FireMonkey New Features and Enhancements

Go Up to FireMonkey

Contents

Metal Drivers GPU Support

On the macOS platform you can now build applications against the Metal API, rather that the old Quartz API and OpenGL.

Since Apple deprecated OpenGL, even when it is still available in Catalina, it is likely that will not be available in the future. FireMonkey Metal support offers a smooth migration to future requirements and better performance for screen rendering.

RAD Studio 10.4 introduced support for the Metal API in the iOS platform as well, where it is not required, but can offer better performance for FireMonkey applications in general.

How to Work with Metal

There are a few global settings to enable the use of the Metal driver for UI rendering and some of its configuration.

These settings are below and the default value in indicated:

FMX.Types.GlobalUseMetal: Boolean=False;

It allows using Metal for UI rendering. Boolean variable that specifies whether to use the Metal context for rendering graphics.

To enable the Metal context you have to set the global variable FMX.Types.GlobalUseMetal to True in the initialization section. By default this global variable is False to stay compatible with previous Delphi version.

> *Note: When you use custom OpenGL functions or custom OpenGL Shader in its own code, migration to Metal may no be totally transparent.*

FMX.Types.GlobalEventDrivenDisplayUpdates: Boolean=True; When this value is set to True, the draw loop pauses and updates are event-driven. The default value is True.

This is useful for games, but currently has no equivalent in OpenGL on Android.

> *Note: This setting is for Metal only.*

FMX.Types.GlobalPreferredFramesPerSecond: integer=60; It is the rate where the draw loop updates its contents. Integer variable to set the rate at which the draw loop update its contents when GlobalEventDrivenDisplayUpdates is set to False.

> *Note: This setting is for Metal only.*

New Window Styled Memo Implementation

the new implementation for the styled TMemo component on the Windows platform has better support for IME and other improvements.

To activate this new implementation you have to add to the uses statement the new unit FMX.Memo.Style.New. FireMonkey uses the old implementation for styled TMemo as default in all platforms.

Adding the new unit FMX.Memo.Style.New automatically replaces the old implementation of styled memo for all TMemo controls in the entire application.

The implementation of this new style memo for 10.4 is limited to the Windows platform. Here are some of the key features available:

- The new styled memo includes a completely redesigned IME input. The new styled memo now works like a native IME text input in Windows:
    - The user may enters IME text, to choose where to insert the IME text.
    - While the user is inputting IME text, IME text is not a part of Memo lines. IME text is drawn over the top.
    - Now the IME window changes its position, when the user scrolls the content of the memo.

    - When inputting IME text, the user can apply the Left and Right buttons for making corrections in current IME Text.

- The new styled memo supports new ways for selecting text. For example: you can double click and hold the mouse button down while moving it for selecting multiple words in the text (selecting entire words).

- New styled memo implementation consists of different parts: selection, layout, check spelling, context menu, etc.

- There are new additional methods in TTextService for better controlling IME state.

Additiontal FireMonkey Features and Notable Fixes

- The Toolbar component in FireMonkey now has Hint, ShowHint, and ParentShowHint properties.

- Improved support for Emoji ZWJ Sequences, for example, 👨‍👩‍👧‍👦 Family: Man, Woman, Girl, Boy.

- New method to access Caret position from mouse in TMemo.

- There is a new method (TAndroidHelper.ShouldNativeLibrariesBeExtracted) to return the value of the 'extractNativeLibs' manifest attribute, which returns False for an application packaged as an app bundle.

- Significantly increased the performance of rendering chars in TextLayout.

- On macOS, there is a new entitlement option for Bluetooth.

New TBufferedLayout Component

FireMonkey now has a new layout control that stores its graphical content (including its internal controls) in a memory buffer, so in case there is no change, it can be repainted by displaying the buffer rather than repainting each internal control.

In many cases the net effect is faster refresh of the UI of the control. However, the use of TBufferedLayout in large surfaces may have a negative consequence in terms of memory usage, and that is why this behavior is not part of the core library.

The new control is easy to use like other layouts, and has the following icon:

See Also

FMX.Types.GlobalUseMetal

FMX.Types.GlobalEventDrivenDisplayUpdates

FMX.Types.GlobalPreferredFramesPerSecond

Retrieved from "http://docwiki.embarcadero.com/RADStudio/Sydney/e/index.php?title=FireMonkey_New_Features_and_Enhancements&oldid=271417"

This page was last edited on 22 May 2020, at 00:02.