modify the settings as required. In particular, ensure that the server URL is accurate.

Note: Set the username and password to eliminate having to enter the information each time that you start the application. The default is `admin` for both fields.

5. Close the Settings window to return to the device's main menu (in the simulator, select **Hardware | Home**).

You are now ready to log in and start using the sample application. The next chapter provides an overview of the functionality available in Scribble and in Asteroids.

Chapter

3 Application Overview

This chapter briefly describes the functionality available in each of the two sample applications, Scribble and Asteroids.

To launch the application in the simulator:

1. Start the application by tapping its icon.

2. Log in, using `admin` in both the username and password fields. If you have configured the device's setting as described in the previous chapter, these fields will already be filled out.

3. Tap the **Connect** button.

Scribble

Scribble is a simple canvas that users can use to draw lines, change line color and erase lines. You can also share the application with another user.

When you first launch Scribble, it displays a blank canvas. You can draw lines on the canvas by clicking the mouse and dragging (if using a simulator), or using touch gestures (if using an actual device).

Scribble also offers the following functionality:

- Erasing lines: tap the **Clear** button in the lower left-hand corner to erase the canvas.
- Changing pen color:
  a. Tap the **Color** button in the lower left-hand corner to display the text entry view.
  b. Type a new color in the **Color** field; if you type an unrecognized color, the color of the scribbles will remain the same.
  c. If needed, hide the virtual keyboard and scroll up to display the Done button in the top right-hand corner.
  d. Tap **Done** to return to the Scribble canvas view.

4. Viewing the Diagnostics Panel. See "Using the Diagnostics Panel" on .
5. Sharing the application with another user. See "Sharing the Application" on .
6. Closing the application: tap the **Disconnect** button. Note that the service application also closes.

Asteroids

Asteroids is a PureWeb-enabled implementation of the classic video game with a twist -- it supports both single-player and two-player modes; each player independently controls his or her own ship.

Single-Player Mode

- Press any key to start the game.
- Use the up/down arrow keys on the keyboard (or buttons on the device) to move forward/reverse.
- Use the left/right arrow keys (or buttons on the device) to change heading.
- Press the red button icon to fire torpedoes.
- Activate shields by pressing the shield icon. Each player gets 5 seconds of shield time per game. Note that shields are also enabled in two-player mode.
- View the Diagnostics Panel. See "Using the Diagnostics Panel" on .
- Share the application. See "Sharing the Application" on .
- Close the application by tapping the **Disconnect** button. Note that the service application also closes.

Two-Player Mode

Once the application is shared, additional two-player mode functionality includes:

- Collide with an asteroid with your shields up to destroy the asteroid and collect the points.
- Collide with an unshielded opponent with your shields up to destroy their ship and collect all of their points.
- Collide with a shielded opponent with your shields up to create an elastic collision.

Using the Diagnostics Panel

The Diagnostics Panel is described in detail in the Debugging chapter of the *PureWeb STK Developer Guide*, however below is a brief overview of the functionality it offers.

To access the panel, tap the **Diagnostics** button in the top right-hand corner. This will display the content of the Trace feature, which is used to examine messages to and from the server.

The panel also offers the following other two features:

- Options: tap the **Options** icon at the bottom of the screen to access this feature. The parameters in this view impact image bandwidth and quality.
- Application State: tap the **AppState** icon to displays an XML representation of the application's state manager.

When you are done with the Diagnostics Panel, tap the **Back** button in the top left-hand corner to return to the application's main view.

Sharing the Application

Both the Scribble and Asteroids sample contain functionality to illustrate collaboration. This section describes how one user can share the application with a second user so that they are both logged in the same session.

A few important points to note before getting started:

- By default, the sample application connects using the hostname of the local host. If you are trying to share the application from another computer, you must modify the URL, `http://localhost:8080/pureweb/...` by replacing `localhost` with the IP address or DNS name of the server.
- Cookie sharing between browser tabs may cause issues with the functionality of the sample application. We recommend that you use two separate browser instances if you want to open two sessions of the same application. Google Chrome users can also use and incognito window.

First User:

1. Tap the **Share** button to open an email application.

2. Send a second person the URL displayed in the body of the email.

Note: The email functionality is not included in the Apple iPhone Simulator. To share the application, copy and paste the share URL into a browser.

Second User:

1. Open the received message and tap the URL to display the PureWeb collaboration login page in your browser.

2. Select your preferred client from the drop down list. All possible clients are listed, but only the clients that have actually been built will work.

3. Enter `Scientific` as your password and tap the **Sign In** button to join the Scribble client session.

4   Exploring the Code

One of the main purposes of including sample applications with the STK is for developers who are new to PureWeb to use these samples as a starting point for coding and to experiment with them.

Therefore, don't be shy about editing the code and trying new things. That's the best way to learn, and you'll see how quickly you'll become proficient with PureWeb.

This chapter gives some pointers about the code in the sample iOS client; however, it assumes that you are familiar PureWeb concepts such as views, state, commands and event handlers, which are described in the *Developer's Guide*.

For an overview of the service-side code, please refer to the *Quick Start Guide* for the service development platform you want to work with.

The pointers below describe items of interest in the code from a PureWeb enablement perspective.

AppDelegate.h File

Notice how the `AppDelegate` class is updated to conform to the `PWFrameworkDelegate` and `PWWebClientDelegate`. The `PureWeb.h` header is included and the two protocol methods are added to the `AppDelegate` class.

AppDelegate.m File

There are a few important methods to note:

- `PWWebClientDelegate sessionStateChanged`, which is used to notify of session state changes like connected, disconnected, errors, etc.
- `PWFrameworkDelegate stateInitialized`, which is called after successfully connecting and initializing application state. At this point, it is possible to write to `AppState`.
- `application:didFinishLaunchingWithOptions`, which sets the delegates on the `PWFramework` and `PWWebClient`.
- `PWWebClient disconnectSynchronous`, which is used to guarantee disconnection when the application terminates.

PWLoginViewController.m File

Note the following.

• Basic authentication is set on the client in order to connect to the PureWeb server.

• The `PWFramework` connect method is asynchronous and the callback is `PWWebClientDelegate sessionStateChanged`.

SampleViewController.m File

This file illustrates several key concepts.

Displaying Remoted Views

Navigate to the `(void)loadView` section to see how a view on the client is connected to the remote view by setting the name of the view.

Sending Input

If you are not using a command-driven design, you will need to send input events. For simplicity, the PureWeb views have helper methods for queuing input events.

In Scribble, see the following section for an example:

`(void)scribbleGesture:(UIPanGestureRecognizer *)gesture`

In Asteroids, see the following section for an example; each button sends a key event to the service:

`(void)accelerateBegan`

Sending Commands

Also, if you have commands registered in your service, you can queue your command through the client.

See the following section in Scribble for an example:

`(IBAction)clearButtonPushed:(id)sender`

Application State Handlers

There are two types of application state handlers available.

• Value changed handlers fire only when a specific value changes.

• Child changed handlers fire whenever it's children are updated.

The Scribble sample monitors the change in line color using a value changed handler.

See the following section for an example:

```
[PWFramework sharedInstance].state.stateManager
addValueChangedHandler
```

5 Debugging

Using the sample application as models, this section describes how to debug a PureWeb iOS client in Xcode. The information in this section is based on Xcode 4.6; it may vary slightly if you are using a different version.

For instructions on how to debug the service, please refer to the applicable PureWeb service *Quick Start Guide*. Additional debugging features, in particular the Diagnostics Panel, are described in detail in the *Developer's Guide*.

Procedure

1. Open Xcode and select your project from the Welcome dialog.

2. Open the `AppDelegate.m` implementation file in the Classes folder and navigate to the `didFinishLaunchingWithOptions` function.
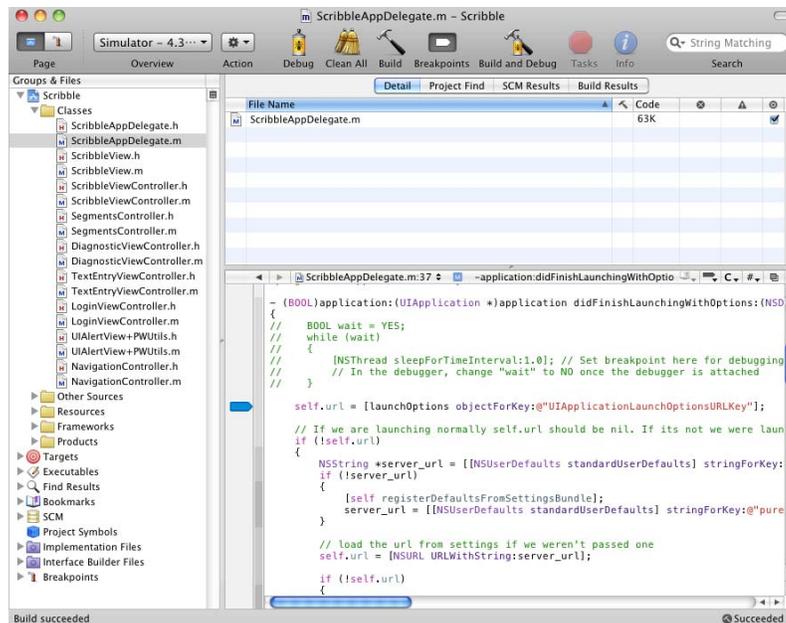
3. Click on the side panel beside the editor window to set a breakpoint at the desired location.

4. Click the **Run** icon in the top left-hand corner to run the client code and trigger the breakpoint.