# Photino.Native & Photino.NET V2

Release Notes

Contents

[Photino.Native & Photino.NET V2 1](#_Toc78800110)

[What’s New 3](#_Toc78800111)

[New Properties 3](#_Toc78800112)

[New Methods 4](#_Toc78800113)

[Windows 4](#_Toc78800114)

[Linux 4](#_Toc78800115)

[Mac 4](#_Toc78800116)

[Bugs Fixed 5](#_Toc78800117)

[Windows 5](#_Toc78800118)

[Linux 5](#_Toc78800119)

[Mac 5](#_Toc78800120)

[Breaking Changes 6](#_Toc78800121)

[Renamed Properties 6](#_Toc78800122)

[Renamed Methods 6](#_Toc78800123)

[Deprecated Properties 6](#_Toc78800124)

[Deprecated Methods 6](#_Toc78800125)

[Other Changes 6](#_Toc78800126)

[Known Issues 7](#_Toc78800127)

[Windows 7](#_Toc78800128)

[Linux 7](#_Toc78800129)

[Mac 7](#_Toc78800130)

[New and Overhauled Samples 8](#_Toc78800131)

[New VS Project Templates and dotnet CLI / VS Code Samples NuGet Package 9](#_Toc78800132)

[Blazor Support 10](#_Toc78800133)

## What’s New

* Improved .NET Intelli-sense documentation
* Improved organization of the source code
* Improved ability to add new startup parameters and features in the future
* Improved development experience for configuration for Photino.NET users
* Multi-window support
* Programmatically close native windows
* Most properties can be set before and/or after native window initialization
* Extended fluent methods for initialization and ability to call them after initialization as well
* Handlers can now be established prior to initializing the native window
* Native browser control’s context menu and developer tools access can now be set in code
* New setting to automatically enable permissions for browser control access to local hardware such as camera and microphone without prompting the user
* New settings to use OS default window position and size for native window
* Programmatically set the native browser control’s Zoom level
* Minimize and maximize native windows programmatically
* Center native windows before and/or after initialization
* Chrome-less native windows allows developers to create custom title bars, footers and menus
* Improved support for Full Screen (Kiosk) mode
* Improved support for Topmost (Always on Top) native windows
* Improved logging
* Ability to retrieve values from native windows as well as set them via expanded set of properties

### New Properties

* Centered (bool)
* Chromeless (bool)
* ContextMenuEnabled (bool)
* DevToolsEnabled (bool)
* FullScreen (kiosk mode) (bool)
* GrantBrowserPermissions (bool)
* IconFile (string)
* Maximized (bool)
* Minimized (bool)
* StartupString (string)
* StartupUrl (string)
* TemporaryFilesPath (string)
* Topmost (bool)
* UseOSDefaultLocation (bool)
* UseOSDefaultSize (bool)
* WebMessageReceivedHandler (EventHandler<string>)
* WindowClosingHandler (NetClosingDelegate)
* WindowCreatingHandler (EventHandler)
* WindowCreatedHandler (EventHandler)
* WindowLocationChangedHandler (EventHandler<Point>)
* WindowSizeChangedHandler (EnventHandler<Size>)
* Zoom (int)

### New Methods

* RegisterWindowCreatingHandler()
* RegisterWindowCreatedHandler()
* RegisterWindowClosedHandler()
* SetChromeless()
* SetContextMenuEnabled()
* SetDevToolsEnabled()
* SetGrantBrowserPermissions()
* SetHeight()
* SetLeft()
* SetSize()
* SetLocation()
* SetLogVerbosity()
* SetTemporaryFilesPath()
* SetTitle()
* SetTopMost()
* SetWidth()
* SetZoom()
* SetUseOsDefaultLocation()
* SetUseOsDefaultSize()

### Windows

* The Edge browser control’s temporary files are created in AppData\Local by default and the location can now be set in code.

### Linux

* n/a

### Mac

* n/a

## Bugs Fixed

* WindowClosing event now works properly
* Fixed several issues related to custom schemes (xxx://)

### Windows

* Memory leak injecting JavaScript code
* Now supports multiple windows
* Fixed pixelated window and taskbar icons

### Linux

* Fixed crash when creating multiple windows

### Mac

* Fixed crash when calling SetIconfile()
* Fixed crash on closing

## Breaking Changes

In order to solve some technical issues, we had to make a few breaking changes in V2. The biggest change has been how PhotinoWindow is initialized. The constructor now takes a single parameter; an instance of the parent window if one exists. Properties and/or fluent methods are used configure nearly every setting prior to initialization of the native window and browser control. The only property the developer is required to set is either StartUrl or StartString allowing the native browse control to immediately render the UI. This requirement is necessary to enable multiple window support in Windows as well as provide a better experience for developers. The native window and browser control are automatically initialized and displayed when WaitForClose() is called. Nearly all properties and fluent methods can also be called after initialization of the native window and browser control to change settings at runtime, providing a consistent developer experience.

### Renamed Properties

* IsOnTop 🡪 Topmost

### Renamed Methods

* FullScreen() 🡪 SetFullScreen() (or use FullScreen property)
* Hide() 🡪 Minimize() (or use Minimized property)
* UserCanResize() 🡪 SetResizable() (or use Resizeable property)

### Deprecated Properties

* Children
* WasShown

### Deprecated Methods

* AddChild()
* Dispose()
* RemoveChild()
* Show()

### Other Changes

* PhotinoWindow no longer implements IDsposable
* PhotinoWindowOptions initialization parameter has been removed. Use XXXHandler properties and/or ResgieterXXHandler() fluent methods and/or Events directly instead. Use of RegisterXXHandler() methods is preferred.

## Known Issues

* ShowState() doesn’t pop up in Alert window.

### Windows

* n/a

### Linux

* Access to local hardware from native browser control is not yet supported by the browser control (out of our control)
* Reading the minimized (iconified) state of a native window doesn’t work in GTK 3 (out of our control)
* Calling GetSize() and SetSize() in rapid succession does not work properly. E.g. :
  + .Height = .Height + 5;
  + .Width = .Width + 5;

### Mac

* SendWebMessage() is not working (root cause of custom schemes issue as well?)
* .NET Application doesn’t exit when all native windows are closed
* Closing the main window does not close child windows
* Center property and method make window ‘prominent’ per macOS / Cocoa standards and don’t exactly center the window (not a bug)
* Access to local hardware from native browser control is not yet supported by the browser control (out of our control)
* Visual error messages from .Native app not implemented as they are in Windows and Linux
* UseOsDefaultLocation sets location to 0,0 as macOS has no default window location (not a bug)
* UseOsDefaultSize is 800x600 as macOS has no default window size (not a bug)
* ContextMenuEnabled cannot be toggled after initialization as it’s not supported in WKWebKit (out of our control)
* DevToolsEnabled cannot be toggled after initialization as it’s not supported in WKWebKit (out of our control)
* GetScreenDpi always return 72 as it’s not supported on macOS (out of our control)

## New and Overhauled Samples

## New VS Project Templates and dotnet CLI / VS Code Samples NuGet Package

## Blazor Support

After carefully reviewing .NET 6, MAUI (Xamarin forms .Next) and related technologies, the Photino team has decided to not pursue Blazor support in V2.