XOJO

Upgrade Guide

2015 Release 3

Xojo, Inc.

Chapter 1

Welcome to Xojo

Enabling ordinary people to create extraordinary apps!

Section 1

Overview

Thanks for trying Xojo. This Guide is designed for people who are upgrading from the prior Real Studio product.

About the Name Change

As many of you may know, Xojo was previously called Real Studio.

As others of you may know, the original name for Real Studio was CrossBasic. We couldn't keep that name because it was a trademark registered by another company. We considered many names and decided on REALbasic as the product name and Real Software as the company name. While REALbasic was a comforting name for some, no doubt invoking a feeling of nostalgia of the thrilling early days of BASIC, for others it was exactly the opposite. We heard from many of you that hearing a development environment with the word "basic" in the name caused many of your colleagues' eyes to roll. Memories of line numbers and an interpreted language that was designed to teach programming rather than write powerful software applications, left some with the wrong idea about what we had created.

After many years, we decided to change the name to Real Studio hoping that this would improve things. And it did. Some developers found they were no longer battling the reputation of BASIC while others were confused because we continued to call the language itself, REALbasic. Real Studio, as a name, has its own problems though. First, because it's not unique and is made up of two real words (no pun intended), it is not a name we can own. Google for example, can't tell the difference between a musician who says, "I'm recording in a real studio now," and a programmer who says, "I'm programming in Real Studio now."

With the new IDE now available and mobile support coming next year, we realized that if we were ever going to rename Real Studio and solve these problems once and for all, this would be the time. Our goal is to have a single brand that we can own.

Choosing a name is exceedingly difficult. First you have to find a name you like. This name has to then pass a variety of tests. For example, ideally it doesn't mean "broken", "slow" or "stinky" in

any language you care about. It can't already be in use by another company in your general product category. In our case that's electronics, computer hardware and software. It needs to either perfectly describe your product or not describe anything at all. In that latter case, you will make the name mean your product.

Starting with this release, Real Studio is now Xojo and Real Software, Inc. is Xojo, Inc. Real Studio and REALbasic will be completely retired. No mention of them will be made in the product, documentation or on our website. Anyone who goes to realstudio.com, realbasic.com or realsoftware.com will see a short explanation about the name change and be redirected to xojo.com.

Some of you will like the name change, some will be neutral and some will dislike it. You can't please everybody. But we think this is the right move and is the right time.

All-New User Interface

This is the 3rd redesign of the user interface. The first design appeared with the initial release of REALbasic back in 1998. It consisted of multiple windows scattered across the desktop and was very reminiscent of Mac applications from the 1990s.

This design was replaced with the release of REALbasic 2005 in June, 2005. This new version used a tab-based design similar to how a web browser works. Each project item was visible in its own tab.

Xojo uses a single window design that makes it easier to navigate your project. This design has been in development for almost two years. Our team has spent an enormous amount of time and energy on it and we are quite proud of the improvements it brings, some of which are:

- Better project organization
- Eliminates reliance on tabs, although tabs can still be used
- More modern UI look and feel
- Faster UI
- Improved usability
- Easier to learn
- Cocoa on OS X

Other Changes

Of course we haven’t just been working on the user interface, there are plenty of changes to other areas, particularly Cocoa and Web Edition.

This Guide is intended to help people already familiar with Real Studio get up to speed quickly with Xojo. But in addition to this Guide, you will also want to spend some time looking over the all-new User Guide books:

- Book 1: Fundamentals

- Book 2: User Interface
- Book 3: Framework
- Book 4: Development

The QuickStarts and Tutorials have also been updated:

- QuickStart: Desktop
- QuickStart: Web
- Tutorial: Desktop
- Tutorial: Web

The Licensing Change

Xojo moves to an all-new licensing model that makes it much easier for people to try and learn Xojo.

Xojo is free to use for development with only two restrictions:

- You cannot build stand-alone apps.
- You cannot save in the XML or Text project formats. Only the binary project format may be used.

You purchase a license to enable these features. The Xojo Pro license enables all features of the IDE, including the ability to save in any project format and built for any target on any platform (Desktop, Console, Web, Databases for OS X, Windows and Linux).

If you do not need access to all the targets, you can instead purchase licenses for just the parts you need.

Chapter 2

User Interface Changes

Learn about the all-new user interface. For full details, refer to the User Guide.

Section 1

Project Chooser

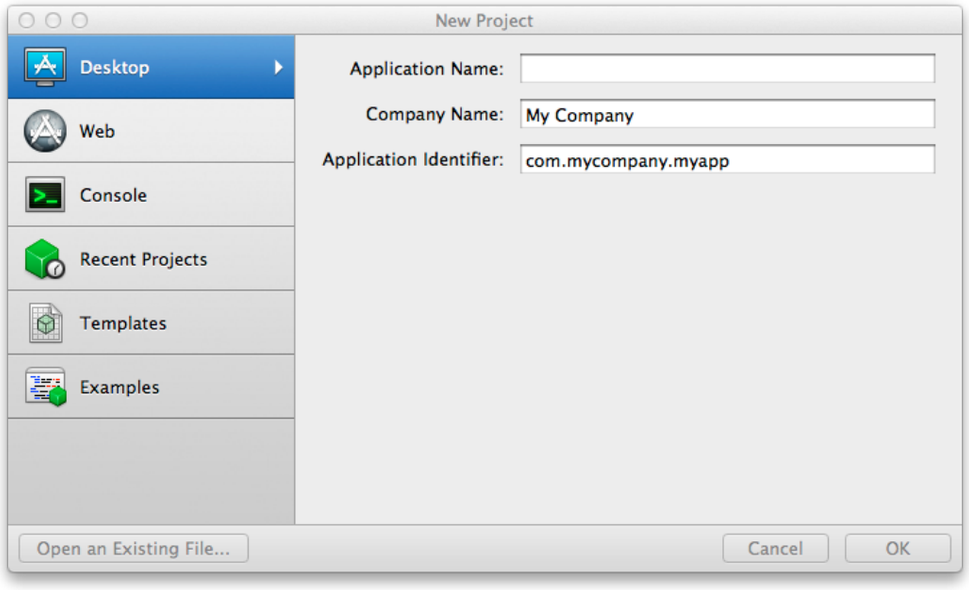
The first thing you see when launching Xojo is the **Project Chooser** window.

Much like the New Project window in prior versions of Real Studio, the Project Chooser is where you select the type of new project you want to create or open a recent project, template, example or another existing project.

Note that when you choose to create a new project, you may optionally specify the Application Name, Company Name and Application Identifier immediately.

Once you've made your choice, click OK to show the main window, referred to as the Workspace.

Figure 2.1 Project Chooser Window

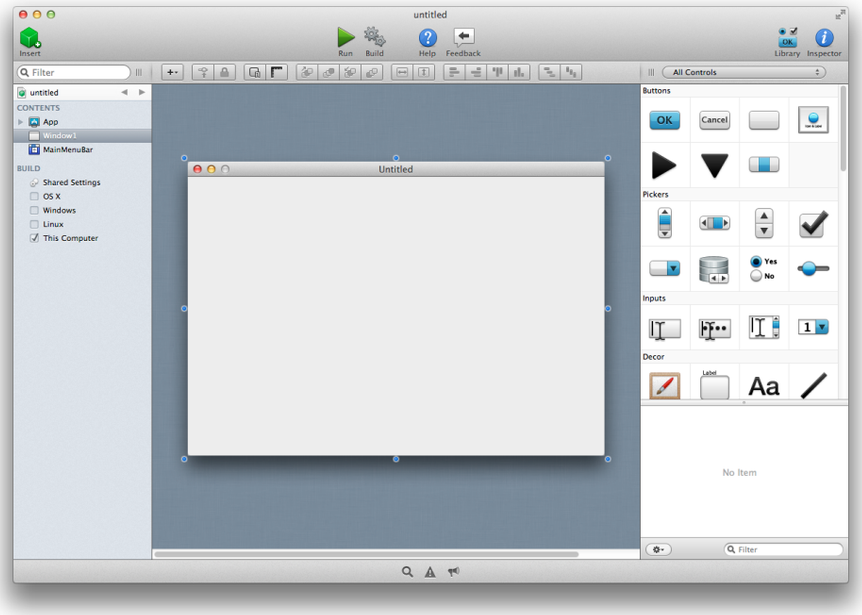


First Looks: The Workspace

The New User Interface

Your first look at the new user interface reveals a completely different layout. Pretty much nothing looks the same!

Figure 2.2 The Xojo User Interface



The main thing to notice is that the window is now divided into four areas:

- **Navigator** is on the left
- **Editors** are in the center
- **Library** and **Inspector** are on the right
- **Panels** are on the bottom

These areas can each be resized, but their placement cannot be changed. The Library/Inspector can be displayed as palettes. And the Library/Inspector and Panels can also be hidden to maximize available space for the editor.

There is an all-new toolbar, which is not customizable. The Editors also have their own all-new toolbars, which are not customizable.

In addition to these obvious changes, the menus have been simplified.

Finding existing features in the changed user interface might be a challenge at first, so do take the time to review this guide before you jump right in.

Navigator

What is the Navigator?

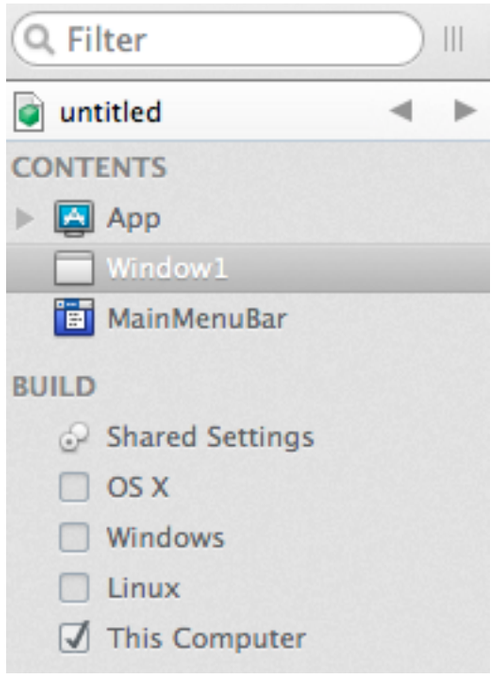
The Navigator is the area on the sidebar on the left that allows you to navigate through your project. The Navigator has three sections, Contents , Run and Build Settings.

The **Contents** section contains the items in your project. These are the windows, web pages, classes, menus and other objects that are used to create your application.

You can click on project items in the Contents section to view them or edit them.

When an item is selected, you can use the arrow keys to move the selection between items.

Figure 2.3 Navigator Showing the Contents Section



The **Profiles** section is only visible after you run your project with Profiling enabled. It remains on screen if you have profile results to review.

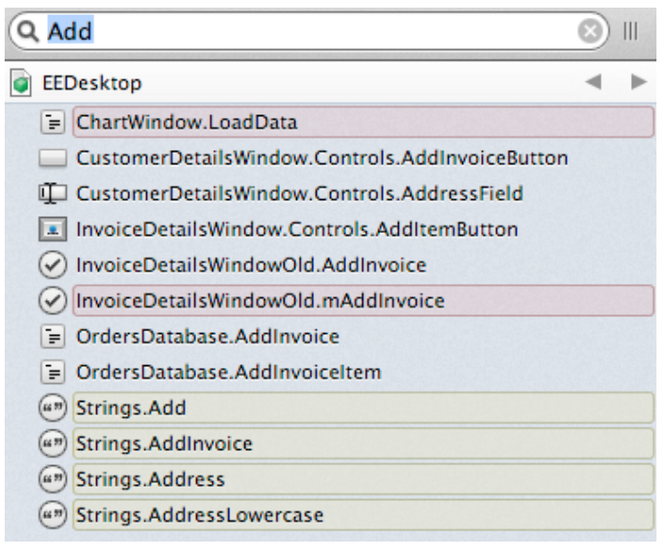
The **Builds Settings** section contains all build-related information, including the build targets (and their settings) and active build steps used by Build Automation.

Filter

The Navigator has a Filter field at the top that can be used to filter what is displayed in the Contents section. Use the Filter to quickly show specific project items based on your criteria. For example, if you know you have a method that is called “AddInvoice”, but you don’t recall what class or window it is on, you can just type “Add” in the Filter field. The Navigator will display all project items that have something containing “Add” in its name (e.g. a method, control name, constant, property).

You can then click on the project item to see it or edit it.

Figure 2.4 Using the Filter

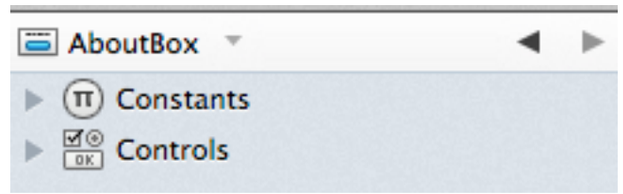


Jump Bar

The Navigator displays items using a hierarchical list. The scope of what is displayed is controlled by the Jump Bar. By default, your entire project is in scope, so the Jump Bar displays your project name.

When you double-click on an item that is a parent, the Jump Bar changes to show the parent. Now only the items that are its children are displayed in the Navigator.

Figure 2.5 Jump Bar with AboutBox Selected



Use the left and right arrows to move back and forth through the history. Click the name in the Jump Bar to see the entire history and to jump quickly to a specific item.

The Jump Bar is incredibly powerful when used with tabs.

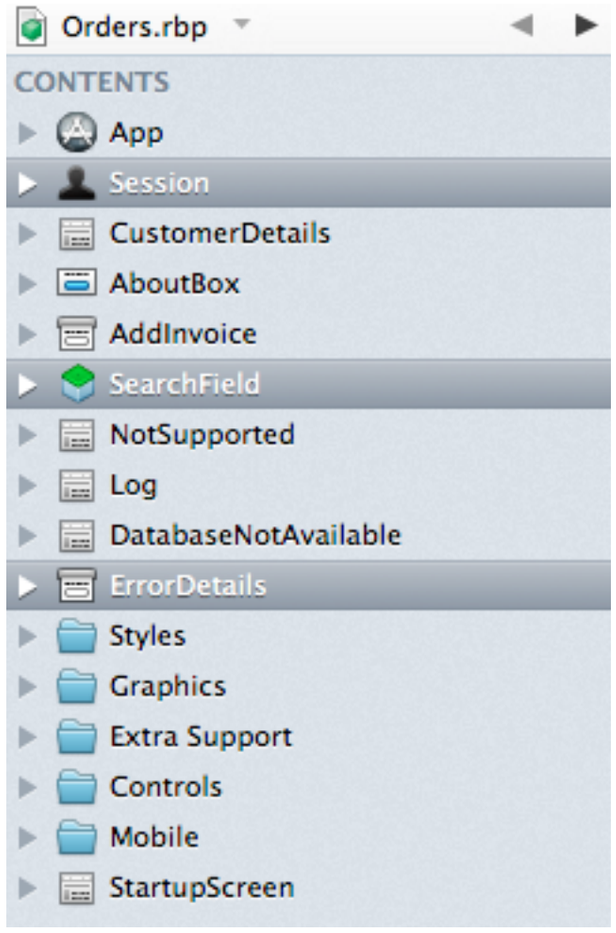
Contents

The Contents section shows the items in your project.

You can click on project items in the Contents section to view them or edit them.

When an item is selected, you can use the arrow keys to move the selection between items. Press return on an item to edit its name. Press return again to apply the change or press ESC to cancel editing.

Figure 2.6 Navigator with Multiple Items Selected



You can also do the usual things such as deleting, copying or pasting project items. You can also drag project items to reorganize them or to move them between project items.

For example, you can drag a method from one class to another class to move it.

In addition, objects can be multi-selected. For example, you can select a Class and a Window, and drag them into a Folder.

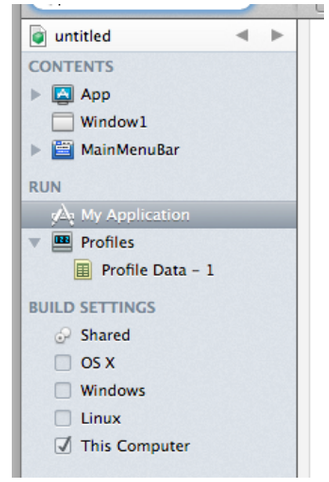
Run

The Run section appears when you run your project. This causes the App name to appear and displays the Debugger.

Should you navigate to another section of your project, clicking on the app name while a project is running displays the debugger.

The Run section is also where profile data appears if you turn on the Profiler.

Figure 2.7 Run Section Displaying App Name and Profiler Data



Build Settings

The Build Settings section is used to view and change the information needed to build and run your project.

The Build section also shows the various build targets available to you. “This Computer” is checked by default and contains the settings for the platform you are currently using.

Shared Settings contain many common settings that were accessible as App class properties in Real Studio. In particular, use the Profile Code property to enable the Profiler.

Check the box next to other targets to have your project built for them next time you build the project. Click on a build target name to change its settings using the Inspector.

The Build section also is where you manage your Build Steps. Builds Steps can be added to the Contents area, but will not run when there. To activate a Build Step, drag it onto the appropriate target. Steps that are before the default “Build” step occur before the project is built and steps after it occur after the project is built.

To disable a Build Step, drag it out of the Target and back into the Contents area.

Build Steps for existing projects are automatically placed in the appropriate target when the project is opened.

Figure 2.8 Build Target Settings with Build Steps

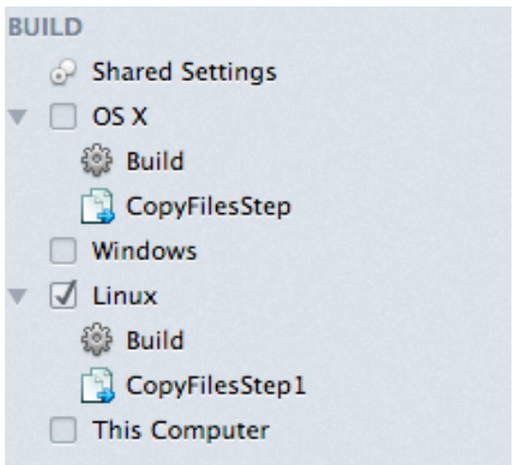
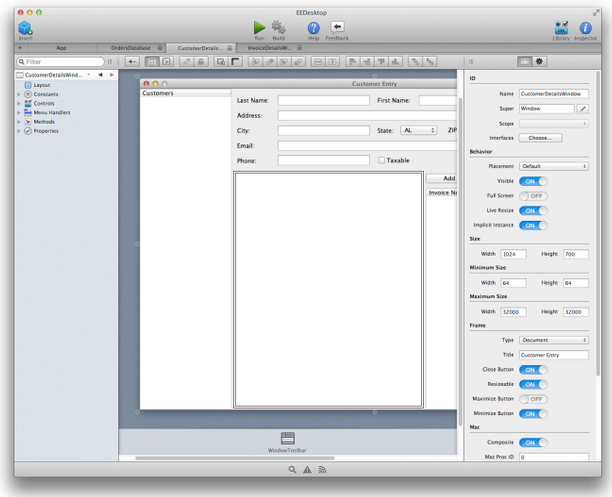


Figure 2.9 Project with Multiple Tabs



Tabs

A tab is simply another view into your project. When you create a new tab, you get new Navigator, Editor and Library/Inspector areas.

You can navigate anywhere you want within a tab, just as you can when there are no tabs. Everything works exactly the same; you just now have multiple views into your project, each of which can show different information.

Tabs can be a great way to keep frequently used items available, particularly when used in conjunction with the Jump Bar.

To open a project item in a tab, you can use the contextual menu and select “Open in New Tab”. You can also use Option+⌘ when double-clicking on a project item to open it in a new tab (on Windows and Linux, use Shift-Control).

Tabs can be locked or unlocked. A locked tab will not have its contents changed when you click on Filter or Search results. In those cases, a new tab (or the next available unlocked tab) is used.

If a project item is already open in a tab, attempting to open it in a tab from a different tab takes you to the tab where it is already open.

Adding Project Items

Use the Insert button on the toolbar or the Insert menu to add new project items.

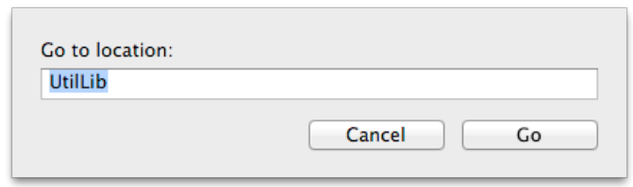
You can also add new project items by dragging a control directly from the Library to the Navigator. This creates a subclass of the control.

Go To Location

If you know its name, you can jump to a specific project item using the Go To Location feature.

Select Project ↠ Go To Location to display the Go To Location window. Enter the name of the project item you want to jump to and press Return (or click Go). The Navigator will select the item.

Figure 2.10 Go To Location Window



Printing

You can print your source code using File ↠ Print from the menu.

When you have project items selected, only the selected items print.

If you do not have anything selected, the entire project prints.

Universal Layout Editor

History

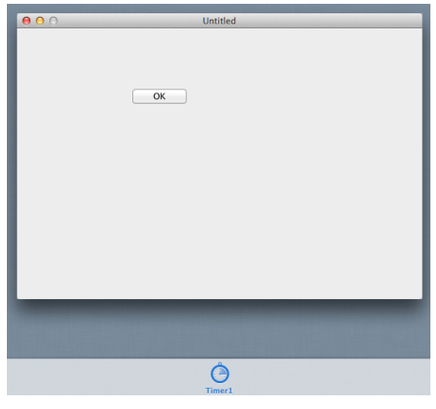
In Real Studio, the Window Editor and the Web Page Editor were two completely different editors. They worked differently and shared almost no code.

With Xojo, these two editors have been combined into a single Universal Layout Editor, which works differently than either the Window or Web Page Editors have in the past.

Shelf

A significant change is the addition of the **Shelf** to the Windows layout editor to keep certain controls organized. It appears on the bottom of the Layout Editor when you add non-visual controls (such as a Timer or Socket), a Toolbar to a window or Dialog to a web page.

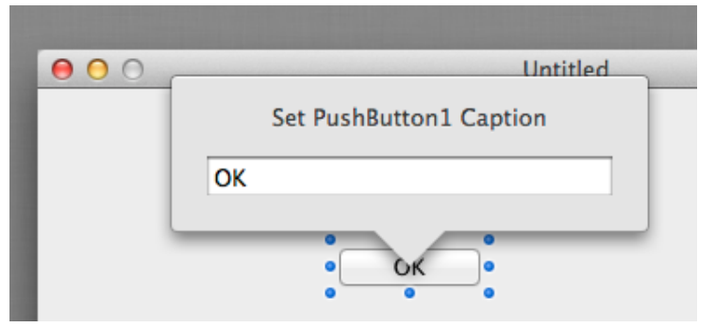
Figure 2.11 Window Editor with Shelf Containing a Timer



Default Values

Controls on the Layout Editor can have their default value specified by pressing *Return* while the control is selected, by clicking the "Pencil" rollover icon  that appears when you move the mouse over a control, or by clicking the "Set Default Value" button on the Layout Editor Toolbar . This opens a pop-out window to enter the default value. For example, with a PushButton, you can specify the Caption. To close the pop-out window, press *Return*, click outside the pop-out window or click the "Set Default Value" button on the Layout Editor toolbar.

Figure 2.12 Setting the Default Value for a PushButton

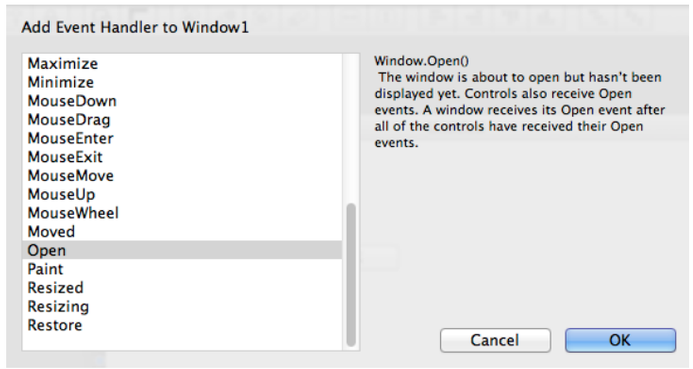


Event Handlers

Event Handlers are now displayed differently. In Real Studio, when you switched to the Code Editor while editing a layout, all the possible event handlers for each control and the window/web page itself were displayed.

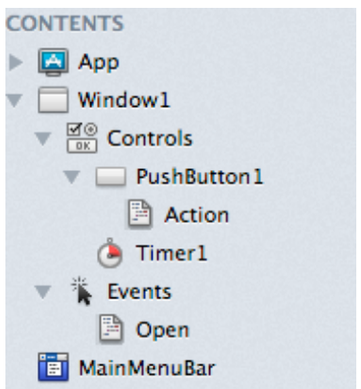
In order to make it easier to implement and find what you are looking for, you now select the event handler you want to implement. Only implemented event handlers appear in the Navigator.

Figure 2.13 Add Event Handler Dialog



To add event handlers, select a window, web page or control and click the "+" button on the window toolbar and choose "Event Handler...". This opens the Add Event Handler dialog where you choose one or more events. As you click on an event you will see the description of it appear on the right. Click OK to add the selected event handlers to the control (or window or web page).

Figure 2.14 Navigator Showing Event Handlers



With this design, only the event handlers you have explicitly added appear in the Navigator and they now appear underneath the control to which they belong. You can quickly switch between the Code Editor for event handlers and the Layout Editor by clicking between the event handler name and the control or by using the "Go to" toggle button on the toolbar.

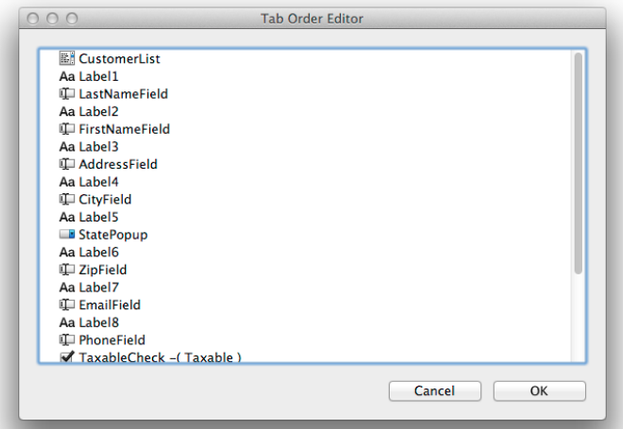
Additional Alignment Guides

As you move controls around on a layout you will see additional alignment guides to help with positioning.

Show Tab Order

Show Tab Order displays each control in the Tab Order Editor. The controls on the layout appear in the editor list in tab order. You can drag controls around to change their tab order on the layout.

Figure 2.15 Tab Order Editor



Fill Height and Width

You can now automatically have a control fill the height and width of its container. Use the Fill Height and Fill Width buttons on the Layout Editor Toolbar  .

Show Measurements

The Show Measurements feature displays the distances of the selected controls from the edgefils of the window or web page.

In conjunction with the alignment guides, this gives you a great way to visually inspect your user interface layout for consistency.

Subclasses

To use subclasses of controls in your layout, drag them from the Navigator to the Layout Editor.

Control Sets

Control Arrays are now known as control sets. They were renamed because they have never been arrays and don’t really work as arrays, which was confusing.

Library and Inspector

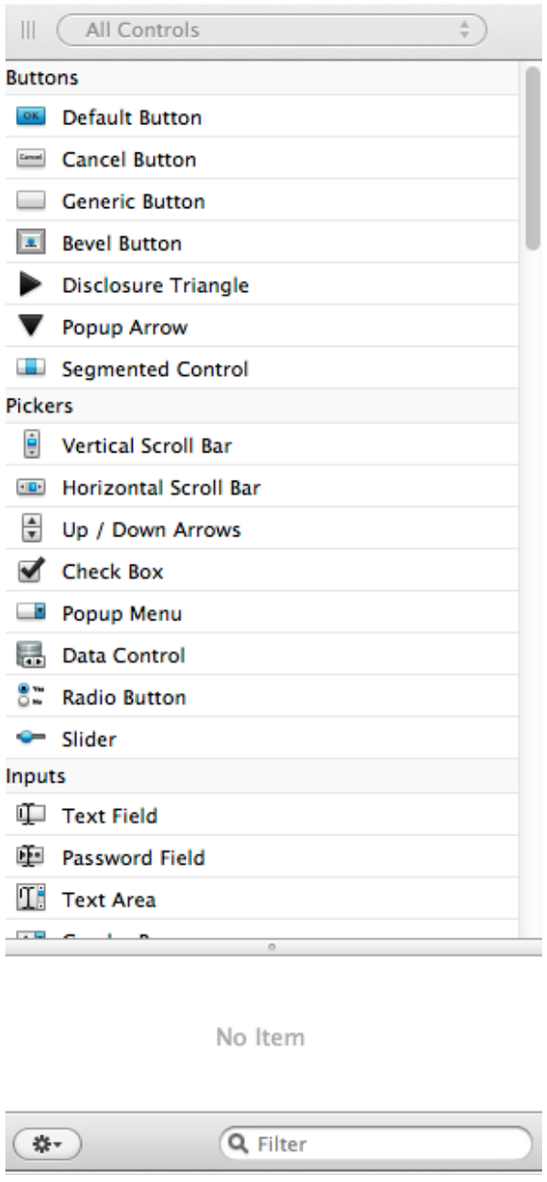
Library

In Real Studio, the Control List (now called the **Library**) was only available when you were editing something that had controls, such as a window or web page. Now the Library is visible everywhere making it easy to add project items and control subclasses to your project!

To show the Library, click the Library button on the toolbar, select View ↠ Library from the menu or press the shortcut key (Command+L on OS X, Ctrl-L on Windows and Linux). The Library displays on the right side of the Workspace by default, but you can also change a preference to have it display as a floating palette.

The Library allows you to group the controls in a variety of ways. By default the controls are shown in large size, but you can also change the size and add group headers. Display settings include: Large Icons, Large Icons with Labels, Small Icons and Labels or Large Icons and Descriptions. Hovering the mouse over any control displays its description in the Description Area at the bottom.

Figure 2.16 Library with Small Icons and Labels



You can also filter the controls. Using the drop-down at the top of the Library you can choose to show only the controls from a specific group. Or you can use the Filter field at the bottom to quickly search for and show controls by name or type. For example, type “button” to see all the button controls.

You can drag controls from the Library onto the Layout Editor if it supports the control you have selected. You can also drag controls directly to the Navigator to immediately create a subclass. Lastly, you can also just double-click the control to add it to either the Navigator (as a subclass) or the Layout Editor, depending on which one has the focus.

Inspector

The Inspector displays information about the current selection. This could be properties of controls on a Layout Editor, project items in the Navigator, Build Settings or method signatures when using the Code Editor.

The Inspector shares its space with the Library. To show the Inspector, click the Inspector button on the toolbar, select View ↠ Inspector from the menu or press the shortcut key (Command+I on OS X, Control-I on Windows and Linux).

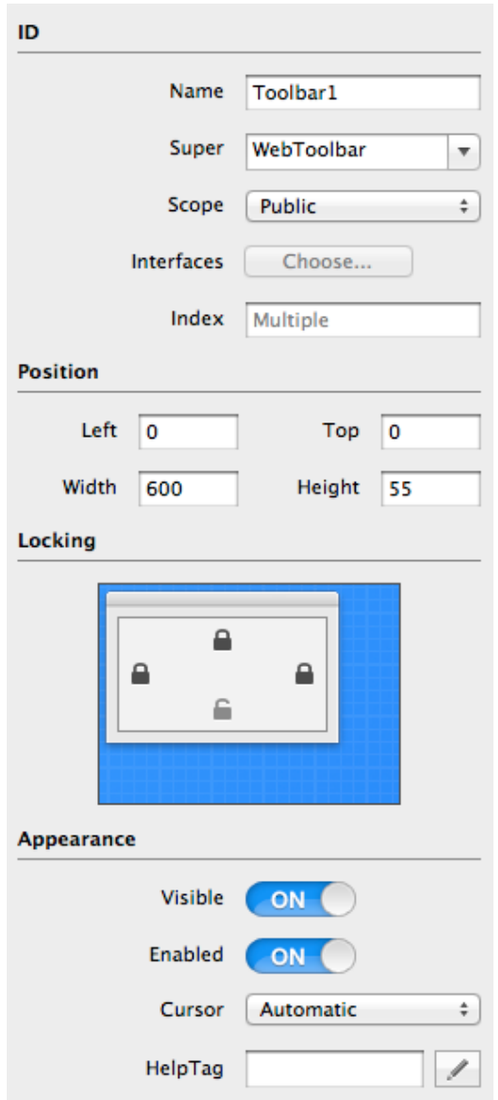
Here are some of the most visible changes when looking at properties in the Inspector:

- The properties themselves have been reorganized and regrouped.
- Locking: The properties for LockLeft, LockRight, LockTop and LockButton are now set visually to help you understand how the locking works. When you click on a lock icon, the graphic resizes to show you how the control will adjust.
- Boolean values: Boolean properties now use an ON/OFF slider instead of a checkbox.

Figure 2.18 Visual Set Control Locking

Figure 2.17 Toggle for Boolean Properties

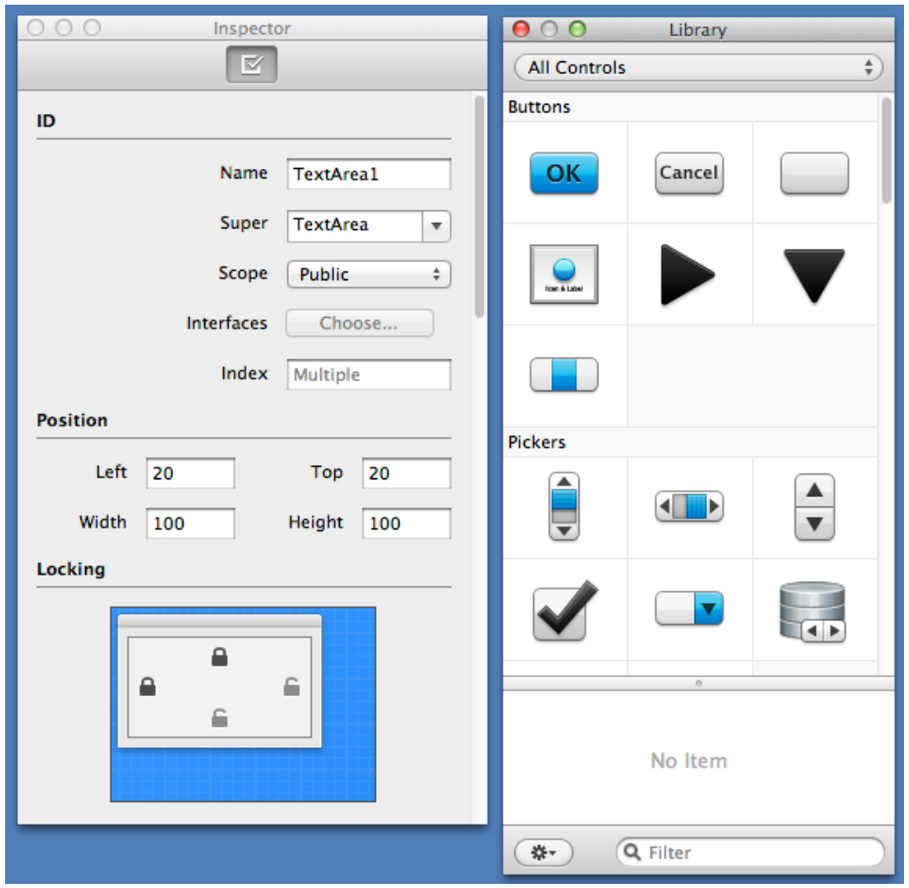
Figure 2.19 Inspector Showing Properties for WebToolbar



Palettes

By default, the Library and Inspector display on the right side of the window and only one appears at a time.

Figure 2.20 Inspector and Library as Palettes



If you would prefer to position the Library and Inspector anywhere and have them both onscreen at the same time, you should use the preference to display them as Palettes.

In the General area of Preferences change “Show Library and Inspector” from “In the project window” to “As floating palettes”.

Panels

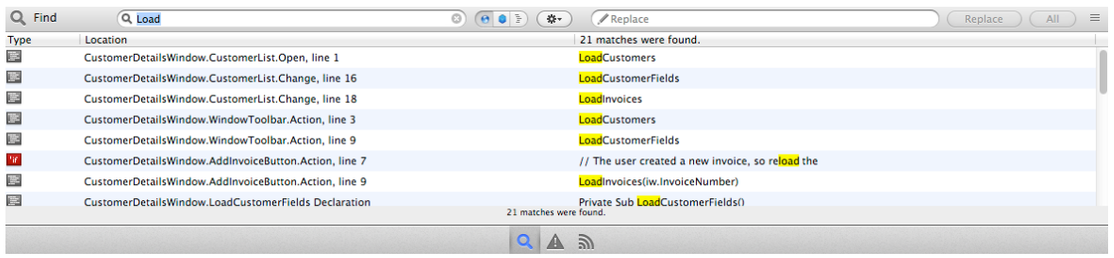
There are three buttons at the bottom of the Real Studio window that open panels for Find, Errors and Messages.

Figure 2.22 Panel Buttons

Find

The Find Panel is used to search (and optionally replace) text in your project. It searches instantly as you type text in the Find field. You can use the “gear” button to change the matching criteria.

Figure 2.21 Find Panel

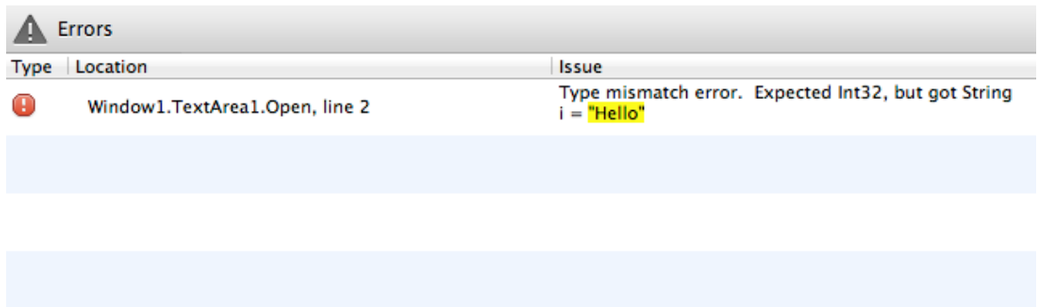


Click once on a Find result to jump to where it is in your project. Find only searches the items displayed in the Navigator (e.g. you have filtered the navigator using either the Filter or Jump Bar).

Errors

The Errors Panel displays compiler errors and warnings. This panel appears automatically when you **Run** or **Build** if there are compiler errors. Click on the issue to jump to where it is in your project.

Figure 2.23 Errors Panel Showing a Compiler Error



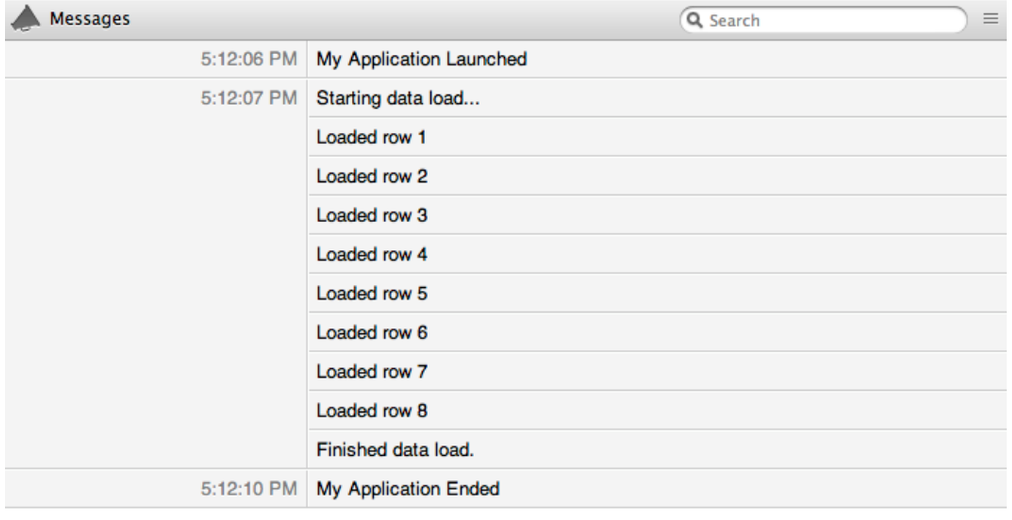
Warnings are only displayed when you use the “Analyze Project” or “Analyze Current Item” commands.

Messages

Messages is a new feature and is primarily used when running your project using the debugger.

When you run your app, messages for Application Launch and Application End are created. Additional system messages may

Figure 2.24 Messages Displaying Text Using System.DebugLog



also be displayed here.

Especially useful is that the output from System.DebugLog now appears in the Messages Panel.

You can search for messages using the search field.

Code Editor

The code editor has been almost completely rewritten for Cocoa support and to improve performance.

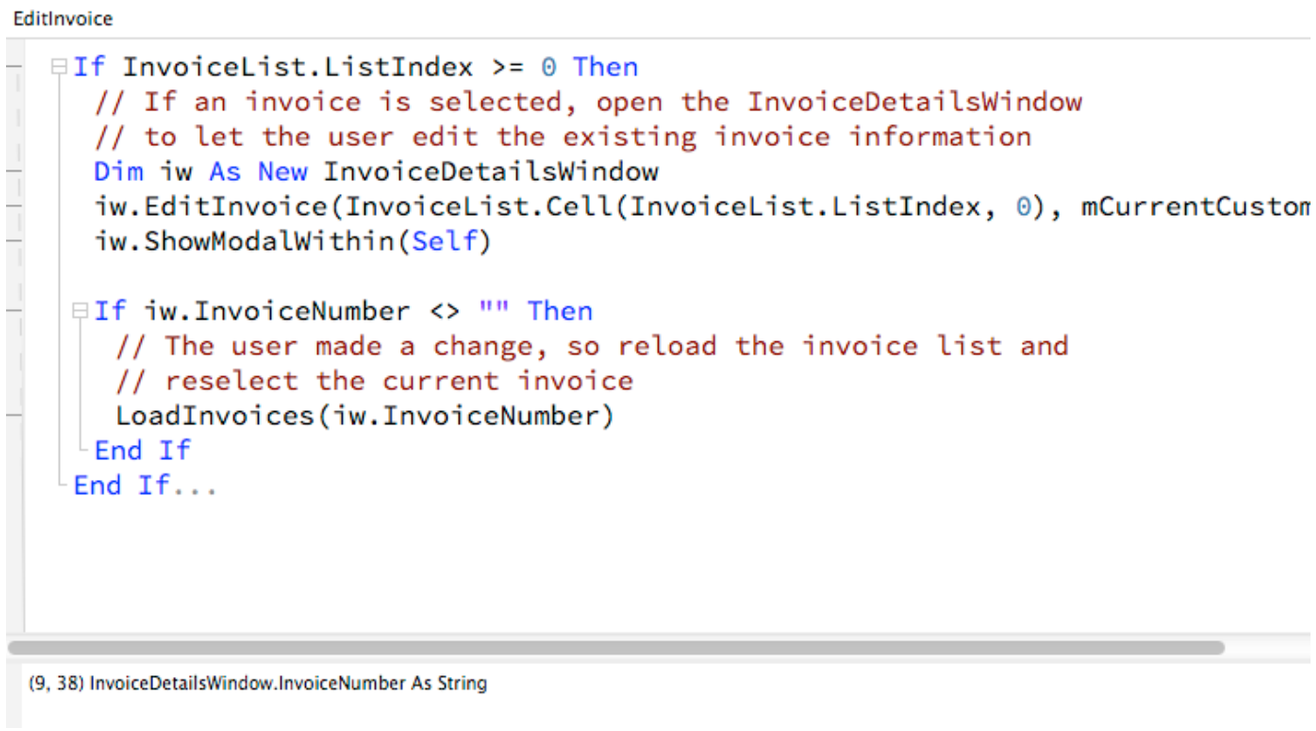
It should largely work the same as before, but there is a new Syntax Help area that replaces how the status bar was used in Real Studio.

Syntax Help

There is a small area at the bottom of the code editor that is used to display syntax help.

The Syntax Help Area displays syntax information, method signatures, declarations and other information about the code that is under the mouse cursor.

Figure 2.25 Syntax Help Area



Section 8

Debugger

When you run your application, a new tab appears with the Debugger.

Should you navigate away from the debugger, simply click on the tab to go back to it.

Stack

The runtime stack now displays as a list.

The debugger has some layout changes. A common complaint with the stack display in Real Studio was that having it in a Popup menu was unintuitive. Now the stack is displayed prominently in a list that is easier to view and navigate.

To the right is the variable watch list, which works the same as before.

Below both of these is the code viewer which lets you see and step through your code.

Message View

At the bottom on the window are three icons. The rightmost icon displays messages. You can display your own messages here by using:

```
System.DebugLog("My message")
```

This is way more useful for debugging purposes than using MsgBox.

In the Messages pane you can search for specific messages using the search field.

Profiler

The Profiler can now be used by all developers.

Profiler results appear in the Navigator in the Run section.

The display has been improved to show you the order in which the methods were called. This allows you to understand the flow of your code as well as how long individual methods took to execute.

Figure 2.27 Profile Results in the Navigator

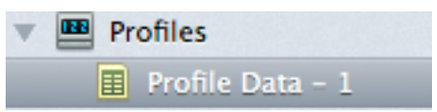


Figure 2.26 Profiler Results

Name	Called	Total (milliseconds)	Average (milliseconds)
▼ App.Open	1	11.0000	11.0000
App.ConnectToDatabase	1	11.0000	11.0000
▶ MainWindow.CancelClose	1	32.0000	32.0000
MainWindow.NameList.CellBackgroundPaint	57	0.0000	0.0000
MainWindow.NameList.Open	1	0.0000	0.0000
▼ MainWindow.Open	1	0.0000	0.0000
MainWindow.ShowSearchPage	1	0.0000	0.0000

Other Changes

- The Profiler now works with Cocoa applications.
- Timing data is now displayed as milliseconds rather than seconds.

Database Editor

When you add a database to your project, it appears in the Navigator (in the Contents section).

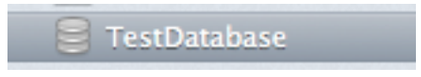


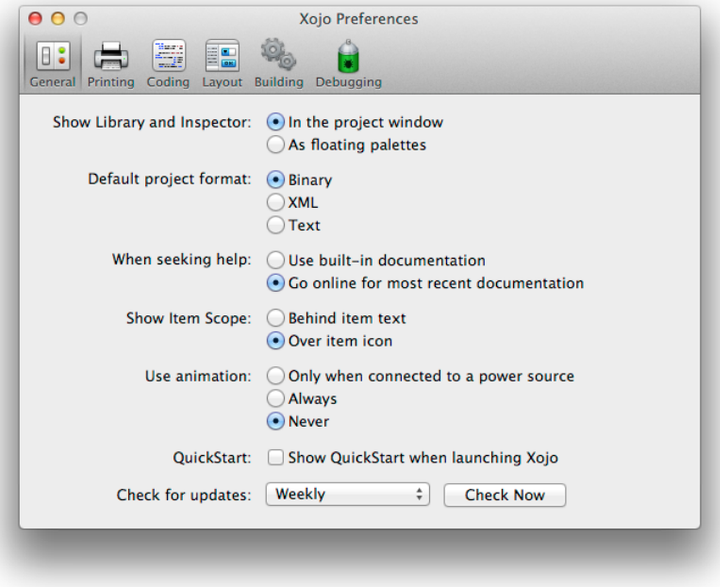
Figure 2.28 Database Displayed in the Navigator

The Database Editor now displays in the Editor area. The Database Editor allows you to add tables and columns.

Preferences

There are two new settings on the General area of Preferences: “Show Library and Inspector” and “Use animation”.

Figure 2.29 Preferences Window



Show Library and Inspector

By default, the Library and Inspector appear on the right-hand side of the Workspace. You can optionally choose to display the Library and Inspector as separate floating palettes.

Project Types, Formats and Templates

Project Formats

The extensions for the project formats have been changed. Your old projects and extensions will continue to work fine, but any new projects that you create will use the new extensions.

You can save your projects in a variety of formats, including the default single-file format (xojo_binary_project), a text-based XML format (xojo_xml_project) and the preferred text-based format (xojo_project).

In Preferences you can change the format that is used by default.

Xojo Project (xojo_binary_project)

This is a binary file format. Your project is stored in a single file (except for external items such as pictures) that is easy to distribute.

XML (xojo_xml_project)

The XML file format is simply an XML representation of the binary file format. It is also a single file, but it is entirely XML. Being XML, this file format can be larger than the same file saved as RBP.

Text (xojo_project)

The Text file format saves your project as separate text files, one for each project item. This file format is ideal for use with version control systems such as Subversion, Git or Mercurial.

This format uses the following extensions for your project items:

- xojo_code
  Contains source code project items such as modules, classes and web pages.
- xojo_window
  Contains windows and container controls from desktop projects.
- xojo_menu
  Contains menus from desktop projects.
- xojo_toolbar
  Contains toolbars from desktop projects.
- xojo_report
  Contains reports from desktop projects.

- xojo_resources
  Contains binary information such as icons, encrypted items and other binary data.

- xojo_uistate
  Contains the UI layout settings such as window positions and sizes.

When used with a version control system, you should add all the files to your repository except for the .xojo_uistate file.

When you delete files or folders from a text project format, they are left on disk so that you can manually handle the deletion in your version control system.

Chapter 3

Other Changes

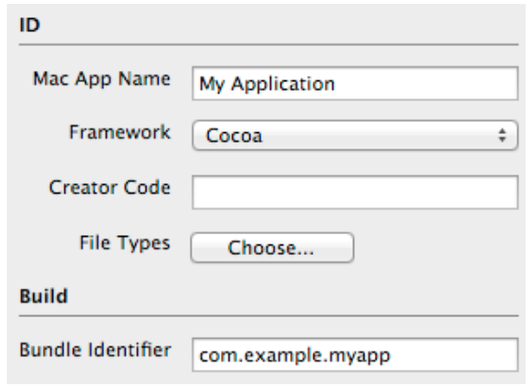
This chapter covers framework and other changes recently made to Xojo.

Cocoa

Default Framework

Starting with Xojo 2013r1, Cocoa is no longer considered “Beta”. Cocoa is now the default for the framework property for OS X desktop apps (in the Build Settings section of the Navigator).

Figure 3.1 Framework Property in OS X Build Settings



Bundle Identifier

Cocoa applications now require a Bundle Identifier. This was known as the Application Identifier in Real Studio. You can change the Bundle Identifier by selecting “OS X” in Build Settings in the Navigator. The Bundle Identifier property appears in the Inspector.

Line Endings

For all OS X applications, Cocoa or not, EndOfLine now returns EndOfLine.Unix instead of EndOfLine.Macintosh.

The change was necessary because Cocoa and any declares would not work properly with EndOfLine.Macintosh. Since Cocoa is now the default framework for OS X, it made sense to make this change now.

If you require the old behavior, you will need to explicitly use EndOfLine.Macintosh.

Threading

Updating the User Interface

Applications can no longer access the user interface from a thread. If your application does access the user interface from within a thread, a ThreadAccessingUIException is raised. You can use this exception to find the problem in your code.

In addition, there is a tool you can use with applications built in prior releases to help you identify areas where threaded UI access may be happening.

Refer to this blog post to learn more:

http://www.xojo.com/blog/en/2013/06/accessing-the-user-interface-from-a-thread.php

For examples on how to change your code to allow UI updates while a thread is running, refer to these example projects in Desktop/UpdateUIFromThread:

- UIThreadingWithTask
- UIThreadingWithTimer

Databases

SQLite

RowID

When using the RealSQLDatabase class for connecting to SQLite, the rowid column was injected into your SQL before it was processed by SQLite. Most of the time this did not cause any issues. But there are times where it could break more complex SQL statements (such as with UNION).

With the new SQLiteDatabase class, the rowid is no longer injected into the SQL. The RealSQLDatabase class (now deprecated) continues to inject the RowID for backwards compatibility.

Transactions

The RealSQLDatabase class required a Commit after you sent any command that changed the state of the database.

With the SQLiteDatabase class, you only need to issue a Commit if you have manually started a transaction by sending the “BEGIN TRANSACTION” command to the database.

You may need to remove unnecessary Commit statements in order to prevent “not in a transaction” database errors.

Backup

SQLite now has support for doing backups.

Backup(destination As SQLiteDatabase, callbackHandler As SQLiteBackupInterface = Nil, sleepTimeInMilliseconds As Integer = 10)

The backup works asynchronously; to receive feedback implement SQLiteBackupInterface and supply it as a parameter.

If you prefer this to work synchronously supply a sleepTimeInMilliseconds of -1.

SQLiteBackupInterface:

- Progress(percent As Double, ByRef cancel As Boolean)
  Percent of completion ranges from 0-100, to cancel the backup set the cancel parameter to True
- Complete()
  Triggered when the backup is finished

- Error(errorCode As Integer)

  Triggered when an error is encountered, currently limited to:

  0 - no error

  1 - source database disconnected

  2 - destination database disconnected

MySQL/PostgreSQL

These database no longer contain the MultiThreaded property. All DB access is now multithreaded by default.

Section 4

Remote Debugger

Remote Debugger Console

There are now two version of the Remote Debugger: Desktop and Console.

The Desktop version is the same as what was in prior versions of Real Studio. The Console version can be used to remote debug console applications, standalone web applications and CGI web applications. Currently it can only be used across a local network (or VPN).

If you are trying to debug a CGI app, set the Remote Debugger to NOT launch automatically. The web browser will launch the app when you go to the appropriate URL.

The Console Remote Debugger runs from Terminal or the command line. The first time you launch it, you are prompted for the settings:

- Machine Name
- Download Directory
- IP Address
- Maximum Connections
- Auto Launch
- Public
- Password

These settings are saved in the RDS.config file in the same folder as the Console Remote Debugger.

You can also provide these options via the command line. Pass “--help” to get a list of available command line options.

Class Name Changes

Although these existing classes will continue to work under their old names, the following classes have new names starting with Xojo:

Old Name	New Name
RealSQLDatabase	SQLiteDatabase
RBScript	XojoScript
RealSQLPreparedStatement	SQLitePreparedStatement
RBVersion	XojoVersion
RBVersionString	XojoVersionString
RBReportControl	XojoReportControl
RBReportDocument	XojoReportDocument
RBScriptAlreadyRunningException	XojoScriptAlreadyRunningException

The old class names are deprecated.

Web

WebListBox
Optimized performance of WebListBox.RowCount.

WebDialog and WebContainer
Open and Shown events have been updated to fix issues with them being called too often or not enough.

WebFileUploader and WebUploadedFile
Now uploads files and requests larger than 256K directly to disk which greatly reduces its memory requirements.

WebFile
Can now load files directly from disk rather than always loading them into memory first.

WebSegmentedControl
You can now add and remove segments from the control at run-time.

WebToolbar
You can now add and remove buttons from a toolbar at run-time.

AddHandler/RemoveHandler
Now work properly with web controls.

Other Changes

JSONItem

Changes were made to how control characters are encoded.

Office Automation

The Office Automation plugin has been separated from Xojo and is no longer loaded automatically. Copy it (from Extra/Office Automation) to your Plugins folder in order to use the Office Automation classes.

If Operator

The If operator behaves much like C/C++'s ternary operator, VB's IIf, or VB.Net's If operator. It takes three parameters: a boolean for the conditional, the value to be returned if the conditional evaluates to true, and the value to be returned if the conditional evaluates to false.

The return type is the common type between the two values. For example, if an Int8 and an Int32 were passed in, the result type is Int32. Having no common type results in a compile error.

For example, this code prints "Big number":

```
Dim myInteger As Integer = 41
MsgBox If(myInteger > 40, "Big number", "Small
number")
```

QuickTime and MoviePlayer

QuickTime

All uses of QuickTime and QTKit have been removed. As part of this, the following have been removed from the framework:

Classes

- EditableMovie
- QT3DAudio
- QTEffect
- QTEffectSequence
- QTGraphicsExporter
- QTGraphicsExporter
- QTSoundTrack
- QTTrack
- QTUserData
- QTVideoTrack
- System.QuickTime

MoviePlayer properties

- MoviePlayer.EditingEnabled
- MoviePlayer.PlayerType
- MoviePlayer.PlaySelection
- MoviePlayer.QTMovieController
- MoviePlayer.QTVRNode
- MoviePlayer.QTVRNodeCount
- MoviePlayer.QTVRPan
- MoviePlayer.QTVRPanMax
- MoviePlayer.QTVRPanMin
- MoviePlayer.QTVRPanTiltSpeed
- MoviePlayer.QTVRTilt
- MoviePlayer.QTVRTiltMax

- MoviePlayer.QTVRTiltMin
- MoviePlayer.QTVRZoom
- MoviePlayer.QTVRZoomMax
- MoviePlayer.QTVRZoomMin
- MoviePlayer.QTVRZoomSpeed
- MoviePlayer.Rate
- MoviePlayer.SelLength
- MoviePlayer.SelStart

MoviePlayer methods

- MoviePlayer.Clear
- MoviePlayer.Copy
- MoviePlayer.Cut
- MoviePlayer.Paste
- MoviePlayer.QTVRHotSpotCount
- MoviePlayer.QTVRHotSpotID
- MoviePlayer.QTVRNodeTypeObject
- MoviePlayer.QTVRNodeTypePanorama
- MoviePlayer.QTVRTriggerHotSpot
- MoviePlayer.QTVRTriggerHotSpotNames
- MoviePlayer.Undo

FolderItem methods

- FolderItem.CreateMovie
- FolderItem.OpenEditableMovie

Global methods

- GetQTCrossFadeEffect
- GetQTGraphicsExporter
- GetQTSMPTEEffect

MoviePlayer

The MoviePlayer control has been rewritten using AVFoundation instead of QTKit on OS X.

Section 9

iOS

Xojo now has the ability to create iOS apps by selecting the iOS project type in the Project Chooser.

Xojo iOS apps can run on iPhones, iPads and related devices. You have to be using OS X in order to create iOS apps with Xojo. You will also need to have Xcode installed (free from the Mac App Store) so that you can get the iOS Simulator (for testing on OS X).

For more information about iOS, refer to the online documentation.

Online Documentation

Section 10

64-Bit

Xojo can now create 64-bit apps for Windows, OS X and Linux. You create a 64-bit app by selecting “x86 64-bit” for the Architecture property in the Inspector for the selected Build Setting.

For more about 64-bit, refer to 64-bit Guidelines:

64-Bit Guidelines

Section 11

Raspberry Pi

Xojo can now create Raspberry Pi Linux apps (ARM). You create a Raspberry Pi app by selecting “ARM 32-bit” for the Architecture property in the Inspector for the Linux Build Setting.

For more about Raspberry Pi, refer to Raspberry Pi Overview:

Raspberry Pi Overview