

Show:  Delphi  C++  
Display Preferences

# What's New in Delphi and C++Builder XE3

## From RAD Studio XE3

Go Up to [What's New](#)

## Contents

- [1 Highlights of the XE3 Release](#)
  - [1.1 New Metro Project Templates and Application Styles for Delphi and C++Builder](#)
  - [1.2 IDE Changes for XE3](#)
    - [1.2.1 Changes in File > New Menu](#)
    - [1.2.2 Changes in Project > Options Dialog Box](#)
    - [1.2.3 Changes in Tools > Options Dialog Box](#)
  - [1.3 Delphi Compiler Changes for XE3](#)
  - [1.4 C++Builder Changes for XE3](#)
    - [1.4.1 Installing Xcode Command Line Tools on Mac OS X 10.7 \(Lion\)](#)
  - [1.5 FireMonkey Changes for XE3](#)
  - [1.6 Debugger Changes for XE3](#)
  - [1.7 Database Changes for XE3](#)
  - [1.8 ITE-ETM Changes for XE3](#)
  - [1.9 LiveBindings Changes for XE3](#)
  - [1.10 RTL Changes for XE3](#)
  - [1.11 VCL Changes for XE3](#)
  - [1.12 Online Help Changes for XE3](#)
  - [1.13 Application Samples Changes for XE3](#)
  - [1.14 SOAP Changes for XE3](#)
  - [1.15 Third Party Changes for XE3](#)

## [\[edit\]](#) Highlights of the XE3 Release

- [New Metro Project Templates and Application Styles for Delphi and C++Builder](#)
- [FireMonkey Enhancements for XE3](#)
- [LiveBindings Enhancements for XE3](#)
- [Entitlement List for Mac OS X Applications](#)

## [\[edit\]](#) New Metro Project Templates and Application Styles for Delphi and C++Builder

- Embarcadero® RAD Studio is preparing for full support of the Windows® 8 Metro® user interface
- See [Developing Metro Style Applications for Windows 8](#)

## [\[edit\]](#) IDE Changes for XE3

## [\[edit\]](#) Changes in File > New Menu

- **New VCL Metro Desktop Application wizards:**
  - **File > New > [VCL Metro Desktop Application - Delphi](#)**
  - **File > New > [VCL Metro Desktop Application - C++Builder](#)**
  - These wizards offer the following choices:
    - Blank Metro Application
    - Grid Metro Application
- **New FireMonkey Metro Desktop Application wizards:**
  - **File > New > [FireMonkey Metro Desktop Application - Delphi](#)**
  - **File > New > [FireMonkey Metro Desktop Application - C++Builder](#)**
  - These wizards offer the following choices:
    - Blank Metro Application
    - Grid Metro Application
- **New FireMonkey Desktop Application wizards:**
  - **File > New > [FireMonkey Desktop Application - Delphi](#)**
  - **File > New > [FireMonkey Desktop Application - C++Builder](#)**
  - These wizards offer the following choices:
    - [HD FireMonkey Application](#)
    - [3D FireMonkey Application](#)

## [\[edit\]](#) Changes in Project > Options Dialog Box

- New [Entitlement List](#) page enables you to select the entitlements you want your Mac OS X applications to have. This is part of provisioning and sandboxing your OS X applications.
- New [Provisioning](#) page allows you to specify your Apple developer certificate and your provisioning profile for your Mac OS X application. This is part of provisioning and sandboxing your OS X applications. These fields are required for submitting an OS X application to the Mac App Store. (The same page occurs in Tools Options.)
- The [Version Info](#) page now has three **Build Options**:
  - **Auto increment build number** has been restored from XE.
  - **Auto generate build numbr** has been retained from XE2.
  - **Do not change build number** has been added.
- The top-level **Directories and Conditionals** page in Project Options has been renamed to [C++ \(Shared Options\)](#). Other than the name change, nothing has changed on this page.

## [\[edit\]](#) Changes in Tools > Options Dialog Box

- New [Provisioning](#) page allows you to specify your Apple developer identity and your provisioning profile, required for developing OS X applications. (The same page occurs in Project Options.)

## [\[edit\]](#) Delphi Compiler Changes for XE3

## [\[edit\]](#) C++Builder Changes for XE3

- **Building Static Packages:** The C++ compiler ([BCC32.EXE](#)) now does a double build in order to separately generate dynamically linked libraries and static packages. Therefore, the following options are no longer needed and are no longer available on the [C++ Linker](#) page of Project Options:
  - Generate import library (-Gi)
  - Generate static package library (.lib) (-Gl)

For more information, see [Building Static Packages](#).

- The top-level **Directories and Conditionals** page in Project Options has been renamed to [C++ \(Shared Options\)](#). No other changes have been made on this page.
- The following new PCH-related option has been added to the [C++ Compiler Advanced](#) page in Project Options:

Minimum contiguous memory allocation block size (in MB)

- The following option has been added to the [C++ Compiler Directories and Conditionals](#) page in Project Options:

Add the Project Dir to Include path

## [\[edit\]](#) Installing Xcode Command Line Tools on Mac OS X 10.7 (Lion)

For C++ OS X development, Xcode is required on the target Mac. A clean install of the Mac OS X 10.7 (Lion) operating system, even with Xcode, does not contain `/usr/include`. Thus for Lion, you also need to install the Xcode command-line tools.

- For instructions on installing the necessary Xcode tools, see [Connecting Your PC to a Mac](#).
- For more information about OS X development, see [Mac OS X Application Development](#).

## [\[edit\]](#) FireMonkey Changes for XE3

Support has been added for the following features in FireMonkey:

- **Actions** - FireMonkey now supports actions and action lists, features that were previously supported only in VCL:
  - VCL and FMX [framework-independent action features are implemented in the RTL](#), in the following units:
    - [System.Classes](#) ([TBasicAction](#) and [TBasicActionLink](#) classes)
    - [System.Actions](#) ([TContainedAction](#), [TContainedActionLink](#), and so forth)

**Important:** Every VCL or FireMonkey application that uses actions must specify the [System.Actions](#) and [System.Classes](#) units in the `uses` section (or `#include` for C++). For more information, see [Changes in Implementation of VCL Actions](#) and [Implementation of Actions in FireMonkey and VCL](#).
  - FireMonkey framework-specific action features are implemented in [FMX units](#) such as:
    - [FMX.ActnList](#)
    - [FMX.StdActns](#)
  - Using the [Action List editor](#) (same as with VCL), you can group FMX actions in [FMX action lists](#) and [assign actions from lists to UI elements](#).
  - FMX actions cannot be managed with the [VCL Action Manager editor](#).
- **Anchors** - Anchored controls "stick to" the sides of containers, and also stretch, if so specified. You can use anchors to [arrange FireMonkey controls](#).
- **Audio-video**
  - FireMonkey offers support for capturing media data (audio and video). For this purpose the following classes are introduced:
    - [TCaptureDevice](#), which is the base class for capturing devices.
    - [TCaptureDeviceManager](#), which offers the possibility to access and manage capturing devices.

- FireMonkey offers support to play media files (audio and video). The [TMedia](#) class is introduced to support playback for media sources.
  - For easy access, the nonvisual component [TMediaPlayer](#) is introduced.
  - For displaying video files, the [TMediaPlayerControl](#) component is introduced.
- **Layout management:** new [FireMonkey layouts](#) simplify the [arrangement of controls](#) in a FireMonkey application.
  - [Flow layout](#)
  - [Grid layout](#)
 For more details about **layout management**, see [FireMonkey Layouts Strategies](#).
- **Text layout**
- **FireMonkey 3D enhancements**
  - A new shader compiler allows you to create native shaders for different platforms from one hlsl source. DirectX is used to create DX9 and DX10 version of shaders and NVideo cg-toolkit to create ARB (OpenGL asm) and GLSL version of shaders.
  - A new [context shader](#) creates and defines FireMonkey [filters](#), [materials](#) and [context for 3D objects](#).
  - The new [FMX.Types3D.TTexture](#) class replaces the use of bitmap textures. This class uses GPU memory to store real data. [TTexture](#) is a limited class, but offers support for textures with different pixel formats, behavior and styles.
  - The new **material system** is based only on shaders, and allows an unlimited variation of lighting and materials in one application. A [material](#) is linked to 3D objects through the [TMaterialSource](#) object that can be positioned anywhere in a form. The existing material can be found in [FMX.Matrics](#) unit and at design time in [Tool Palette](#) under the **Materials** category.
  - Regarding **importing 3D models**, the textures are correctly imported. For more indications about importing 3D models, see the [Importing a 3D Model in a FireMonkey Application](#) tutorial.
- **Gestures:** FireMonkey now supports the gestures that are also supported by the VCL:
  - The [standard gestures](#)
  - Standard Windows 8 gestures for [Metro style applications](#).

For descriptions of the Windows 8 gestures, see <http://msdn.microsoft.com/en-us/library/windows/apps/hh761498.aspx>

FireMonkey does not yet support custom gestures.

- **Styles**
  - FireMonkey styles can now be built in the [Bitmap Style Designer](#).
  - FireMonkey supports Metro style applications. See the [Applying Metro Styles](#).
  - New unit: [FMX.Styles](#), which contains [TStyle Manager](#)
- New units have been added to FireMonkey:
  - [FMX.Materials](#) (used for 3D objects)
- **FireMonkey Sensor Components:**
  - Non-visual components for using device sensors have been added, and you can see them in the **Sensors** tab of the [Tool Palette](#):
    - The Location Sensor (for both Windows and Mac OS X)
    - The Motion sensor (for Windows)
- Some platform-neutral FireMonkey types and math functions have been moved from FMX to the RTL (in the System unit). For details, see [#RTL Changes for XE3](#).
- Virtual keyboard is now supported.
- DirectX 10 is now supported.

## [\[edit\]](#) Debugger Changes for XE3

## [[edit](#)] Database Changes for XE3

- You can build C++ 64-bit Windows database related applications with C++Builder.
- [Support has been added for SQLite databases](#) through the new **SQLite driver**.
- Easier interactions are supported with InterBase ToGo Databases using a new dbExpress driver: [IBToGo Driver](#).
- **TSQLMonitor** is available for [ODBC](#) and [SQLite](#).
- **DBX driver metadata** has been extended to support querying for whether a vendor supports schemas.
- `VendorLib`, `LibraryName`, and `GetDriverFunc` properties are no longer published on **TSQLConnection** and should not be used.
- New `FailIfMissing` property is introduced and determines whether the database connection fails if the database does not exist.
- DataSnap related **DSSession** related functionality is now available in the [Dataspn.DSSession](#) unit instead of former **Dataspn.DSService**.

## [[edit](#)] ITE-ETM Changes for XE3

## [[edit](#)] LiveBindings Changes for XE3

- Introduction of the new visual [LiveBindings Designer](#) -- this is the main tool you use when creating [LiveBindings](#).
- The [LiveBindings Wizard](#) was introduced.
- Added the possibility to **create data sources** ([TPrototypeBindSource](#) or [TBindSourceDBX](#)) from within the [LiveBindings Wizard](#).
- Using a [TPrototypeBindSource](#), now you can bind multiple properties of different objects to the same data.
- A set of [Quick Binding Components](#) components have been introduced in order to make [LiveBindings](#) links seamless. These Quick Binding Components produce auto-generated expressions for easy linking objects.
- LiveBindings can now be created from one control to multiple controls, seamlessly via the LiveBindings Designer.
- Added [LiveBindings menu](#) in [Object Inspector](#).
- [TBindNavigator](#) has been added for cycling through records in a dataset when developing VCL applications.
- Added new LiveBindings components in the palette: [TPrototypeBindSource](#), [TBindSourceDBX](#)
- LiveBindings now **supports TGrid** components.
- Removed the old style of creating LiveBindings (such as **Link to DB Field**, or manual bindings and manual notifications).

## [[edit](#)] RTL Changes for XE3

- Some platform-neutral FireMonkey types and math functions have been moved from FMX to the RTL (in the [System](#) unit):
  - From FMX.Types: [Vector](#)
  - New: [tagVECTOR](#), [tagVECTOR3D](#)
  - From FMX.Types3D: [Vector3D](#), [TVector3D](#)
    - Methods in FMX.Types3D that work with TVector3D have been refactored. For example, **FMX.Types3D.Vector3DAdd** is now [System.Types.TVector3D.AddVector3D](#).

- From FMX.Types3D: [TMatrix3D](#)
- From FMX.Types3D: [TQuaternion3D](#)
- From FMX.Types3D: [Point3D](#)
- Some newly documented methods:
  - [InflateRect](#), [IntersectRectF](#), [MultiplyRect](#)
  - [NormalizeRect](#), [NormalizeRectF](#), [UnionRectF](#)
  - [RectCenter](#)
  - [ScalePoint](#)
- Three new helper types ([TSingleHelper](#), [TDoubleHelper](#), and [TExtendedHelper](#)) replace (and deprecate) the following record types: [TSingleRec](#), [TDoubleRec](#), [TExtendedRec](#).

## [\[edit\]](#) VCL Changes for XE3

- **VCL Actions:** Framework-neutral action features (supported in both FMX and VCL) have been moved from VCL to the RTL:
  - Some action-related functionality has been moved into the following new **System** unit:
    - [System.Actions](#)
      - [System.Actions.TActionListEnumerator](#), moved from [Vcl.ActnList](#)
      - [System.Actions.TContainedAction](#), moved from [Vcl.ActnList](#)
  - The following unit in the RTL contains action-related functionality used by both VCL and FireMonkey:
    - [System.Classes](#)
      - [System.Classes.TBasicAction](#)
      - [System.Classes.TBasicActionLink](#)

**Important:** In VCL applications that use actions, you now need to ensure that the [System.Actions](#) and [System.Classes](#) units appear in the `uses` or `#include` section of your application.
- **VCL Styles:**
  - The VCL Style Designer has been renamed [Bitmap Style Designer](#).
  - Styles are now supported for individual elements of a control. See:
    - [Vcl.Controls.TControl.StyleElements](#)
    - [Vcl.Controls.TStyleElements](#)
  - Standard border styles are now supported. See:
    - [Vcl.Themes.TStyleManager.TFormBorderStyle](#)
- **Animated fade effects** are now supported in VCL controls such as `TButton` and `TBitBtn`. See:
  - [Vcl.Themes.TStyleManager.AnimationOnControls](#)

## [\[edit\]](#) Online Help Changes for XE3

## [\[edit\]](#) Application Samples Changes for XE3

## [\[edit\]](#) SOAP Changes for XE3

## [\[edit\]](#) Third Party Changes for XE3

Retrieved from "[http://docwiki.embarcadero.com/RADStudio/XE3/en/What%27s\\_New\\_in\\_Delphi\\_and\\_C%2B%2BBuilder\\_XE3](http://docwiki.embarcadero.com/RADStudio/XE3/en/What%27s_New_in_Delphi_and_C%2B%2BBuilder_XE3)"

Categories: [Delphi](#) | [C++](#) | [XE3](#)

- [Help Feedback](#)

- Copyright(C) 2012 Embarcadero Technologies, Inc. All Rights Reserved.
- [Current Wiki Page](#)