

GETTING STARTED WITH WINDOWS AND MAC DEVELOPMENT

Introduction to Windows and Mac Development and Setting up the Development Environment

E-Learning Series Course Book – Lesson 1

Embarcadero Technologies

© Copyright 2012 Embarcadero Technologies, Inc. All Rights Reserved.

Americas Headquarters
100 California Street, 12th Floor
San Francisco, California 94111

EMEA Headquarters
York House
18 York Road
Maidenhead, Berkshire
SL6 1SF, United Kingdom

Asia-Pacific Headquarters
L7. 313 La Trobe Street
Melbourne VIC 3000
Australia

Lesson 1 – Introduction to Windows and Mac Development and Setting up the Development Environment

Version: 1.2

Presented: May 3, 2012

Last Updated: May 23, 2012

Prepared by: David Intersimone “David I”, Embarcadero Technologies

© Copyright 2012 Embarcadero Technologies, Inc. All Rights Reserved.

davidi@embarcadero.com

<http://blogs.embarcadero.com/davidi/>

Contents

Lesson 1 – Introduction to Windows and Mac Development and Setting up the Development Environment	2
Introduction	3
FireMonkey Features	3
What is the difference between FireMonkey and RIA platforms like HTML5/CSS, Adobe Flash or Silverlight?.....	6
System Requirements, Install and Setup	8
System Requirements	8
Installation	9
Installing RAD Studio, Delphi or C++Builder	9
Installing InterBase XE Developer Edition.....	18
Installing the Platform Assistant(s)	19
Installing the FireMonkey tools for iOS.....	29
Setting up RAD Studio to support development of iOS projects	30
Summary, Looking Forward, To Do Items, Resources, Q&A and the Quiz	33
To Do Items	34
Links to Additional Resources	34
Q&A:.....	34
Self Check Quiz.....	40
Answers to the Self Check Quiz:	41

Introduction

Welcome to FireMonkey, the first native CPU- and GPU-powered application platform for building rich business applications for Windows (Delphi/C++), Mac (Delphi/C++) and Mobile (Delphi only at this time).



This first lesson takes you through an introduction to the FireMonkey business application platform, installing the RAD Studio XE2 software and setting up your development environment. This lesson will also give you a quick look at building your first FireMonkey application.

FireMonkey Features

Before we start, let's take a look at the FireMonkey Platform Features Overview and learn about the major benefits of using FireMonkey for your Windows and Mac application development -

<http://www.embarcadero.com/products/firemonkey>.

Imagine a next generation platform that helps you build visually spectacular business applications that are faster, more visually stunning, and more connected. Introducing FireMonkey, the first native CPU and GPU powered platform for rich business applications. What you can build in FireMonkey is only limited by your imagination.

FireMonkey Business Application Development Platform



FireMonkey makes it easy for developers to rapidly build visually engaging and data rich HD and 3D applications with blazing native code performance on Windows, Mac and iOS. You can create FireMonkey applications that use a single source code base to target the Mac OS X, Win32, Win64 and iOS (Delphi only in XE2) platforms. FireMonkey uses lightweight vector graphics based GUI controls on top of a cross-platform abstraction, which is implemented for Windows, Mac OS X, and iOS. To render HD and 3D user interfaces, FireMonkey uses:

- Windows: Microsoft DirectX version 9 or later (for HD and 3D)
- Mac: Apple CoreGraphics Quartz 2D (for HD) and OpenGL (for 3D)
- iOS: Apple CoreGraphics Quartz 2D (for HD) and OpenGL/ES (for 3D)

FireMonkey Graphics Architecture



Lightweight controls mean that every pixel is drawn by FireMonkey; no operating specific (heavyweight) controls are used. Being native not only means fast, it also means that applications are compiled into simple and efficient executable files that are easy to distribute and deploy to Windows, Mac OS X or iOS devices.

- **More Visually Stunning and Interactive** - Finally, business applications can get Hollywood level presentation. Easily create rich HD and 3D applications that communicate your business ideas with higher visual fidelity and deeper user involvement. The FireMonkey GPU powered user experience includes scalable vectors graphics, UI components, 3D processing and powerful image effects with CPU powered business logic, animation and database access all working together to help you rapidly create business applications that are more immersive, interactive and appealing. FireMonkey vector based graphics gives application developers all the power of complex graphics capabilities with all the ease of component based, Rapid Application Development. With GPU based, resolution independent, scalable vectors, alpha blending and gradients, you get razor sharp & lightning fast graphics, smooth animation and maximum resolution on every device. You can ensure that your applications always look their very best, no matter what resolution the target machine can run at. FireMonkey also provides an incredibly flexible and powerful way to use styles, layouts and controls. With extremely flexible, powerful and simple control parenting and tooling any part of your application can be customized and high levels of reuse achieved. <http://www.embarcadero.com/products/firemonkey/more-visually-stunning-and-interactive>.
- **More Power and Performance** - No matter how dazzling the visuals are, if the user experience is sluggish or unresponsive it all falls down. So FireMonkey applications are backed by the full power of today's hardware to drive rich business applications with native CPU performance and

E-Learning Series: Getting Started with Windows and Mac Development

GPU powered visuals on PC, Mac, and Mobile devices. FireMonkey effects and transformations are all handled directly in the GPU, giving optimal performance and impact. Advanced animations use the GPU with minimal CPU usage and automatic frame rate correction. With FireMonkey you get the performance and responsiveness that you simply can't get from RIA platforms, virtual machine runtimes, or dynamic languages.

<http://www.embarcadero.com/products/firemonkey/power-and-performance>.

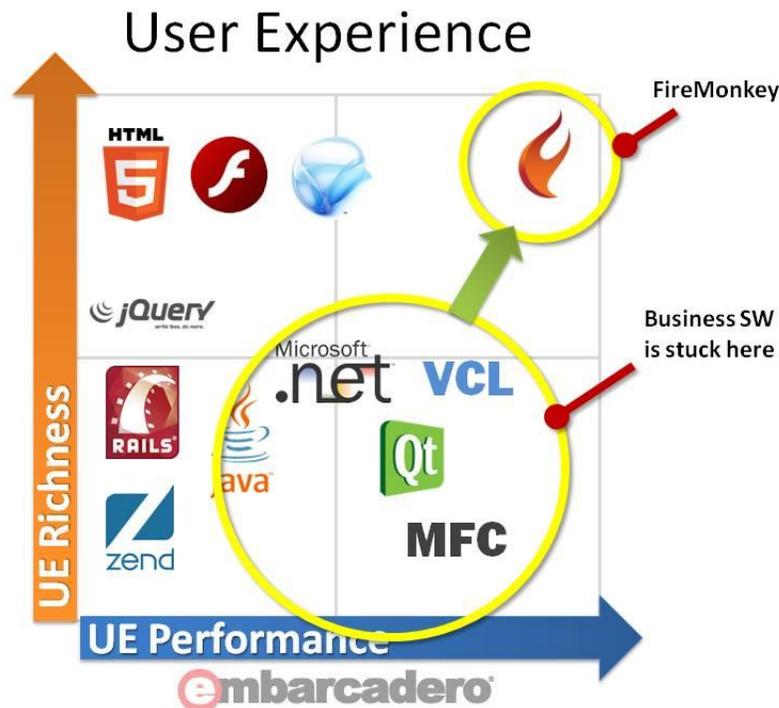
- **More Connected** - What your application does is equally important as what it does it with. FireMonkey applications easily connect to the most popular databases, Web and cloud services, internet protocols and mobile devices – giving you incredible business flexibility for both clients and servers. FireMonkey includes built-in component based access to MS SQL Server, Oracle, IBM DB2, MySQL, Sybase, Informix, InterBase and Firebird databases. And of course, FireMonkey's data access is native which, once again, means high performance. Use LiveBindings to visualize data in your FireMonkey application. LiveBindings offer expression-based binding that enables you to bind objects either to each other, or to any number of data sources. <http://www.embarcadero.com/products/firemonkey/more-connected>.
- **More Agile** - FireMonkey includes a full suite of reusable components for HD and 3D user interfaces, data access, effects, animation and more. Component and property based development makes it simple and intuitive to work with all kinds of graphical elements and resources. As well as to create your own components for you, your team or your customers to use. FireMonkey's RAD development approach allows for fast visual prototyping using customizable components, giving you the freedom to take an idea and quickly show your customer exactly what you are thinking. But unlike prototyping tools, FireMonkey prototypes are working applications that you can take from prototype all the way to production. <http://www.embarcadero.com/products/firemonkey/more-agile>.

What is the difference between FireMonkey and RIA platforms like HTML5/CSS, Adobe Flash or Silverlight?

RIA platforms have rich free-form graphics, time based animation, and media capabilities that are well suited for web content, entertainment and advertising. Because RIA applications are typically web embedded, they use scripting and VM engines to execute program code. This provides cross-platform browser compatibility, but comes with a performance trade off. RIA platforms being entertainment focused are not designed around the needs for business applications. Things like enterprise data and services connectivity, business UIs and graphics, and packaged and enterprise app distribution are not well served with RIA platforms.



FireMonkey is a native application platform, with many of the rich capabilities of RIA platforms such as vector graphics, 3D, animation, effect, and cross-platform but FireMonkey is focused on the unique needs of business, ISV, and industrial use software. FireMonkey is fully CPU and GPU native, UI oriented, and provides powerful database and services connectivity.



System Requirements, Install and Setup

To develop FireMonkey applications, you can choose to use RAD Studio XE2, Delphi XE2, or C++Builder XE2 as your development environment (we'll refer to these products as RAD Studio). RAD Studio runs under Microsoft Windows. If you don't already have RAD Studio, you can download a 30-day trial version at <http://www.embarcadero.com/products/firemonkey/downloads>.

System Requirements

The following Windows platforms are supported for development:

- Microsoft® Windows 7 (32-bit and 64-bit)
- Microsoft Windows Vista™ SP2 (32-bit and 64-bit) requires administrator rights
- Microsoft Windows XP Home or Professional (32-bit and 64-bit) SP2 or SP3
- Microsoft Windows Server 2003 (SP1) or 2008 (32-bit and 64-bit)

For developing 64-bit Windows applications (using Delphi or RAD Studio only) you will need either a PC running a 64-bit version of Windows or a 32-bit development PC connected to a PC running a 64-bit version of Windows.

For developing OS X applications (using Delphi, C++Builder, or RAD Studio) you will need a PC running Windows connected to an Intel-based Mac with 2 GB RAM or more, running OS X 10.6 (Snow Leopard) or OS X 10.7 (Lion) over a local area network using an SSH, VNC or Windows file-sharing solution.

E-Learning Series: Getting Started with Windows and Mac Development

For developing iOS applications (using Delphi XE2 or RAD Studio XE2) you will need Apple's iOS SDK 4.2 or later (iOS SDK 5.1 recommended), Xcode 3.2.5 or later (if you are using Mac OS X 10.7.x Lion, you will need to use Xcode version 4.2.3 or later). If you are using Xcode 4.3 or later, Apple now bundles include and header files in the Xcode app bundle. For C++ developers who want build Mac applications, this means you need to also install the Xcode command line tools to install the "/usr/include" and "/usr/lib" directories and files (see notes about creating remote profiles). You will also need Free Pascal version 2.4.4 or later using the installer provided with the XE2 release.

FireMonkey applications run on multiple platforms with a basic Graphics Processing Unit (GPU). For Windows development, any vendor DirectX 9.0 class or better (Pixel Shader Level 2). For OSX and iOS development, OpenGL (note: qualified Macs all have a qualified GPU).

Additional system requirements are listed at the bottom of the RAD Studio data sheet - <http://www.embarcadero.com/products/rad-studio/rad-studio-data-sheet.pdf>

Installation

There are several steps to installing RAD Studio for use in building FireMonkey applications for Windows, Mac and iOS:

- Installing RAD Studio, Delphi or C++Builder
- Installing the Platform Assistant(s)
- Installing the FireMonkey tools for iOS (Delphi only).

Let's go through each of them to make sure you have all of the right pieces and parts installed to get started building FireMonkey applications during this course.

Installing RAD Studio, Delphi or C++Builder

Depending on whether you have purchased RAD Studio or you are going to use the trial edition, the installation instructions are a little different.

If the product is purchased using Electronic Software Delivery (ESD), you receive a .zip file for RAD Studio or an executable (.exe) file for Delphi or C++Builder.

To install using an ESD download:

1. Make sure you have an active Internet connection.
2. If you are installing RAD Studio:
 - Extract the .zip file into an empty directory on your hard drive. **Note:** Do not attempt to run the installer directly from the .zip file without first extracting all files to a local directory.
 - Double-click the install launcher. The install launcher will appear with buttons to install multiple items that are part of the product.
 - Select Delphi and/or C++Builder to begin product installation
3. If you are installing Delphi or C++Builder: Double-click the setup.exe to begin product installation.

E-Learning Series: Getting Started with Windows and Mac Development

4. When prompted, enter the serial number for the product. You can find the serial number in the order confirmation email.
5. The installer downloads the required .zip files and uncompresses them.
6. The installer asks you to supply an archive location for the .zip files. **Note:** Backing up the downloaded ZIP files is strongly encouraged so that in the future you can do a reinstall without downloading all the ZIP files again.
7. The installer proceeds automatically. Follow the prompts.

If you purchased a Network License (managed by AppWave™ or the Embarcadero License Server), follow these instructions to set up the install image. **Note:** Follow these instructions precisely or your users will be prompted for a serial number and will be unable to install.

To install using a media kit:

1. Copy all of the DVD files to a writable directory on your network.
2. Copy the network license .slip file (received in your order confirmation email) to the product's main installation directory, where Setup.exe is located (typically, <Disc Drive:>\DelphiCBuilder\Install).
3. Instruct your users to run Setup.exe. When a user runs Setup.exe, the product detects the .slip file and installs the software. During installation, the .slip file is moved to the license folder in the installation directory.
4. The installer proceeds automatically. Follow the prompts.
5. Optionally, you can install the Microsoft SDK Help. By default, it is not installed.
6. Optionally, you can install any or all of the third party products. Installed unless you de-select them.

To install from a download URL:

1. Copy just the Setup.exe and .slip file into the same directory on your machine.
2. Run Setup.exe and the installer will read the slip to determine which SKU (product version) you have purchased.
3. The installer will copy the .slip file to the correct directory.
4. The installer proceeds automatically. Follow the prompts.
5. Optionally, you can install the Microsoft SDK Help. By default, it is not installed.
6. Optionally, you can install any or all of the third party products. Installed unless you de-select them.

To install a Trial edition:

If you are going to use the trial edition you will find the Delphi, C++Builder and RAD Studio trial downloads at

- Delphi XE2 Architect - <https://downloads.embarcadero.com/free/delphi>
- C++Builder XE2 Architect - https://downloads.embarcadero.com/free/c_builder
- RAD Studio XE2 Architect - https://downloads.embarcadero.com/free/rad_studio

Home > Products > Application Development > FireMonkey > FireMonkey Download

Overview Test

FireMonkey Download

FireMonkey is free but to create FireMonkey applications you need to download Delphi XE2, C++Builder XE2 or RAD Studio XE2.

Choose a RAD tool:

Delphi XE2

Embarcadero® Delphi® XE2, now with FireMonkey™, is the fastest way to deliver ultra-rich and visually stunning native applications for Windows, Mac and iOS, allowing developers to dramatically reduce coding time and build applications 5x faster.

30-day trial Buy a copy

[Learn more >](#)

C++Builder

Embarcadero C++Builder® XE2, now with FireMonkey™, is the only true visual C++ development environment - designed for ultra-fast, component-based creation of highly-maintainable, visually stunning GUI applications for Windows and Mac OS X.

30-day trial Buy a copy

[Learn more >](#)

RAD Studio XE2

Embarcadero® RAD Studio XE2 is the ultimate application development suite and the fastest way to build data-rich, visually engaging applications for Windows, Mac, mobile, .NET, PHP and the Web.

30-day trial Buy a copy

[Learn more >](#)

Each Free, 30-day fully-functional trial includes the IDE, compilers, libraries and examples. The trial does not include the command line compilers, VCL/FireMonkey source code and some bundled tools. The latest trial versions include Update 4. There are three parts to successfully receiving a trial installation:

1. Select the trial product you want to download and click the download button. After you accept the export agreement, your browser will download a zip file containing the installer. The installer will download additional files as needed.

The screenshot shows the Embarcadero website's 'Select Product' page. The page has a dark header with the Embarcadero logo, social media links, and navigation menus. The main content area is titled 'Select Product' and includes a progress indicator with three steps: 'Step 1: Select Product' (active), 'Step 2: Register/Sign-In', and 'Step 3: Check Email'. Below the progress indicator, there is a table with columns for 'PLATFORMS', 'LANGUAGES', 'VERSION', and 'SIZE'. The table lists 'Embarcadero RAD Studio XE2 Architect - 30 day trial' with a description, trial details, and supported languages. A row shows 'Windows' as the platform, 'XE' as the version, and '9.6MB' as the size. There are 'PRODUCT INFO' and 'DOWNLOAD' buttons next to the row. At the bottom, there is a footer with 'HOME', 'COPYRIGHT', 'PRIVACY', 'FEEDBACK', and a copyright notice for 2011 Embarcadero Technologies, Inc.

PLATFORMS	LANGUAGES	VERSION	SIZE
Windows	English, French, German and Japanese	XE	9.6MB



2. You will be asked to Register or Sign-in to the Embarcadero Developer Network (EDN) so that Embarcadero can send you your trial key.

embarcadero Follow Us: [Twitter] [YouTube] [Facebook] [LinkedIn] [Shopping Cart] FREE TRIALS EDN SELECT COUNTRY search...

COMPANY PRODUCTS SOLUTIONS RESOURCES NEWS & EVENTS SUPPORT

Register/Sign-In

✓ Step 1: Select Product
Step 2: Register/Sign-In
Step 3: Check Email

You have selected **Embarcadero RAD Studio XE2 Architect - 30 day trial**. To download this product, please complete the registration form or sign-in below. Upon completion, you will receive further information and installation instructions at the email address you entered in the registration form.

Sign-In »
If you have registered with us before, please login.

Your Login: [Text Field]
Password: [Text Field]
[Forgot Password](#) **LOGIN**

Note: After registering, you have free access to the Embarcadero Developer Network (EDN). Download free trials of our products, access special community areas, members-only technical papers, on-demand videos and webinars, plus other technical resources. You can even select time zone and language preferences.

First name * [Text Field]
Last name * [Text Field]
Email * [Text Field]
Company name * [Text Field]
Individual *
Student *
Country * -- Please Select -- [Dropdown]
State * -- Please Select -- [Dropdown]
Phone * [Text Field]
Create Password * [Text Field]
Retype password * [Text Field]

* Required Fields
 Yes, I want to receive general marketing communications regarding this product and related Embarcadero products and events.

SUBMIT

All information collected will be kept confidential. View our Privacy Policy.

embarcadero Follow Us: [Twitter] [YouTube] [Facebook] [LinkedIn] [Shopping Cart] FREE TRIALS EDN SELECT COUNTRY search...

COMPANY PRODUCTS SOLUTIONS RESOURCES NEWS & EVENTS SUPPORT

Check Email

✓ Step 1: Select Product
✓ Step 2: Register/Sign-In
Step 3: Check Email

Thank You

Your download of **Embarcadero RAD Studio XE2 Architect - 30 day trial** should start in a few seconds, and an email with important information has been sent to david.intersimone@gmail.com.

If your download does not start automatically, please click [here](#).

Your trial period is **30 days** from the time you install the product (some variations occur, please check email).

Please ensure your spam filter will allow messages from activation@embarcadero.com.

Server Response from: ETNACDC02

GETTING STARTED
▶ Product Information
▶ Community
▶ Downloads

RELATED CONTENTS
▶ White Papers
▶ Product Demos
▶ Documentation

HOME COPYRIGHT PRIVACY FEEDBACK © Copyright 2011 Embarcadero Technologies, Inc. All Rights Reserved.

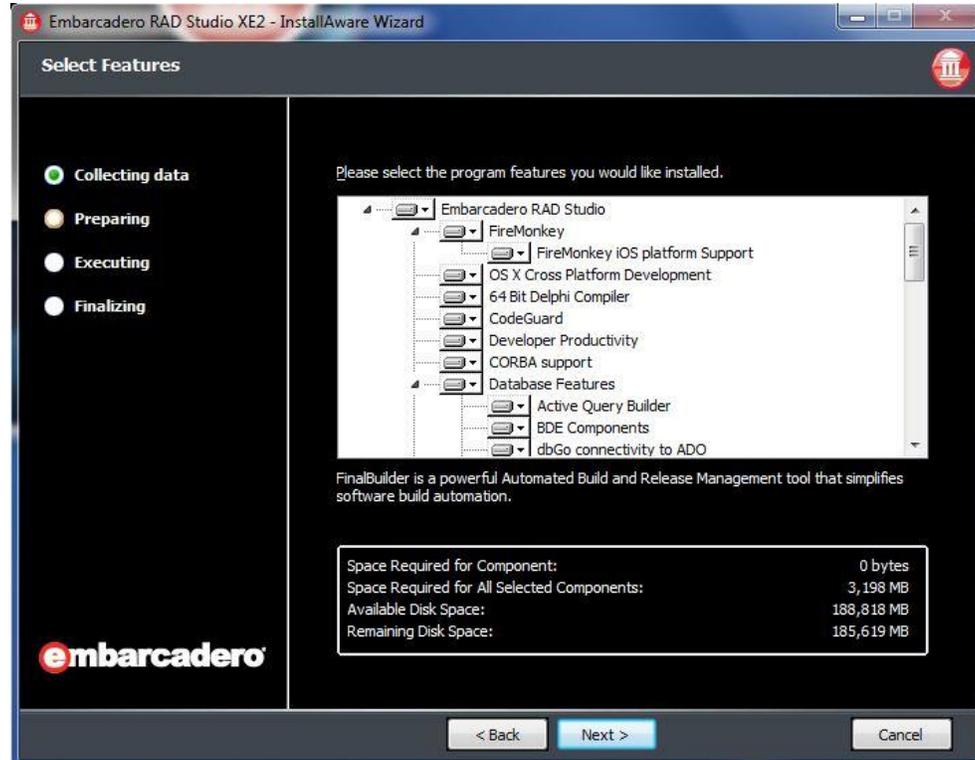
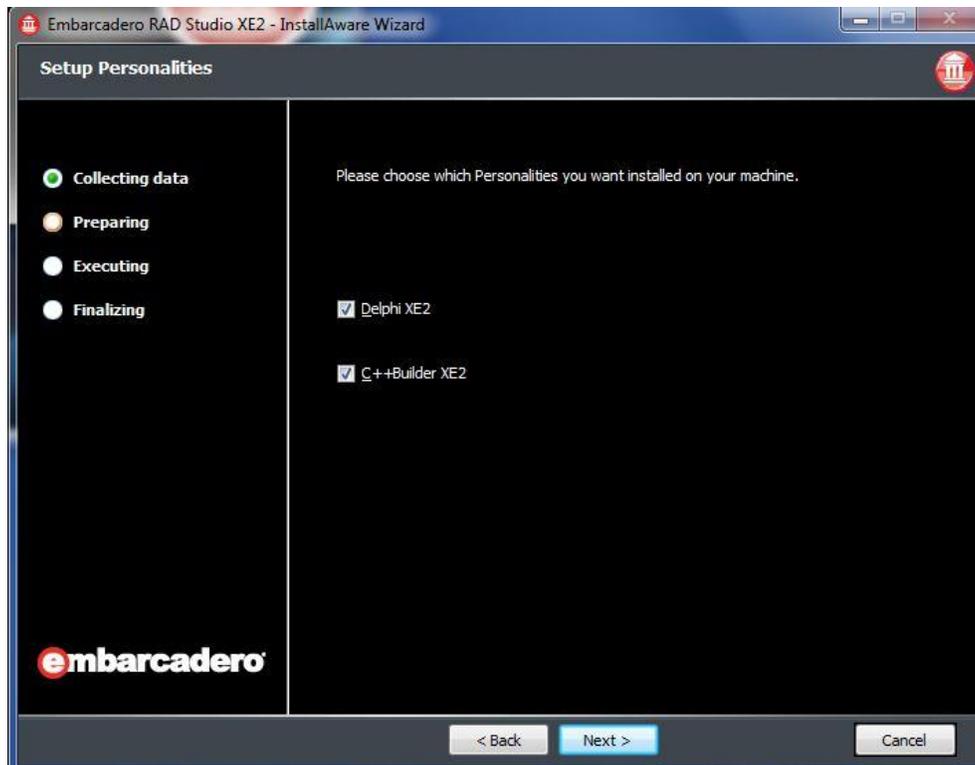
3. Check your email for the trial key that will be sent to you. Your trial period is **30** days from the time you install the product (some variations occur, please check email). Please ensure you spam filter will allow messages from “activation@embarcadero.com”.



Follow these steps to install the trial

1. Make sure you have an active Internet connection.
2. If you are installing the RAD Studio trial:
3. Extract the .zip file into an empty directory on your hard drive. Note: Do not attempt to run the installer directly from the .zip file without first extracting all files to a local directory.
4. Double-click the install launcher executable. The install launcher will appear with buttons to install multiple items that are part of the product.
5. Select Delphi and/or C++Builder to begin product installation
6. If you are installing Delphi or C++Builder trial: Double-click the setup.exe to begin product installation.
7. When prompted, enter the serial number for the product. You can find the serial number in the trial confirmation email.
8. The installer downloads the required .zip files and uncompresses them.
9. The installer asks you to supply an archive location for the .zip files. Note: Backing up the downloaded ZIP files is strongly encouraged so that in the future you can do a reinstall without downloading all the ZIP files again.
10. The installer proceeds automatically. Follow the prompts.

E-Learning Series: Getting Started with Windows and Mac Development



E-Learning Series: Getting Started with Windows and Mac Development

If all went well during the installation, you should be able to start the RAD Studio IDE, register (one time only) the product using your Embarcadero Account ID, Password and Product Key.



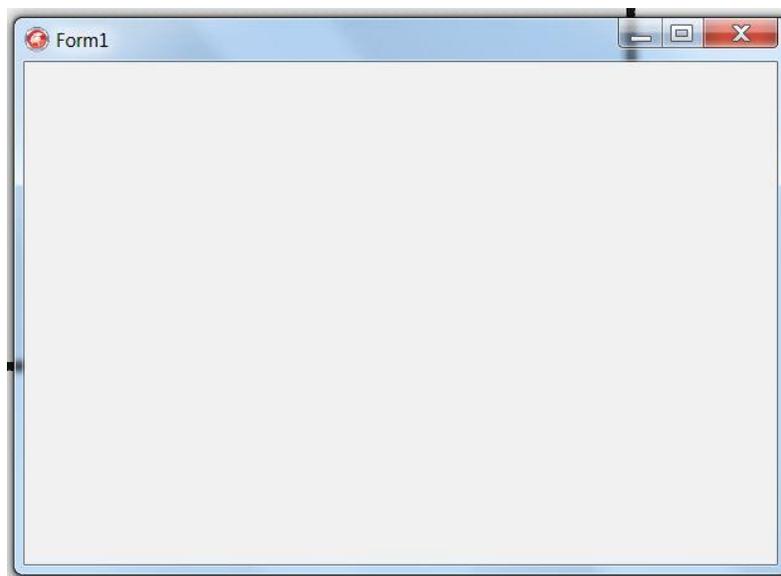
After the splash screen appears, you will be looking at the RAD Studio IDE.



Select the **Help > About** menu item from the main menu to bring up the about box (that should look something like the following screen – notice that I have updated the help files to update 5):



To make sure we can build FireMonkey applications for Windows, let's use the FireMonkey HD application wizard. From the main menu, select **File > New** and choose the "FireMonkey HD Application – Delphi" or "FireMonkey HD Application – C++Builder" menu item. Click on the Run icon (the green play button on the tool bar) or hit the F9 hot key. The project will be compiled and you should see a main application window appear with the caption "Form1".



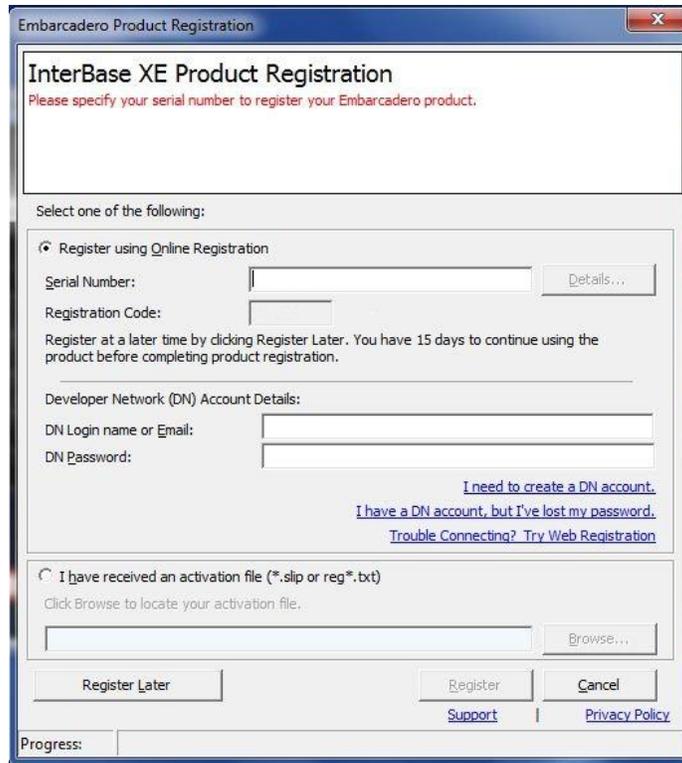
Click on the close button to end the program and return back to the IDE. You've built your first FireMonkey Windows 32-bit application! Your IDE is ready to help you create FireMonkey applications.

Additional instructions for the installation process are available at http://docwiki.embarcadero.com/RADStudio/en/Installation_Notes_for_XE2.

Installing InterBase XE Developer Edition

To be able to build database applications, the RAD Studio installer also includes Embarcadero InterBase XE Developer Edition. You can install this by clicking the Install InterBase on the main RAD Studio install window. Links to the InterBase XE Developer Edition for Win-32 and Win-64 will be displayed along with a Developer Edition key you can use to register InterBase.





We will cover more about building database applications for Windows and Mac in Lesson 6.

Installing the Platform Assistant(s)

One of the major benefits of using FireMonkey is the ability to create applications for both Windows and Mac OS X from one codebase. When you create a Mac OS X application, you need to use RAD Studio on the Windows operating system, and target your application to Mac OS X. You can debug your applications running on Mac OS X, and then package them for Mac OS X. To debug and deploy an application to Mac OS X, you need to have the Platform Assistant installed on your Mac OS X machine. You can also use the Windows version of the Platform Assistant to debug applications running on other Windows 32 and 64-bit computers (including Windows instances running in the Amazon EC2, Microsoft Azure or other cloud infrastructures).

The platform assistant does more than just facilitate debugging. You can also use IDE and the platform assistant to deploy your application and any additional files that your application needs. For example, your application might need additional DLLs, INI files, data files, bitmaps, etc.

In order to use the IDE to develop cross-platform applications, you need to install and run the Platform Assistant on the target platform. The Platform Assistant file name is 'PAServer', the remote application server. If you use the default port number (64211), you should be able to use the Platform Assistant without changing any parameters.

Installing PAServer on your Windows and or Mac OS X machine

You will need to transfer the PAServer installer to your remote Windows machine, and run the installation. The Windows PAServer installer can be found at:

```
C:\Program Files\Embarcadero\RAD Studio\9.0\PAServer\setup_paserver.exe
```

You will need to transfer the PAServer installer to your Mac OS X machine, and run the installation. The Mac PAServer installer can be found at:

```
C:\Program Files\Embarcadero\RAD Studio\9.0\PAServer\setup_paserver.zip
```

PAServer is version specific for each version of the IDE. If you install a new version of the IDE, you must uninstall and reinstall and re-install the PAServer version that comes with the new version of the IDE. For example, when we released Update 4 of XE2, there was also an updated version of the Platform Assistant.

Updating the PAServer for an updated IDE release

If you are using the Platform Assistant for cross-platform application development, after you install an IDE update (for example XE2 Update 4), you should uninstall the Platform Assistant and then install the updated Platform Assistant.

To uninstall the Platform Assistant: On Windows use the existing Uninstall PAServer.exe to uninstall the Platform Assistant. On the Mac, use the existing Uninstall command to uninstall the Platform Assistant. Note: The Uninstall programs are located in the installation directory for the Platform Assistant on the PC and on the Mac.

To install the updated Platform Assistant on your target platforms: On Windows double-click PAServer.exe. On the Mac, move the PAServer.zip file to the Mac and double-click PAServer.zip.

Starting PAServer on a Mac

PAServer is a console application. Open the Terminal window on the Mac, go to `/Users/<your accountname>/Applications/Embarcadero/PAServer`, and then type:

```
./paserver
```

PAServer will ask you to provide a Remote Profile password to allow the RAD Studio IDE to connect to your Mac. You can define a remote password, or leave it blank. PAServer will then display “Acquiring permission to support debugging...” and a dialog box will pop up with the text “Developer Tools Access is trying to take control of another process. Type your password to allow this”. Enter your Mac administrator name and password and press click the “Take Control” button.



Notes:

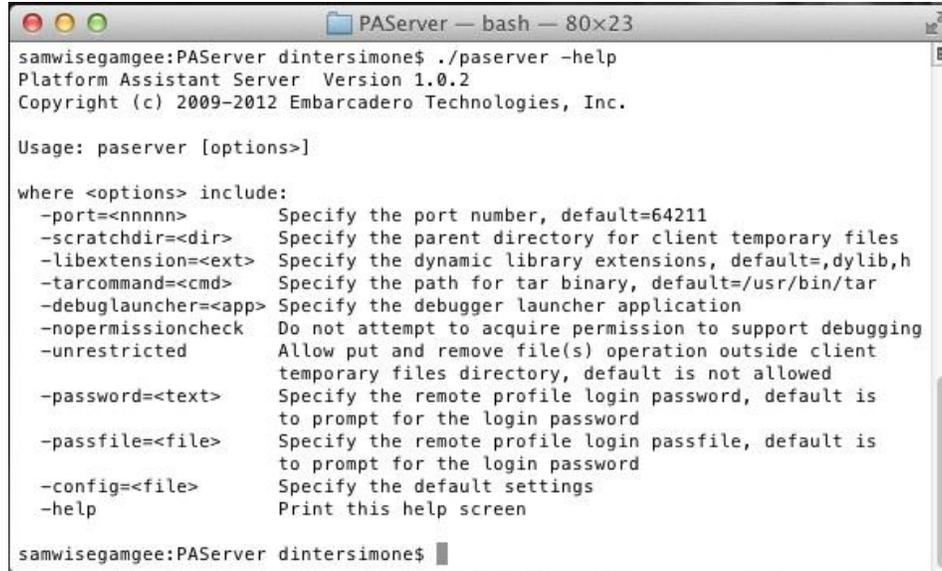
- The Developer Tools Access dialog only appears the first time you start the Platform Assistant during any session on the Mac. If you do not enter the correct password (or if you press Cancel), the message failed is displayed on the Mac, followed by “You will be prompted for Developer Tools Access the next time you start the debugger.”
- You can run and deploy your OS X applications for the rest of your session on the Mac (but you cannot yet debug). When you attempt to run with debugging (such as Run or F9), the Developer Tools Access dialog box appears. You need to log on to Developer Tools Access in order to debug your OS X application using the IDE.
- To delay the Developer Tools Access step, start the Platform Assistant with the `-nopermissioncheck` command-line option: `./paserver -nopermissioncheck`. This option delays the need to enter the password for Developer Tools Access until you actually debug an application.

PAServer Command-Line Options

To display "verbose" help, start PAServer from the command line and include the `-help` option:

Platform Assistant Server Version 1.0.2
Copyright (c) 2009-2011 Embarcadero Technologies, Inc.
Usage: paserver [<options>]
where <options> include:

<code>-port=<nnnnn></code>	Specify the port number, default=64211
<code>-scratchdir=<dir></code>	Specify the parent directory for client temporary files
<code>-libextension=<cmd></code>	Specify the dynamic library extensions, default=,dylib, h
<code>-tarcommand=<cmd></code>	Specify the path for tar binary, default=/usr/bin/tar
<code>-debuglauncher=<app></code>	Specify the debugger launcher application
<code>-nopermissioncheck</code>	Do not attempt to acquire permission to support debugging
<code>-unrestricted</code>	Allow put and remove file(s) operation outside client temporary files directory
<code>-password=<text></code>	Specify the remote profile login password (default: prompt for the password)
<code>-passfile=<file></code>	Specify the remote profile login passfile (default: prompt for the password)
<code>-config=<file></code>	Specify the default settings
<code>-help</code>	Print this help screen



```
PA Server — bash — 80x23
samwisegamgee:PA Server dintersimone$ ./paserver -help
Platform Assistant Server Version 1.0.2
Copyright (c) 2009-2012 Embarcadero Technologies, Inc.

Usage: paserver [options>]

where <options> include:
  -port=<nnnnn>          Specify the port number, default=64211
  -scratchdir=<dir>      Specify the parent directory for client temporary files
  -libextension=<ext>    Specify the dynamic library extensions, default=,dylib,h
  -tarcommand=<cmd>     Specify the path for tar binary, default=/usr/bin/tar
  -debuglauncher=<app>  Specify the debugger launcher application
  -nopermissioncheck    Do not attempt to acquire permission to support debugging
  -unrestricted         Allow put and remove file(s) operation outside client
                        temporary files directory, default is not allowed
  -password=<text>      Specify the remote profile login password, default is
                        to prompt for the login password
  -passfile=<file>      Specify the remote profile login passfile, default is
                        to prompt for the login password
  -config=<file>        Specify the default settings
  -help                Print this help screen

samwisegamgee:PA Server dintersimone$
```

Displaying Help when PAServer is running

After PAServer is running on the target platform, you can enter h at the command prompt to display help for the five commands to which a PAServer process can respond:

- q - stop the server
- c - print all clients
- p - print port number
- s - print scratch directory
- g - generate login passfile



```
dintersimone — paserver — paserver — 80x24
Last login: Tue May 1 16:29:22 on ttys000
samwisegamgee:~ dintersimone$ /Users/dintersimone/Applications/Embarcadero/PAServer/paserver ; exit;
Platform Assistant Server Version 1.0.2
Copyright (c) 2009-2012 Embarcadero Technologies, Inc.

Remote Profile password <press Enter for no password>:

Acquiring permission to support debugging...succeeded

Starting Platform Assistant Server on port 64211.
>h
q - stop the server
c - print all clients
p - print port number
s - print scratch directory
g - generate login passfile
>p
64211
>s
/Users/dintersimone/Applications/Embarcadero/PAServer/scratch-dir
>
```

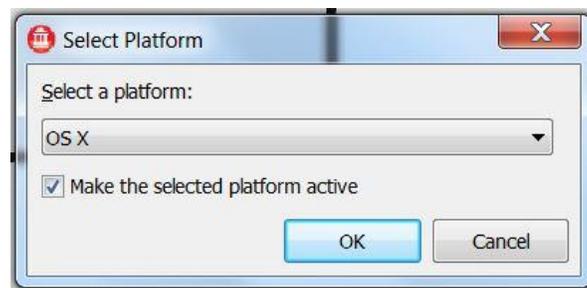
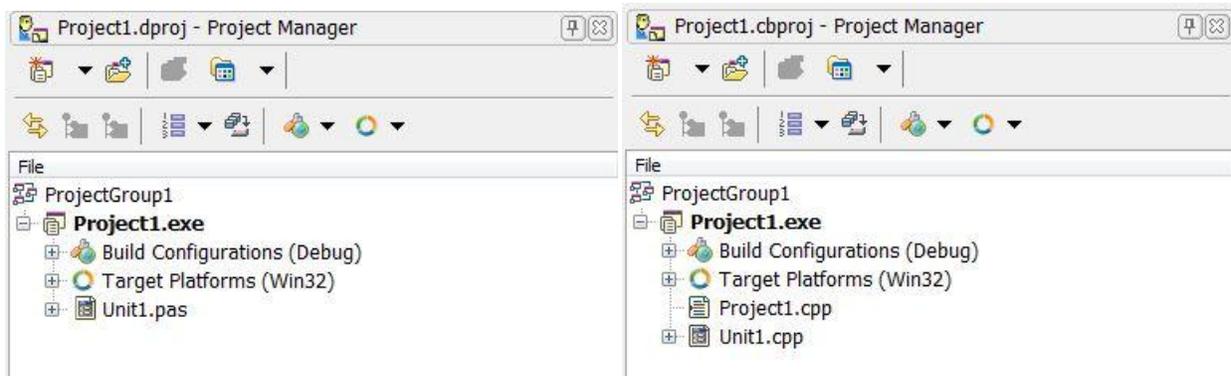
Notes:

- The scratch directory is used by the Platform Assistant for temporary file operations, including remote profiles and project files.
- The .passfile is a password file that contains an encrypted password for the Platform Assistant; you can share this file with anyone who needs to use the connection, without compromising the privacy of the password.

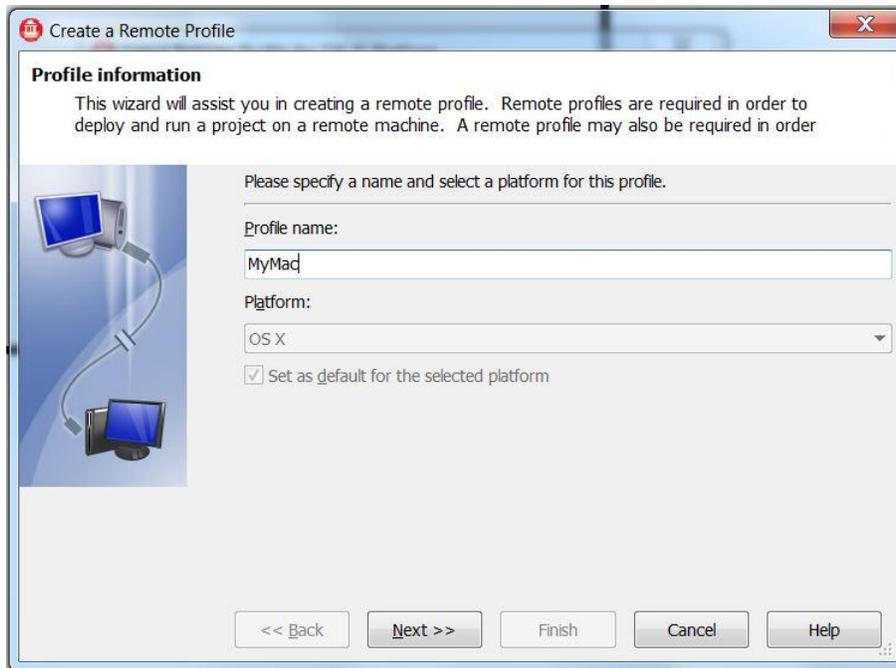
Testing your PAServer Installation

To make sure we can build FireMonkey applications for the Mac, use the FireMonkey HD application wizard. From the main menu, select **File > New** and choose the “FireMonkey HD Application – Delphi” or “FireMonkey HD Application – C++Builder” menu item (if you didn’t close the IDE when you created your FireMonkey application to test the installation, you don’t have to repeat these steps; the starting project is already created).

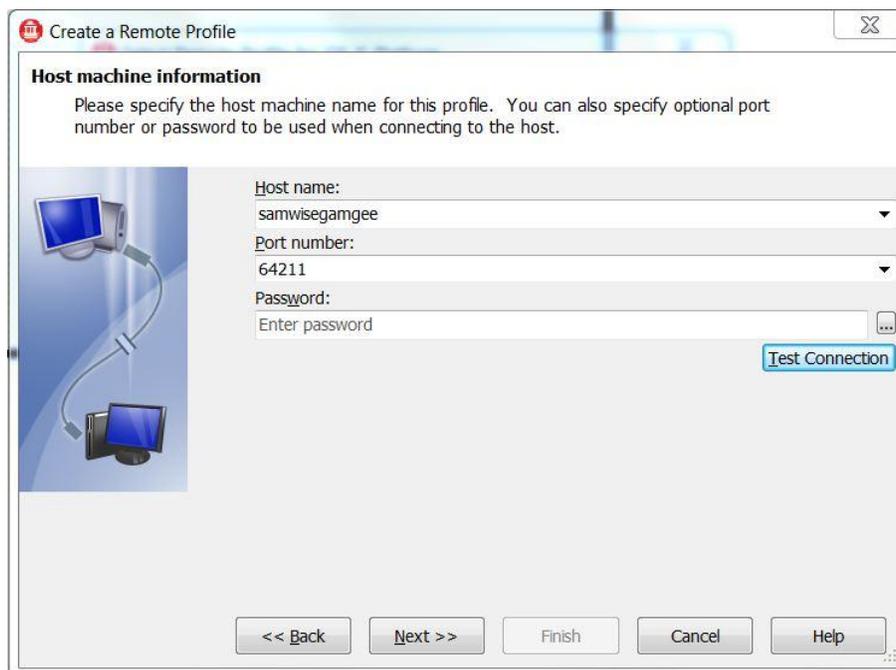
In the project window of the IDE (it is on the upper right hand side), right mouse click on the “Target Platforms” node for the project and select “Add platform...” and choose “OS X” from the platform list and click the OK button.

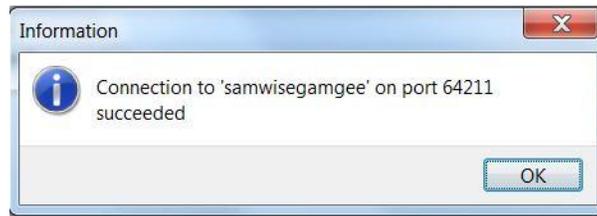


Expand the “Target Platform” node in the project window and right mouse click on the “OS X” target to bring up a popup menu and choose the “Assign Remote Profile...” menu item. Next, click the “Add...” button, in the “Select Remote Profile for OSX Platform” window to start the remote profile wizard. Give your remote profile a name (“MyMac”) and click the “Next” button.



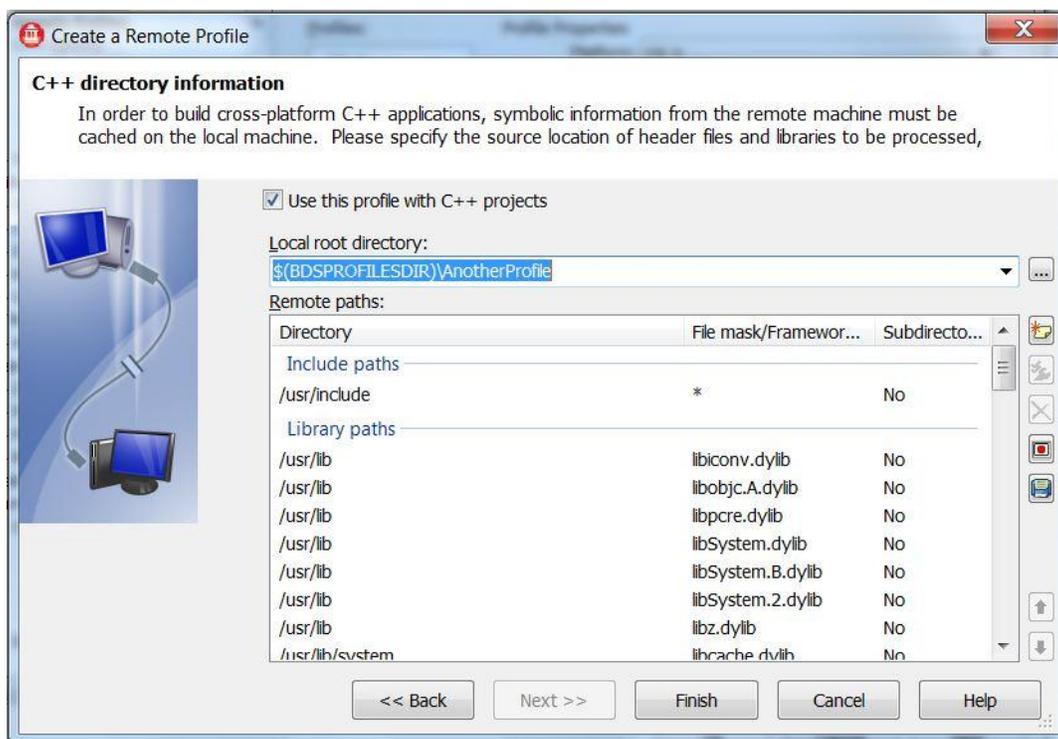
The Remote Profile wizard next asks you for the Host Name (or IP address) of your Mac. Also, notice that the wizard lists the default port number used by PAServer (you can change the port number to match the number you set in PAServer). If you set a password for the PAServer, key the same password in the wizard dialog. Click the “Test Connection” to make sure you’ve entered the right information. If you get the “connection succeeded” dialog, click the “OK” button. Click the “Next” button in the wizard.





The final step in creating a remote profile (C++Builder projects only) allows you to set up a list of directory paths needed by the C++ compiler and linker. If you are building Delphi FireMonkey applications you can click the “Finish” button to complete the remote profile.

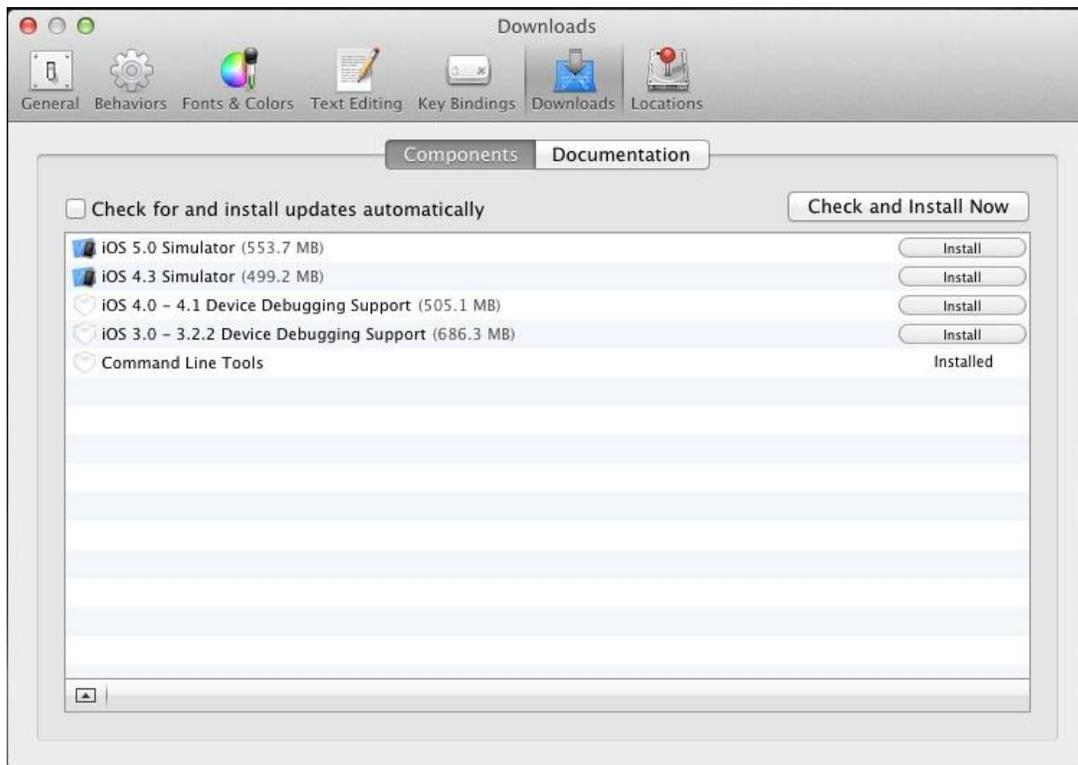
If you are building C++ FireMonkey applications, this is where you can specify the location of Mac header files and libraries to be processed. The wizard will populate the remote paths lists with the standard Mac paths.



Note for C++ developers: If you are using Xcode 4.3.x or later, Apple has bundled the include files, library files and other files into the Xcode app bundle, so there won't be any files found in the “/usr/include” and “/usr/lib” folders. In order for the RAD Studio IDE to be able to cache information from these files,

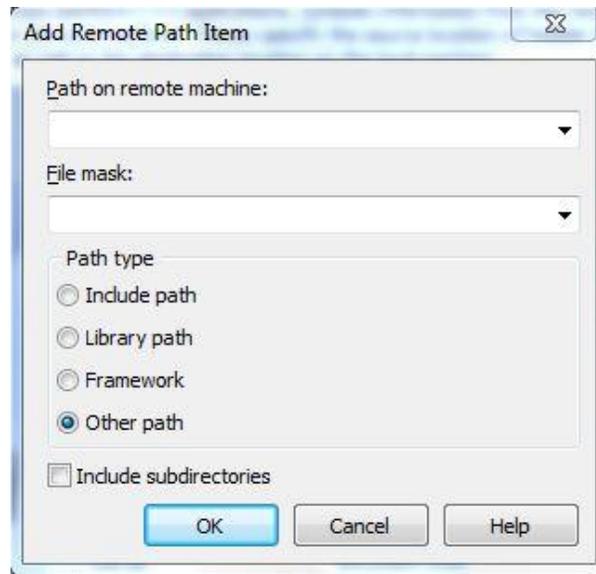
E-Learning Series: Getting Started with Windows and Mac Development

you need to install the Xcode command line tools. In Xcode choose the **Xcode > Preferences** menu item and in the preferences window, click on the Downloads icon. If you've already installed the command line tools you will see the following information:

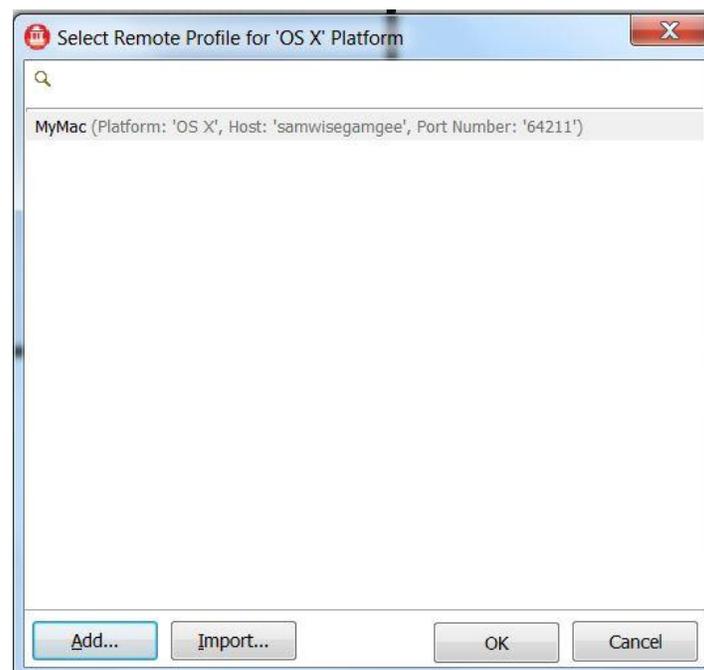


If you have not installed the Xcode command line tools, click on the Install button to the right of the Command Line Tools entry.

You can also click the "Add a new path item" button to enter additional paths for any additional Mac include and library files.



After you've added any additional paths on your remote machine, click the "Finish" button on the Create remote profile window. Your Mac remote profile is now setup.

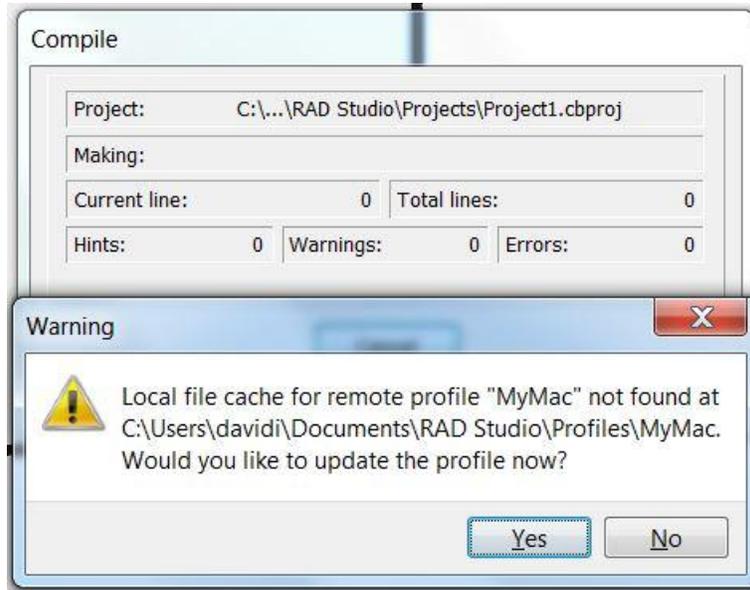


Click the "OK" button to select the remote Mac OS X profile for your project.

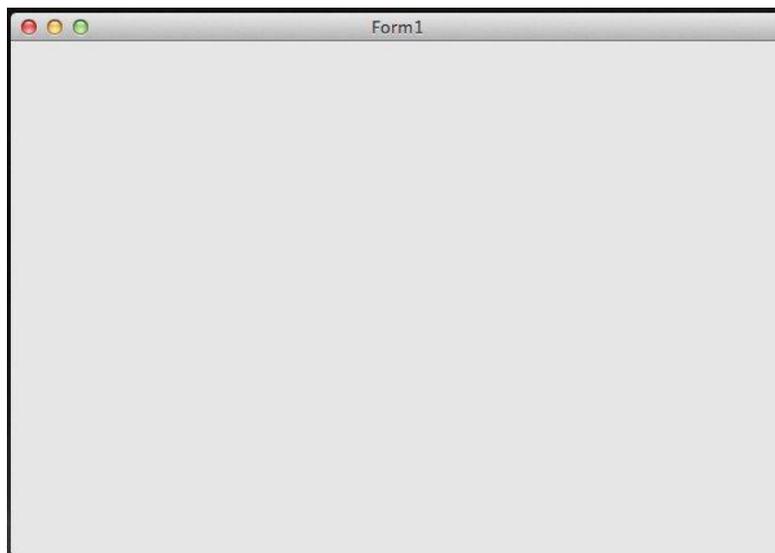
Check to make sure that you can compile and run the empty form application on your Mac. Click on the Run icon (the green play button on the tool bar) or hit the F9 hot key.

E-Learning Series: Getting Started with Windows and Mac Development

If you are building a C++ project and this is the first time you've used your Mac remote profile, a warning dialog will pop up saying that a local cache for profile "MyMac" was not found and asks you if you would like to update the profile now. Click the "Yes" button. If you are building a Delphi project, you will not see this warning dialog.



The project will be compiled and you should see a main application window appear on your Mac with the caption "Form1".



Click on the close button to end the program on your Mac. Return back to the IDE. You've built your first FireMonkey Mac application!

Installing the FireMonkey tools for iOS

You can also create FireMonkey applications targeting iOS. To build an application targeting iOS, you need to install Xcode and the Free Pascal compiler, as well as the special FireMonkey-iOS library, on your Mac OS X machine. Your project's files must be shared between RAD Studio and Xcode. To facilitate this sharing, you must store your files in a location that is accessible to both programs at the same time. You can do this in any of the following ways:

- A network file server
- A local folder shared by the PC and the Mac
- Your virtual machine environment if you are running Windows in a VM on your Mac

Note: We recommend that you set up a shared folder on the Mac and save your project files directly to the Mac from RAD Studio.

In order to build, run, debug, and deploy iOS apps on the Mac, you need to join one or two Apple developer programs, obtain a code-signing certificate, and create provisioning profiles (all described in this section). You can join the Apple Mac and iOS developer programs as an individual developer or as a member (or leader) of a team in an enterprise (business) program or university program. For more information about the developer programs, see <http://developer.apple.com/programs/which-program/>. After you join a program and log in using your Apple Developer ID, you will find the appropriate platform- and program-specific documentation available to you. Please use the Apple documentation as your principal guide.

Join the Apple Developer Program - <http://developer.apple.com/programs/register/> (Required; Registration is free). Joining the Apple Developer Program:

- Gives you your Apple Developer ID. A Registered Apple Developer user ID is required for iOS application development.
- Gives you access to the basic software tools (such as Xcode and iOS SDK) that you need in order to develop applications on the Mac platform (part of FireMonkey iOS development occurs on the Mac).
- Enables you to test your iOS apps using simulators in Xcode.
- Gives you the Apple documentation about iOS development.
- Allows you to report bugs and attend Apple developer events.

Join the iOS Developer Program - <http://developer.apple.com/programs/ios/> (Requires an annual payment of \$99 to Apple), Joining the iOS Developer Program:

- Gives you access to developer tools, pre-release software, and technical support
- Enables you to place your completed iOS applications in the Mac App Store.
- Gives you access to the portal through which you create the Development Provisioning Profile, which is necessary for running your applications to development iOS devices.

For additional information about joining the Apple iOS Developer's Program, see <http://developer.apple.com/support/ios/enrollment.html>.

E-Learning Series: Getting Started with Windows and Mac Development

A similar program, the Mac Developer Program - <http://developer.apple.com/programs/mac/>, is available for Mac developers. If you want to develop for both the Mac and iOS and place your Mac and iOS apps in the App Store, you need to join both developer programs (Mac and iOS). If you plan to place only your iOS apps (not any Mac apps) in the App Store, you can join only the iOS Developer Program.

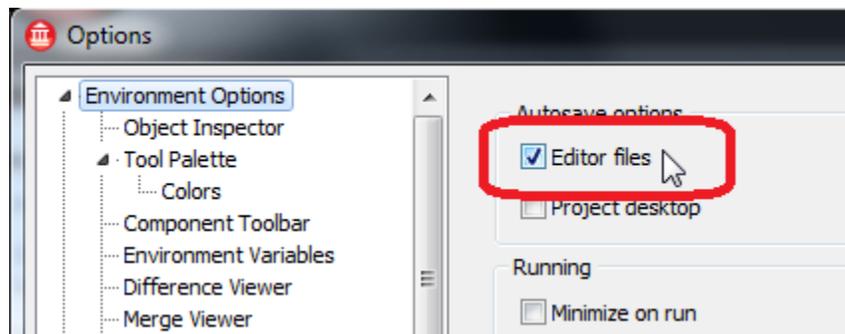
To build iOS applications (Delphi and RAD Studio only), use the following installation and setup steps on your development Mac:

1. Install Xcode on the Mac (once you are a member of the iOS Developer Program, you can obtain a download of the specific required versions of Xcode and the iOS SDK by visiting the iOS downloads area on the Apple Developer Portal).
2. Find the FireMonkey-iOS.dmg disk image file on your development PC and transfer it to the Mac. The FireMonkey-iOS.dmg is typically installed in the RAD Studio installation location, under the “C:\Program Files (x86)\Embarcadero\RAD Studio\9.0\FireMonkey-iOS” directory:
3. On the Mac, open FireMonkey-iOS.dmg and run the two installer packages:
 - fpc-2.6.0.intel-macosx.pkg
 - FireMonkey-iOS-XE2.pkg



Setting up RAD Studio to support development of iOS projects

RAD Studio iOS projects share project files with Xcode through the file system. Turning on the Autosave option can avoid unnecessary round trips when you thought a file was saved for compiling but actually was not. To enable Autosave, go to the menu item **Tools > Options**, then select Environment Options and ensure that the “Editor files” check box is selected in the Autosave options.

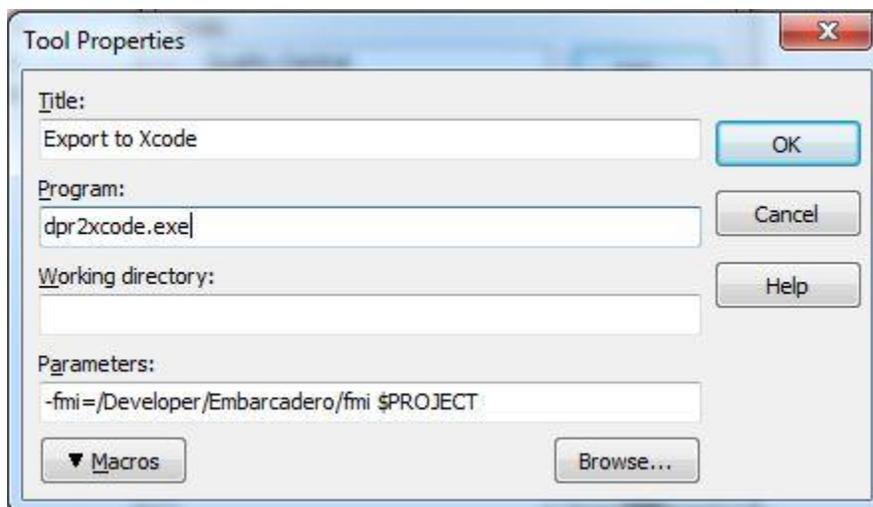


RAD Studio iOS projects must be exported as Xcode projects to run on the iOS simulator or iOS device. The dpr2xcode utility, installed in the “/bin” directory, performs this export for you.

You must run dpr2xcode at least once for each project to generate the initial Xcode project, and run dpr2xcode again whenever changes or additional files are added to the project. Dpr2xcode saves the project files (the ones you need for iOS development on the Mac) in an Xcode folder inside your project directory. You need to transfer the Xcode folder to the Mac in order to continue iOS app development. I use a shared folder so that I can work on the design and code in my RAD Studio IDE and compile, debug and run in Xcode on my Mac.

Adding a Tools menu item makes it easy to run dpr2xcode within RAD Studio. To automate dpr2xcode, follow these instructions:

1. Use the **Tools > Configure Tools** menu item.
2. On the Tool Options dialog box, click Add.
3. On the Tool Properties dialog box, complete the fields as follows:

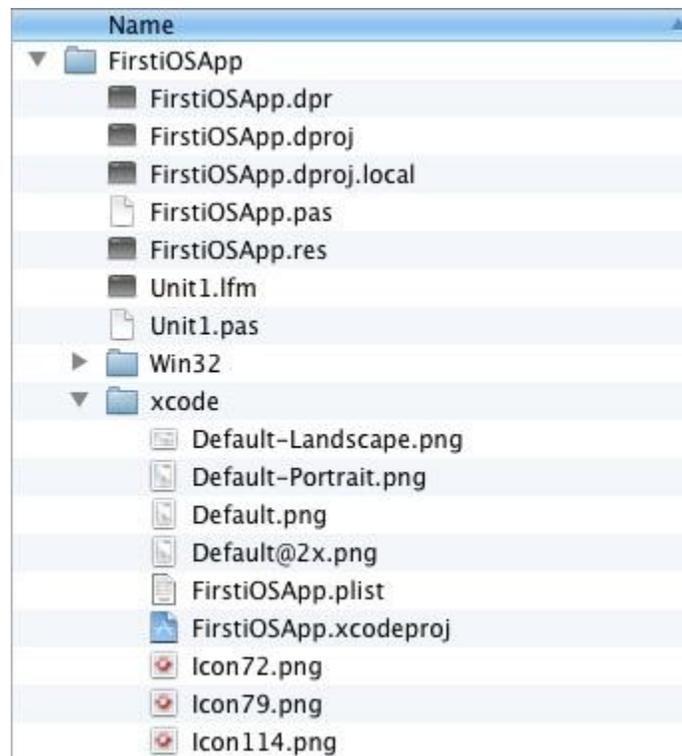


- Title: **Export to Xcode** (or whatever descriptive text you want)
- Program: **dpr2xcode.exe**

E-Learning Series: Getting Started with Windows and Mac Development

- Working directory: (leave this blank)
- Parameters: **-fmi=/Developer/Embarcadero/fmi \$PROJECT** (Note: The FMI sources are typically installed in /Developer/Embarcadero/fmi. If you did not install the FMI sources in this location, add the location in the Parameters field as follows: *-fmi=/myFmiLocation \$PROJECT*)

Make sure you check to see that you have installed the FireMonkey for iOS software. We'll use the **File > New > FireMonkey HD iOS – Delphi** project wizard. Save the project using **File > Save All** and put the project in a shared folder where your Mac can get at it (I named the folder and project “FirstiOSApp”). Select **Tools > Export to Xcode** (to use dpr2xcode to create the Xcode project and files). Move to your Mac, start Xcode and load the Xcode version of the project, (which is in the Xcode sub-folder) named “FirstiOSApp.xcodeproj”.



Click the “Run” button to compile and execute the app using an iOS simulator that is installed (or on your iOS device if that is provisioned and connected). If the iOS application builds and runs, then your setup is ready for you to get started with iOS application development.

E-Learning Series: Getting Started with Windows and Mac Development



This information should get you started if you need to get a head start building iOS applications. There are a few additional steps required to be able to provision your iOS device, Code-Sign your application and upload your iOS applications to the Apple AppStore. We will focus more attention on building iOS applications later in this E-Learning Series.

Summary, Looking Forward, To Do Items, Resources, Q&A and the Quiz

In Lesson 1 you were introduced to building Windows and Mac application using the FireMonkey business application platform. You learned how to download, install and setup the RAD Studio environment. You installed the Platform Assistant Server (PAServer) and created a remote profile to enable you to build applications for Windows and Mac using one codebase. Finally, you learned how to install the FireMonkey for iOS tools and how to join the Apple Developer programs for Mac and iOS.

In Lesson 2, you'll see how to create your first Windows and Mac desktop applications using Delphi and C++. Delphi developers will also create your first iOS application.

E-Learning Series: Getting Started with Windows and Mac Development

In the meantime, here are some things to do, articles to read and videos to watch to enhance what you learned in Lesson 1 and to prepare you for lesson 2.

To Do Items

If you don't own and/or haven't installed RAD Studio XE2, Delphi XE2 or C++Builder XE2, download the free trial and follow the Lesson 1 course book to install and setup XE2. Make sure you know how to use the **File > New** project wizards to create and test that your installation is complete and your development environment is set up. This will prepare you to take part in future lessons.

Links to Additional Resources

- Getting Started Course landing page - <http://www.embarcadero.com/firemonkey/firemonkey-e-learning-series>
- The FireMonkey Application Platform - http://docwiki.embarcadero.com/RADStudio/en/FireMonkey_Application_Platform
- What's new in Delphi and C++Builder XE2 - http://docwiki.embarcadero.com/RADStudio/en/What%27s_New_in_Delphi_and_C%2B%2BBuilder_XE2
- Solutions for Windows and Mac Software Application Development - <http://www.embarcadero.com/solutions/solutions-for-windows-and-mac-software-application-development>
- FireMonkey Datasheet - http://www.embarcadero.com/images/firemonkey_datasheet_21mar12.pdf
- Video: Building your first Delphi and C++Builder FireMonkey HD Applications - <http://edn.embarcadero.com/article/42097>
- InterBase Labs – Step by Step Tutorials - <http://www.embarcadero.com/database-in-action/interbase-labs>

Q&A:

Here are some of the answers for the questions I've received (so far) for this lesson. I will continue to update this Course Book during and after course.

Q: I can't see David I's Mac screen on the Delphi lesson 1 webinar.

A: Attendees couldn't see the Mac demos in the Delphi lesson because I didn't log in to "Go to Webinar" using the Mac client. Sorry :(The Delphi lesson 1 replay has the Mac parts. Sorry for my mistake. It will not happen in future webinars.

Q: "Is right to left" support for visual controls on the roadmap?

A: "Right to Left" support is on the roadmap for FireMonkey.

Q: Can I compile my Delphi applications for windows and Mac on my windows machine to send it to a friend who uses a Mac without using PAServer?

A: Yes, compiling a Mac OS X application is done in the Windows IDE using our compiler for the Mac. You can grab the Mac application bundle in the OSX32 sub-folder for your project and give it to friends who have Macs.

E-Learning Series: Getting Started with Windows and Mac Development

Q: Can iOS code only be generated from a Delphi codebase? Or are there plans to translate C++ source as well?

A: In XE2 we are supporting FireMonkey development for iOS using Delphi. Our R&D teams are working on next generation compilers that will support ARM processors to allow you to build native code applications for iOS and Android.

Q: Will XE2 have a set of iOS styles and icons so XE2 apps look like iPhone apps? Or do we have to supply our own icons and styles? (Apple is really picky about look and feel of apps they sell in their App Store)

A: FireMonkey applications are being accepted by Apple. Several Apps are already in the App Store. You have full control over the look of your FireMonkey applications using styles.

Q: Will there be a speed improvement so iOS HD apps will load faster?

A: We're working on performance improvements in FireMonkey updates and future releases.

Q: Has the licensing of XE2 DataSnap changed so developers can use it in commercial apps without paying royalties?

A: That changed a very long time ago. When you purchase the Enterprise or Architect edition of our products, you can deploy DataSnap with your applications without any royalties.

Q: Can FM iOS apps be installed on other people's iPhone without going through the Apple App Store?

A: Yes, using so called AdHoc deployment. You can also use Xcode and a USB cable to deploy apps to connected iOS devices.

Q: Will these sessions be recorded?

A: Yes. The lesson replays, Course Book, example programs and other information will be available on the "Getting Started" course landing page at <http://www.embarcadero.com/firemonkey/firemonkey-e-learning-series>

Q: Will there be a Database Connector SDK? I use databases which you don't natively support. I'd like to have the tools to connect them correctly.

A: You can use DBExpress, ODBC, ADO and other database connector technologies with FireMonkey. You can connect direct to SQLite on iOS using calls from your FireMonkey iOS applications. Anders Ohlsson showed a FireMonkey SQLite iOS example on his blog at <http://blogs.embarcadero.com/ao/2011/10/04/39146>

Q: Is Linux on the roadmap?

A: Linux support is on the roadmap.

Q: I have everything setup and working for iOS 5.0 but I can't send anything to a device because they are all iOS 5.1. If I update to the latest Xcode will Free Pascal and FireMonkey still work?

A: In order to do iOS 5.1 you will need to upgrade to Xcode 4.3 and Lion. FireMonkey for iOS works with iOS 5.1. Read Anders Ohlsson's blog post at <http://blogs.embarcadero.com/ao/2012/03/16/39235> for additional information and comments. Anders also has a large number of blog posts about iOS development with FireMonkey. You can find the list at <http://blogs.embarcadero.com/ao/category/ios>.

Q: How is the FireMonkey integration with TeeChart?

E-Learning Series: Getting Started with Windows and Mac Development

A: TeeChart Lite for FireMonkey is included with XE2. Steema Software has the full version of TeeChart for FireMonkey. You can find additional product information at

http://www.steema.com/entry/71/TeeChart_2C_RAD_Studio_XE2_26_FireMonkey

Q: When I get XE2 updates, do I need to install them in order or just grab the latest update?

A: The XE2 even numbered updates require that you uninstall and reinstall the product. The odd numbered updates can be run as regular updates.

Q: Are there any tools for migrating my current VCL to FireMonkey?

A: Check out MidaConverter at <http://www.midaconverter.com/>

Q: Will you ever support Voxel based graphics?

A: Nothing is currently planned for FireMonkey, but I will forward your request to the Product Management and R&D teams.

Q: Why don't we see FireMonkey components like there are for VCL?

A: The source code for FireMonkey is in the product, just like it is for the VCL. If you are asking about additional third party components for FireMonkey, several of our technology partners are shipping (and/or working on) FireMonkey components including Steema Software (<http://www.steema.com/entry/71>), TMS Software (<http://www.tmssoftware.com/site/tiw.asp>, <http://www.tmssoftware.com/site/tmsfmxpath.asp>, <http://www.tmssoftware.com/site/tmsfmxtableview.asp>), Mitov Software (<http://www.mitov.com/>) and Fast Reports(<http://www.fast-report.com/en/news/9497.html>).

Q: Can we use PAServer for iOS development?

A: PAServer is not used for iOS development in this release. We currently use Xcode for iOS debugging and deployment to your iOS device.

Q: Would be great to see David I in our University of Wolverhampton England. We need XE2 workshop in our Uni? Would you like to come and do workshop in our Uni?

A: I would be very happy to do a workshop at your university if I am invited and we can work it into my travel schedule to Europe (I am in the UK several times each year).

Q: Is Blackberry 10 on the roadmap?

A: We have a Java mobile DataSnap connector for BlackBerry at this time. You can write apps using Java for BlackBerry and connect to your Windows based DataSnap servers. RadPHP can also build mobile applications for BlackBerry devices.

Q: Could you confirm here that Mac developer membership is free?

A: You can develop for Mac for free. If you want to submit apps to the Mac Store you have to be an Apple Developer at \$99/yr. Additional information is available on Apple's developer programs site at <https://developer.apple.com/programs/>

Q: What XE2 update # do you recommend to have installed for programming FMX apps?

A: Update 4. A hot fix for Update 4 is coming shortly, and I would recommend using it as soon as it comes out.

Q: Is RAD Studio for \$99US offer still active?

E-Learning Series: Getting Started with Windows and Mac Development

A: Delphi and C++ customers can upgrade to RAD Studio until the end of June 2012. Offer details are available at <http://www.embarcadero.com/radoffer>

Q: Is the official roadmap posted somewhere?

A: John Thomas (JT), Director of Product Management for Developer Tools has blogged about future directions for our developer tool products. You can find JT's post at <http://blogs.embarcadero.com/itembarcadero/2011/09/17/may-the-roadmap-rise-with-you/>.

Q: FireMonkey seems to be very GUI-centric at this point, but how much of the VCL is leveraged (can be leveraged)? For example, TComponent is used as a common ancestor, and there's a new TControl and TForm, etc. It took years to develop the VCL, so I'm assuming it will take some time to get FireMonkey to a mature state.

A: FireMonkey is a new business application platform. It is a modern platform abstraction layer built on top of industry standard graphics libraries and hardware for building High Definition and 3D applications. FireMonkey is built on years of Embarcadero's component development experience. FireMonkey is already available on Win32, Win64, Mac OSX and iOS. FireMonkey works with most of the non-visual controls that Delphi and C++Builder developers have been using for years. FireMonkey works with the runtime libraries that have been available for many releases of our tools. It is being used by developers all over the world. Apple has already accepted several FireMonkey based iOS apps in their App Store. FireMonkey will continue to grow and evolve to meet the platform, hardware and software architectures for years to come.

Q: How soon are we going to see Delphi XE2 books in the market?

A: As soon as the authors complete their books. We've been told that Bob Swart and Marco Cantu are both working on XE2 books :)

Q: Is Xcode 4.3 supported? If not, any idea when?

A: Yes. I am using latest version of Xcode 4.3 and OS X Lion on my Mac. Anders Ohlsson has blogged about upgrading to Xcode 4.3 at <http://blogs.embarcadero.com/ao/2012/03/16/39235>

Q: What about support for a report engine on other platforms?

A: Any non-visual reporting controls that work with XE2 and generate PDF files can be used to print reports. For reporting engines that work on Windows and Mac, FastReports is now demoing FastReport for Windows and Mac. I blogged about this (with links) at <http://blogs.embarcadero.com/firemonkey/index.php/2012/05/04/fast-reports-announces-fastreport-fmx-for-firemonkey/>

Q: I have help update 4, when I use check for updates, I get a message that there are no updates available. How can I get help update 5?

A: Information about XE2 help update 5 is available at http://docwiki.embarcadero.com/RADStudio/en/Readme_-_Help_Update_5_for_Delphi_and_C%2B%2BBuilder_XE2

Q: If I am running XE2 on 64-bit Windows (in a Mac virtual machine), can I test 32-bit applications on the same 64-bit Windows platform or do I need to target it to a different 32 bit platform. If on same platform, do I need a remote profile and pa server to create/run the 32 bit app on the 64 bit system?

A: You can compile, debug and run Delphi Win32 and Win64 and C++ Win32 applications on a 64-bit Windows machine. If you are running the IDE on 32-bit Windows you can compile 32-bit and 64-bit

E-Learning Series: Getting Started with Windows and Mac Development

applications but you will need to use PAServer running on 64-bit Windows target to be able to deploy and debug the application.

Q: The FireMonkey 30 day trial will not cover the timeframe for the entire course. Will the trial be extended?

A: My hope is that you will see the value and productivity using FireMonkey during the first few lessons and upgrade to XE2. If not, I will work with our sales teams to see about extending the trial period.

Q: Will Android be supported?

A: Android native code support is on the roadmap.

Q: Does FireMonkey support Unicode?

A: Yes, FireMonkey is all Unicode. Delphi and C++Builder have supported Unicode since our 2009 releases.

Q: I don't see the Mac OS X target platform in my XE2 installation.

A: This attendee may have the Starter Edition which only supports FireMonkey and VCL for Win32.

Q: Will VCL updated and move forward?

A: VCL will be updated as long as Microsoft Windows provides the Windows API as it current exists or is extended. For the next release of Windows (Windows 8), Microsoft has announced that they still support 32-bit and 64-bit applications that use the Windows API. FireMonkey is our business application platform for the future. Using FireMonkey will allow you to move your native code applications to additional platforms in the future using the same codebase.

Q: Is the Project Indy set of TCP/IP components an individual package or part of the RAD studio? Can it be used on Mac OS too?

A: Project Indy components are included in XE2 and work on both Windows and Mac.

Q: What is the relationship between the RAD studio & FireMonkey?

A: RAD Studio XE2 = Delphi + C++Builder + Prism + RadPHP. FireMonkey is our Business Application Platform for creating HD and 3D applications on Windows, Mac and iOS.

Q: Can C++Builder XE2 Trial run beside C++ XE Professional?

A: Yes you can.

Q: What is the video hardware requirement for FireMonkey HD applications?

A: If you need any 3D aspects then you need a video card with a GPU that supports at least DirectX version 9 (which includes just about every graphics chip included in PCs and Macs for the past 7+ years). If you don't use any FireMonkey image effects or 3D, FireMonkey will support graphics processors that work with GDI+. From the XE2 product datasheet: Basic GPU – any vendor DirectX 9.0 or better (Pixel Shader Level 2).

Q: I am running XE2 Enterprise on Win7-32. I have FireMonkey HD in my File/New menu, but not the list of FireMonkey choices that "David I" has. Am I missing something?

A: David has customized his menu. Use the **File > New > Other...** menu item and you'll see all of the project choices. To customize your **File > New** menu, use the **File > New > Customize** menu item.

E-Learning Series: Getting Started with Windows and Mac Development

Q: Why is it called FireMonkey?

A: Why not? What would you call it?

Q: Will C++ Builder 2010 work with FireMonkey?

A: FireMonkey only works with our Delphi and C++Builder XE2 releases. Mac support for GUI and command line applications is only supported in XE2.

Q: Will FireMonkey work to create notification area apps?

A: It may not be in the box, but you have full access to the Windows API and the Mac OS X API in your applications.

Q: Can earlier C++Builder apps be upgraded to FireMonkey?

A: To build for Mac you need to port your VCL applications to FireMonkey. Most of your non-Windows specific non-visual controls should be usable in FireMonkey applications. Check out the MIDA Converter conversion tool to migrate the forms and code in your Delphi and C++Builder VCL apps.

Q: After class, I installed RAD Studio XE2 Architect trial. I need a better understanding of the various editions so I'll research that.

A: The Architect editions contain everything you need to create a variety of applications. We have 5 different editions: Starter, Professional, Enterprise, Ultimate and Architect at different price points. Which edition you need depends on what types of applications you are going to build. Each comes with FireMonkey (Starter Edition is Windows only). You can take a look at the Feature Matrix for each of the products to see what is included in each of the editions for Delphi, C++Builder and RAD Studio. Each feature matrix is linked on the right hand column of each product page on our web site:

- <http://www.embarcadero.com/products/cbuilder/cbuilder-feature-matrix.pdf>
- <http://www.embarcadero.com/products/delphi/delphi-feature-matrix.pdf>
- <http://www.embarcadero.com/products/rad-studio/rad-studio-feature-matrix.pdf>

Q: Should I drop this class if I don't know C++? I am an experienced VBA programmer, and took two semesters of Java in 2007. Should I still follow along just to learn using FireMonkey?

A: You don't specifically need to know C++ or Delphi to follow either of the courses. You can do a lot of work with no or few lines of code - depending on how much business logic code you need to write. Since you have a lot of VBA experience - you might have an easier time with the Delphi course. In the programming language lecture, I will just cover some of the programming things you need to know to work with FireMonkey - I am not going to try and teach a whole programming language in 30 minutes :) There are plenty of videos and tutorials on the Internet about programming in Delphi and C++ to help you:

Delphi:

- Embarcadero DocWiki - Delphi Reference - http://docwiki.embarcadero.com/RADStudio/en/Delphi_Reference
- Delphi Basics - <http://www.delphibasics.co.uk/>
- About Delphi Programming - <http://delphi.about.com/>
- A Beginner's Guide to Delphi - <http://delphi.about.com/od/beginners/a/delphicourse.htm>
- Introduction to Delphi - <http://www.functionx.com/delphi/Lesson01.htm>

C++:

- A Quick Introduction to C++ by Tom Anderson (29 pages) <http://www.cs.washington.edu/homes/tom/c++example/c++.pdf>
- MIT Open Courseware: Introduction to C++ - <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-096-introduction-to-c-january-iap-2011/>
- C++ Language Tutorial - <http://www.cplusplus.com/doc/tutorial/>
- Bjarne Stroustrup – The C++ Programming Language home page - <http://www2.research.att.com/~bs/C++.html>
- Embarcadero DocWiki C++ Reference - http://docwiki.embarcadero.com/RADStudio/en/C%2B%2B_Reference
- C++Builder tutorials and VCL topics – a little dated, but an easy place to learn - <http://www.functionx.com/bcb/index.htm>
- C++ Builder Practical Learning Series – again a little dated but still classic tutorials - <http://www.yevol.com/en/bcb/>
- C++Builder Developer’s Journal - <http://bcjournal.org/>
- Programming Abstractions – Stanford, Instructor - Julie Zelenski - successor to Stanford iTunesU “Programming Methodology” course. Covers such advanced programming topics as recursion, algorithmic analysis, and data abstraction using the C++ programming language - <http://itunes.apple.com/WebObjects/MZStore.woa/wa/viewPodcast?id=384232917>

If you have any additional questions – send me an email - davidi@embarcadero.com

Self Check Quiz

1. With RAD Studio XE2 (the combination of Delphi XE2 and C++Builder XE2) you can create native code applications for:

- a) Win 32
- b) Win 64
- c) Mac OS X
- d) Apple iOS
- e) All of the above

2. What RAD Studio tool allows the IDE to debug and deploy applications to Mac and other remote Windows machines?

- a) Remote debugger
- b) Remote desktop
- c) PAServer
- d) Dpr2Xcode

3. Using FireMonkey you can create HD (High Definition) and 3D applications

E-Learning Series: Getting Started with Windows and Mac Development

- a) True
- b) False

4. What do we call the feature that allows you to create connection information for remote computers?

- a) Remote Desktop
- b) Remote Profile
- c) Cross Platform Configuration
- d) Remote Configuration

Answers to the Self Check Quiz:

1e, 2c, 3a, 4b