



API Reference Manual

Version 1.2

Contents

1	Namespace Index	1
1.1	Packages	1
2	Class Index	3
2.1	Class List	3
3	Namespace Documentation	5
3.1	Package UWK	5
3.1.1	Enumeration Type Documentation	6
3.1.1.1	Source	6
3.1.2	Function Documentation	6
3.1.2.1	LOGCB	6
3.1.2.2	PROCESSCB	6
3.1.2.3	SUCCESSCB	7
4	Class Documentation	9
4.1	UWK.BridgeEventArgs Class Reference	9
4.1.1	Detailed Description	9
4.1.2	Member Data Documentation	9
4.1.2.1	Args	9
4.1.2.2	MethodName	9
4.2	UWK.BridgeObject Class Reference	10
4.2.1	Detailed Description	10
4.2.2	Member Function Documentation	10
4.2.2.1	Bind	10
4.2.2.2	Invoke	10

4.2.3	Member Data Documentation	11
4.2.3.1	Name	11
4.2.3.2	Properties	11
4.3	UWK.Command Struct Reference	11
4.3.1	Detailed Description	12
4.3.2	Member Function Documentation	12
4.3.2.1	GetSPParam	12
4.3.2.2	Init	12
4.3.2.3	NewCommand	12
4.3.2.4	Post	13
4.3.2.5	SetSPParam	13
4.3.2.6	SpanSPParams	13
4.3.3	Member Data Documentation	13
4.3.3.1	fourcc	13
4.3.3.2	id	13
4.3.3.3	iParams	13
4.3.3.4	numIParams	13
4.3.3.5	numSPParams	13
4.3.3.6	retCode	13
4.3.3.7	src	13
4.4	UWK.CommandHandler Class Reference	14
4.4.1	Detailed Description	14
4.5	UWK.CommandProcessEventArgs Class Reference	14
4.5.1	Detailed Description	14
4.6	UWK.Plugin Class Reference	15
4.6.1	Detailed Description	16
4.6.2	Member Function Documentation	16
4.6.2.1	AllocateString	16
4.6.2.2	ClearCommands	16
4.6.2.3	GetBytes	16
4.6.2.4	GetString	16
4.6.2.5	Init	16
4.6.2.6	PostCommand	16
4.6.2.7	ProcessCommand	16

4.6.2.8	ProcessCommand	16
4.6.2.9	Shutdown	16
4.6.2.10	Update	17
4.6.2.11	UpdateTexture	17
4.6.3	Event Documentation	17
4.6.3.1	ProcessInbound	17
4.6.3.2	ProcessReturn	17
4.7	UWK.Process Class Reference	17
4.7.1	Detailed Description	18
4.7.2	Member Function Documentation	18
4.7.2.1	Go	18
4.7.2.2	KillIt	18
4.7.2.3	Stop	18
4.8	UWK.SubBuffer Class Reference	18
4.8.1	Detailed Description	19
4.9	UWK.TextureInterop Struct Reference	19
4.9.1	Detailed Description	19
4.10	UWK.TextureSet Class Reference	19
4.10.1	Detailed Description	20
4.10.2	Member Function Documentation	20
4.10.2.1	Release	20
4.10.2.2	Update	20
4.11	uWebKitStandard Class Reference	20
4.12	UWKCore Class Reference	20
4.12.1	Detailed Description	22
4.12.2	Member Function Documentation	22
4.12.2.1	ClearCookies	22
4.12.2.2	ClosePopup	22
4.12.2.3	CreateView	22
4.12.2.4	GetView	22
4.12.2.5	Init	22
4.12.2.6	Init	22
4.12.2.7	RemoveView	22
4.12.3	Property Documentation	22

4.12.3.1	ProcessUp	23
4.13	UWKKeys Class Reference	23
4.13.1	Detailed Description	27
4.14	UWKPopup Class Reference	27
4.14.1	Detailed Description	28
4.15	UWKView Class Reference	28
4.15.1	Detailed Description	31
4.15.2	Member Function Documentation	31
4.15.2.1	Back	31
4.15.2.2	BringToFront	31
4.15.2.3	captureTexture	32
4.15.2.4	DrawGUI	32
4.15.2.5	DrawTextIME	32
4.15.2.6	EvaluateJavaScript	32
4.15.2.7	Forward	32
4.15.2.8	Hide	32
4.15.2.9	InvalidateAllViews	32
4.15.2.10	LoadHTML	32
4.15.2.11	LoadTextAssetHTML	32
4.15.2.12	LoadURL	33
4.15.2.13	OnGUIMouse	33
4.15.2.14	OnWebGUI	33
4.15.2.15	ProcessKey	33
4.15.2.16	PushToBack	33
4.15.2.17	Remove	33
4.15.2.18	RestartAllViews	33
4.15.2.19	Show	33
4.15.2.20	Show	33
4.15.2.21	UpdateView	33
4.15.3	Member Data Documentation	34
4.15.3.1	ContentWidth	34
4.15.3.2	Icon	34
4.15.3.3	JSPopup	34
4.15.3.4	Name	34

4.15.3.5	ResizeToContents	34
4.15.3.6	Title	34
4.15.3.7	TSet	34
4.15.3.8	URL	34
4.15.4	Property Documentation	34
4.15.4.1	Active	35
4.15.4.2	AlphaMask	35
4.15.4.3	Height	35
4.15.4.4	MainTexture	35
4.15.4.5	MobileRect	35
4.15.4.6	SmartRects	35
4.15.4.7	Transparency	35
4.15.4.8	Valid	35
4.15.4.9	Visible	36
4.15.4.10	Width	36
4.15.4.11	windowId	36

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

UWK	5
-------------------------------	---

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

UWK.BridgeEventArgs (Event Handler for JavaScript to Unity callbacks) . . .	9
UWK.BridgeObject (Bridge objects expose functions and properties to JavaScript. Use the Bridge itself to set these)	10
UWK.Command (UWebKit uses a Command structure to pass commands, events, and data <-> the web core. This idiom is used so that the system can readily use multiple cores and to avoid interfering with Unity's rendering and game logic)	11
UWK.CommandHandler (Wraps a Command to prevent boxing/unboxing of structure in event handling)	14
UWK.CommandProcessEventArgs (Event arguments for CommandProcessEvent ent)	14
UWK.Plugin (Interop class representing the native uWebKit Unity plugin) . . .	15
UWK.Process (Class representing the web process Rendering web content in an external process keeps the Unity memory space clean, utilization of additional processor cores, and lets the Unity application recover from issues with web pages)	17
UWK.SubBuffer (When SmartRects are enabled, SubBuffers hold the update information for dirty subregions)	18
UWK.TextureInterop (Structure for communicating texture updates with native plugin)	19
UWK.TextureSet (One or more textures that are used to draw a UWKView and 2D or 3D space. The textures are updated on the fly in response to updates from the web core)	19
uWebKitStandard	20
UWKCore (UWKCore is responsible for creating/managing views and popup windows. It is automatically created and added at runtime when your application requests a web view to be created)	20
UWKKeys (Class to map keyboard input from Unity to Web core)	23
UWKPopup (Minimal WebPopup using uWebKit and Unity GUI (uWebKit Pro))	27

[UWKView](#) ([UWKView](#) encapsulates a WebKit WebView and is used to view
and interact with the view's content) [28](#)

Chapter 3

Namespace Documentation

3.1 Package UWK

Classes

- class [BridgeEventArgs](#)
Event Handler for JavaScript to Unity callbacks.
- class [BridgeObject](#)
Bridge objects expose functions and properties to JavaScript. Use the Bridge itself to set these.
- class **Bridge**
The JavaScript <-> Unity Bridge This static class is used to embed Javascript objects and values in the context of a loaded web page. It is also capable of receiving callbacks from Javascript on the page. The bridge is persistent across pages and page loads. See UnityPageInfo.cs in Examples.
- struct [Command](#)
uWebKit uses a [Command](#) structure to pass commands, events, and data <-> the web core. This idiom is used so that the system can readily use multiple cores and to avoid interfering with Unity's rendering and game logic.
- class [CommandProcessEventArgs](#)
Event arguments for CommandProcessEvent.
- class [CommandHandler](#)
Wraps a [Command](#) to prevent boxing/unboxing of structure in event handling.
- class **UWKConfig**
- class [Plugin](#)
Interop class representing the native uWebKit Unity plugin.
- class [Process](#)
Class representing the web process Rendering web content in an external process keeps the Unity memory space clean, utilization of additional processor cores, and lets the Unity application recover from issues with web pages.
- struct [TextureInterop](#)

Structure for communicating texture updates with native plugin.

- class [SubBuffer](#)

When SmartRects are enabled, SubBuffers hold the update information for dirty sub-regions.

- class [TextureSet](#)

One or more textures that are used to draw a [UWKView](#) and 2D or 3D space. The textures are updated on the fly in response to updates from the web core.

Enumerations

- enum [Source](#) { **PLUGIN** = 0, **PROCESS** = 1 }

A [UWK.Command](#) is either generated via the Plugin (Unity) or via the native Web process.

Functions

- delegate void **BridgeEventHandler** (object sender, [BridgeEventArgs](#) e)
- delegate void **CommandProcessEventHandler** (object sender, [CommandEventArgs](#) e)
- delegate void [PROCESSCB](#) (IntPtr pcmd)
[Plugin](#) -> Managed calling delegate.
- delegate void [SUCCESSCB](#) (ref [Command](#) cmd)
C++ side only and only in process, only here for reference.
- delegate void **ERRORCB** (ref [Command](#) cmd)
- delegate void [LOGCB](#) (string message)
Delegate for logging purposes (native -> Unity)

3.1.1 Enumeration Type Documentation

3.1.1.1 enum [UWK::Source](#)

A [UWK.Command](#) is either generated via the [Plugin](#) (Unity) or via the native Web process.

3.1.2 Function Documentation

3.1.2.1 delegate void [UWK.LOGCB](#) (string *message*)

Delegate for logging purposes (native -> Unity)

3.1.2.2 delegate void [UWK.PROCESSCB](#) (IntPtr *pcmd*)

[Plugin](#) -> Managed calling delegate.

3.1.2.3 delegate void UWK.SUCCESSCB (ref Command *cmd*)

C++ side only and only in process, only here for reference.

Chapter 4

Class Documentation

4.1 UWK.BridgeEventArgs Class Reference

Event Handler for JavaScript to Unity callbacks.

Public Member Functions

- **BridgeEventArgs** (string methodName, string[] args)

Public Attributes

- string [MethodName](#)
The name of the method to be called.
- string[] [Args](#)
The arguments for the method call.

4.1.1 Detailed Description

Event Handler for JavaScript to Unity callbacks.

4.1.2 Member Data Documentation

4.1.2.1 string [] UWK.BridgeEventArgs.Args

The arguments for the method call.

4.1.2.2 string UWK.BridgeEventArgs.MethodName

The name of the method to be called.

The documentation for this class was generated from the following file:

- UWKBridge.cs

4.2 UWK.BridgeObject Class Reference

Bridge objects expose functions and properties to JavaScript. Use the Bridge itself to set these.

Public Member Functions

- void [Invoke](#) (string methodName, string[] parms)
Invoke the specified methodName with parms.
- void [Bind](#) (string methodName, BridgeEventHandler handler)
Bind the specified methodName and handler.

Public Attributes

- string [Name](#)
The name of the bridge object, used to access it in Javascript.
- Dictionary< string, string > [Properties](#) = new Dictionary<string, string> ()
The properties accessible to Javascript.

4.2.1 Detailed Description

Bridge objects expose functions and properties to JavaScript. Use the Bridge itself to set these.

4.2.2 Member Function Documentation

4.2.2.1 void UWK.BridgeObject.Bind (string methodName, BridgeEventHandler handler)

Bind the specified methodName and handler.

4.2.2.2 void UWK.BridgeObject.Invoke (string methodName, string[] parms)

Invoke the specified methodName with parms.

4.2.3 Member Data Documentation

4.2.3.1 string UWK.BridgeObject.Name

The name of the bridge object, used to access it in Javascript.

4.2.3.2 Dictionary<string, string> UWK.BridgeObject.Properties = new Dictionary<string, string> ()

The properties accessible to Javascript.

The documentation for this class was generated from the following file:

- UWKBridge.cs

4.3 UWK.Command Struct Reference

uWebKit uses a [Command](#) structure to pass commands, events, and data <-> the web core. This idiom is used so that the system can readily use multiple cores and to avoid interfering with Unity's rendering and game logic.

Public Member Functions

- void [Init](#) ()
Initializes a command which will be sent to the uWebKit process.
- [CommandHandler Post](#) ()
Posts a command to the command queue for processing.
- string [GetSParam](#) (int index)
Retrieve the commands string parameter at the specified index.
- void [SetSParam](#) (int index, string value)
Sets the commands index parameter to the specified string.
- void [SpanSParams](#) (int startIndex, string s)
This function is deprecated in favor of [Plugin.GetString](#) and [Plugin.AllocateString](#).

Static Public Member Functions

- static [Command NewCommand](#) (string fourcc, params object[] parms)
Allocate a new command with the given fourcc and variable number of int/string parameters.

Public Attributes

- uint **id**
Unique ID of this command, valid once the [Command](#) has been posted.
- string **fourcc**
FOURCC value which is used to designate the type of command.
- Source **src**
The origin of the command, [Plugin](#) (Unity) or Native.
- int[] **iParams**
Array of integer values passes as arguments or return values.
- int **numIParams**
Number of active iParams.
- int **numSParams**
Number of active sParams.
- int **retCode**
Return code: < 0 error, 0 == unprocessed, > 0 = success.
- readonly SUCCESSCB **cbSuccess**
- readonly ERRORCB **cbError**
- IntPtr **pthis**

4.3.1 Detailed Description

uWebKit uses a [Command](#) structure to pass commands, events, and data <-> the web core. This idiom is used so that the system can readily use multiple cores and to avoid interfering with Unity's rendering and game logic.

4.3.2 Member Function Documentation

4.3.2.1 string UWK.Command.GetSParam (int *index*)

Retrieve the commands string parameter at the specified index.

4.3.2.2 void UWK.Command.Init ()

Initializes a command which will be sent to the uWebKit process.

4.3.2.3 static Command UWK.Command.NewCommand (string *fourcc*, params object[] *parms*) [static]

Allocate a new command with the given fourcc and variable number of int/string parameters.

4.3.2.4 CommandHandler UWK.Command.Post ()

Posts a command to the command queue for processing.

4.3.2.5 void UWK.Command.SetSParam (int *index*, string *value*)

Sets the commands index parameter to the specified string.

4.3.2.6 void UWK.Command.SpanSParams (int *startIndex*, string *s*)

This function is deprecated in favor of [Plugin.GetString](#) and [Plugin.AllocateString](#).

4.3.3 Member Data Documentation

4.3.3.1 string UWK.Command.fourcc

FOURCC value which is used to designate the type of command.

4.3.3.2 uint UWK.Command.id

Unique ID of this command, valid once the [Command](#) has been posted.

4.3.3.3 int [] UWK.Command.iParams

Array of integer values passes as arguments or return values.

4.3.3.4 int UWK.Command.numIParams

Number of active iParams.

4.3.3.5 int UWK.Command.numSParams

Number of active sParams.

4.3.3.6 int UWK.Command.retCode

Return code: < 0 error, 0 == unprocessed, > 0 = success.

4.3.3.7 Source UWK.Command.src

The origin of the command, [Plugin](#) (Unity) or Native.

The documentation for this struct was generated from the following file:

- [UWKCommand.cs](#)

4.4 UWK.CommandHandler Class Reference

Wraps a [Command](#) to prevent boxing/unboxing of structure in event handling.

Public Member Functions

- **CommandHandler** (ref [Command](#) cmd)
- void **OnProcessReturn** (object sender, [CommandProcessEventArgs](#) args)

Events

- [CommandProcessEventHandler](#) **Process**

4.4.1 Detailed Description

Wraps a [Command](#) to prevent boxing/unboxing of structure in event handling.

The documentation for this class was generated from the following file:

- [UWKCommand.cs](#)

4.5 UWK.CommandProcessEventArgs Class Reference

Event arguments for CommandProcessEvent.

Public Member Functions

- **CommandProcessEventArgs** ([Command](#) cmd)

Public Attributes

- [Command](#) **Cmd**

4.5.1 Detailed Description

Event arguments for CommandProcessEvent.

The documentation for this class was generated from the following file:

- [UWKCommand.cs](#)

4.6 UWK.Plugin Class Reference

Interop class representing the native uWebKit Unity plugin.

Public Member Functions

- static int **LoadLibrary** (string libraryName)
- static void **UWK_InitProcess** ()

Static Public Member Functions

- static bool **Init** (LOGCB log)
Initializes the plugin and sets the logging system.
- static void **Update** ()
Ticks the plugin.
- static void **Shutdown** ()
Shutdown and cleanup the plugin.
- static void **ClearCommands** ()
Clears all commands and event handlers.
- static void **PostCommand** (ref **Command** cmd)
Posts a command to the command queue and retrieves an id number.
- static void **ProcessCommand** (ref **Command** cmd)
Processes either a return or inbound command.
- static void **ProcessCommand** (IntPtr pcmd)
Processes either a return or inbound command.
- static uint **UpdateTexture** (bool isBackBuffer, int mip, int child, ref **TextureInterop** textureInterop)
Grabs texture data for a TextureSet's backbuffer or SubTexture.
- static string **GetString** (int page, int sz)
Retrieves a string allocated on the uWebKit memory paging system.
- static bool **GetBytes** (int page, int sz, byte[] bytes)
Retrieves raw bytes allocated on the uWebKit memory paging system.
- static int **AllocateString** (string value, ref int size)
Allocates a string on the uWebKit memory paging system.

Events

- static CommandProcessEventHandler **ProcessInbound**
Event handler for inbound commands.
- static CommandProcessEventHandler **ProcessReturn**
Event handler for plugin generated commands that have returned once being processed by the native side.

4.6.1 Detailed Description

Interop class representing the native uWebKit Unity plugin.

4.6.2 Member Function Documentation

4.6.2.1 `static int UWK.Plugin.AllocateString (string value, ref int size) [static]`

Allocates a string on the uWebKit memory paging system.

4.6.2.2 `static void UWK.Plugin.ClearCommands () [static]`

Clears all commands and event handlers.

4.6.2.3 `static bool UWK.Plugin.GetBytes (int page, int sz, byte[] bytes) [static]`

Retrieves raw bytes allocated on the uWebKit memory paging system.

4.6.2.4 `static string UWK.Plugin.GetString (int page, int sz) [static]`

Retrieves a string allocated on the uWebKit memory paging system.

4.6.2.5 `static bool UWK.Plugin.Init (LOGCB log) [static]`

Initializes the plugin and sets the logging system.

4.6.2.6 `static void UWK.Plugin.PostCommand (ref Command cmd) [static]`

Posts a command to the command queue and retrieves an id number.

4.6.2.7 `static void UWK.Plugin.ProcessCommand (IntPtr pcmd) [static]`

Processes either a return or inbound command.

4.6.2.8 `static void UWK.Plugin.ProcessCommand (ref Command cmd) [static]`

Processes either a return or inbound command.

4.6.2.9 `static void UWK.Plugin.Shutdown () [static]`

Shutdown and cleanup the plugin.

4.6.2.10 `static void UWK.Plugin.Update () [static]`

Ticks the plugin.

4.6.2.11 `static uint UWK.Plugin.UpdateTexture (bool isBackBuffer, int mip, int child, ref TextureInterop textureInterop) [static]`

Grabs texture data for a TextureSet's backbuffer or SubTexture.

4.6.3 Event Documentation

4.6.3.1 `CommandProcessEventHandler UWK.Plugin.ProcessInbound [static]`

Event handler for inbound commands.

4.6.3.2 `CommandProcessEventHandler UWK.Plugin.ProcessReturn [static]`

Event handler for plugin generated commands that have returned once being processed by the native side.

The documentation for this class was generated from the following file:

- UWKPlugin.cs

4.7 UWK.Process Class Reference

Class representing the web process Rendering web content in an external process keeps the Unity memory space clean, utilization of additional processor cores, and lets the Unity application recover from issues with web pages.

Public Member Functions

- `bool Go ()`
Start the web process.
- `void KillIt ()`
Kill the web process.
- `void Stop ()`
Posts an exit command to the command queue.

Public Attributes

- `bool KillAttempted = false`

Static Public Attributes

- static bool **Disabled** = false

4.7.1 Detailed Description

Class representing the web process. Rendering web content in an external process keeps the Unity memory space clean, utilization of additional processor cores, and lets the Unity application recover from issues with web pages.

4.7.2 Member Function Documentation

4.7.2.1 bool UWK.Process.Go ()

Start the web process.

4.7.2.2 void UWK.Process.KillIt ()

Kill the web process.

4.7.2.3 void UWK.Process.Stop ()

Posts an exit command to the command queue.

The documentation for this class was generated from the following file:

- UWKProcess.cs

4.8 UWK.SubBuffer Class Reference

When SmartRects are enabled, SubBuffers hold the update information for dirty subregions.

Public Attributes

- bool **Active**
- int **X**
- int **Y**
- int **Width**
- int **Height**
- Texture2D **Texture**
- Color32[] **Pixels**
- GCHandle **PinnedPixels**

4.8.1 Detailed Description

When SmartRects are enabled, SubBuffers hold the update information for dirty subregions.

The documentation for this class was generated from the following file:

- UWKTextureSet.cs

4.9 UWK.TextureInterop Struct Reference

Structure for communicating texture updates with native plugin.

Public Attributes

- int **Dirty**
- IntPtr **Pixels**
- int **X**
- int **Y**
- int **Width**
- int **Height**

4.9.1 Detailed Description

Structure for communicating texture updates with native plugin.

The documentation for this struct was generated from the following file:

- UWKTextureSet.cs

4.10 UWK.TextureSet Class Reference

One or more textures that are used to draw a [UWKView](#) and 2D or 3D space. The textures are updated on the fly in response to updates from the web core.

Public Member Functions

- void **Init** (int width, int height, bool smartRects)
- void [Update](#) ()
Update the necessary textures based on changes to the page.
- void [Release](#) ()
Release the associated texture and buffer data. It is critical to call this to avoid leaks.

Public Attributes

- Texture2D **BackBuffer**
- Color32[] **Pixels**
- GCHandle **PinnedPixels**
- [SubBuffer](#)[,] **SubBuffers**
- int **Width**
- int **Height**
- bool **SmartRects**

4.10.1 Detailed Description

One or more textures that are used to draw a [UWKView](#) and 2D or 3D space. The textures are updated on the fly in response to updates from the web core.

4.10.2 Member Function Documentation

4.10.2.1 void UWK.TextureSet.Release ()

Release the associated texture and buffer data. It is critical to call this to avoid leaks.

4.10.2.2 void UWK.TextureSet.Update ()

Update the necessary textures based on changes to the page.

The documentation for this class was generated from the following file:

- UWKTextureSet.cs

4.11 uWebKitStandard Class Reference

Static Public Attributes

- static string **HTTPS_HTML** = "</center></body></html>"
- static string **ONEVIEW_HTML** = "</center></body></html>"

The documentation for this class was generated from the following file:

- UWKStandard.cs

4.12 UWKCore Class Reference

[UWKCore](#) is responsible for creating/managing views and popup windows. It is automatically created and added at runtime when your application requests a web view to be created.

Static Public Member Functions

- static void [Init](#) (bool inactivation)
Main initialization of web core, must be called before any UWKViews are created.
- static void [Init](#) ()
Unity 3.4 MonoDevelop on Windows chokes on default parameters, so we have an override.
- static [UWKView CreateView](#) (string name, string URL, int width, int height, bool smartRects)
The main method to create a [UWKView](#), note that views with identical names are reused.
- static int **GetNumViews** ()
- static [UWKView CreateView](#) (string name, int width, int height, bool smartRects)
- static [UWKView CreateView](#) (string name, string URL, int width, int height)
- static [UWKView CreateView](#) (string name, int width, int height)
- static void [ClosePopup](#) ([UWKPopup](#) p)
Closes the popup.
- static void [RemoveView](#) ([UWKView](#) view)
Remove a [UWKView](#) from the core.
- static [UWKView GetView](#) (string name)
Gets the [UWKView](#) associated with the unique name.
- static void [ClearCookies](#) ()
Clear the persistent cookies (saved session data) associated with this application.

Static Public Attributes

- static bool **DestroyViewsOnLevelLoad** = false
- static bool **ProcessError** = false
- static bool **RuntimeError** = false
- static bool **imeEnabled** = false
- static bool **StandardVersion** = false
- static bool **ProductTrial** = false
- static bool **ActivationRequired** = false
- static string **ProductKey** = ""
- static UWKProcessStartedDelegate **UWKProcessStarted**
- static UWKProcessZombifiedDelegate **UWKProcessZombified**
- static UWKProcessWillRestartDelegate **UWKProcessWillRestart**
- static UWKProcessRestartedDelegate **UWKProcessRestarted**

Properties

- static bool [ProcessUp](#) [get]
Gets a value indicating whether the web process is running.

4.12.1 Detailed Description

[UWKCore](#) is responsible for creating/managing views and popup windows. It is automatically created and added at runtime when your application requests a web view to be created.

4.12.2 Member Function Documentation

4.12.2.1 `static void UWKCore.ClearCookies () [static]`

Clear the persistent cookies (saved session data) associated with this application.

4.12.2.2 `static void UWKCore.ClosePopup (UWKPopup p) [static]`

Closes the popup.

4.12.2.3 `static UWKView UWKCore.CreateView (string name, string URL, int width, int height, bool smartRects) [static]`

The main method to create a [UWKView](#), note that views with identical names are reused.

4.12.2.4 `static UWKView UWKCore.GetView (string name) [static]`

Gets the [UWKView](#) associated with the unique name.

4.12.2.5 `static void UWKCore.Init () [static]`

Unity 3.4 MonoDevelop on Windows chokes on default parameters, so we have an override.

4.12.2.6 `static void UWKCore.Init (bool inactivation) [static]`

Main initialization of web core, must be called before any UWKViews are created.

4.12.2.7 `static void UWKCore.RemoveView (UWKView view) [static]`

Remove a [UWKView](#) from the core.

4.12.3 Property Documentation

4.12.3.1 `bool UWKCore.ProcessUp` `[static, get]`

Gets a value indicating whether the web process is running.

The documentation for this class was generated from the following file:

- UWKCore.cs

4.13 UWKKeys Class Reference

Class to map keyboard input from Unity to Web core.

Public Types

- enum **KeyboardModifier** {
NoModifier = 0x00000000, **ShiftModifier** = 0x02000000, **ControlModifier** = 0x04000000, **AltModifier** = 0x08000000,
MetaModifier = 0x10000000, **KeypadModifier** = 0x20000000, **GroupSwitch-Modifier** = 0x40000000, **KeyboardModifierMask** = 0xfe000000 }
- enum **Modifier** {
META = KeyboardModifier.MetaModifier, **SHIFT** = KeyboardModifier.ShiftModifier, **CTRL** = KeyboardModifier.ControlModifier, **ALT** = KeyboardModifier.AltModifier,
MODIFIER_MASK = KeyboardModifier.KeyboardModifierMask, **UNICODE_ACCEL** = 0x00000000 }
- enum **MouseButton** {
NoButton = 0x00000000, **LeftButton** = 0x00000001, **RightButton** = 0x00000002, **MidButton** = 0x00000004,
XButton1 = 0x00000008, **XButton2** = 0x00000010, **MouseButtonMask** = 0x000000ff }
- enum **FocusPolicy** {
NoFocus = 0, **TabFocus** = 0x1, **ClickFocus** = 0x2, **StrongFocus** = TabFocus | ClickFocus | 0x8,
WheelFocus = StrongFocus | 0x4 }
- enum **QtKey** {
Key_Escape = 0x01000000, **Key_Tab** = 0x01000001, **Key_Backtab** = 0x01000002, **Key_Backspace** = 0x01000003,
Key_Return = 0x01000004, **Key_Enter** = 0x01000005, **Key_Insert** = 0x01000006, **Key_Delete** = 0x01000007,
Key_Pause = 0x01000008, **Key_Print** = 0x01000009, **Key_SysReq** = 0x0100000a, **Key_Clear** = 0x0100000b,
Key_Home = 0x01000010, **Key_End** = 0x01000011, **Key_Left** = 0x01000012, **Key_Up** = 0x01000013,

Key_Right = 0x01000014, **Key_Down** = 0x01000015, **Key_PageUp** = 0x01000016, **Key_PageDown** = 0x01000017,

Key_Shift = 0x01000020, **Key_Control** = 0x01000021, **Key_Meta** = 0x01000022, **Key_Alt** = 0x01000023,

Key_CapsLock = 0x01000024, **Key_NumLock** = 0x01000025, **Key_ScrollLock** = 0x01000026, **Key_F1** = 0x01000030,

Key_F2 = 0x01000031, **Key_F3** = 0x01000032, **Key_F4** = 0x01000033, **Key_F5** = 0x01000034,

Key_F6 = 0x01000035, **Key_F7** = 0x01000036, **Key_F8** = 0x01000037, **Key_F9** = 0x01000038,

Key_F10 = 0x01000039, **Key_F11** = 0x0100003a, **Key_F12** = 0x0100003b, **Key_F13** = 0x0100003c,

Key_F14 = 0x0100003d, **Key_F15** = 0x0100003e, **Key_F16** = 0x0100003f, **Key_F17** = 0x01000040,

Key_F18 = 0x01000041, **Key_F19** = 0x01000042, **Key_F20** = 0x01000043, **Key_F21** = 0x01000044,

Key_F22 = 0x01000045, **Key_F23** = 0x01000046, **Key_F24** = 0x01000047, **Key_F25** = 0x01000048,

Key_F26 = 0x01000049, **Key_F27** = 0x0100004a, **Key_F28** = 0x0100004b, **Key_F29** = 0x0100004c,

Key_F30 = 0x0100004d, **Key_F31** = 0x0100004e, **Key_F32** = 0x0100004f, **Key_F33** = 0x01000050,

Key_F34 = 0x01000051, **Key_F35** = 0x01000052, **Key_Super_L** = 0x01000053, **Key_Super_R** = 0x01000054,

Key_Menu = 0x01000055, **Key_Hyper_L** = 0x01000056, **Key_Hyper_R** = 0x01000057, **Key_Help** = 0x01000058,

Key_Direction_L = 0x01000059, **Key_Direction_R** = 0x01000060, **Key_Space** = 0x20, **Key_Any** = **Key_Space**,

Key_Exclaim = 0x21, **Key_QuoteDbl** = 0x22, **Key_NumberSign** = 0x23, **Key_Dollar** = 0x24,

Key_Percent = 0x25, **Key_Ampersand** = 0x26, **Key_Apostrophe** = 0x27, **Key_ParenLeft** = 0x28,

Key_ParenRight = 0x29, **Key_Asterisk** = 0x2a, **Key_Plus** = 0x2b, **Key_Comma** = 0x2c,

Key_Minus = 0x2d, **Key_Period** = 0x2e, **Key_Slash** = 0x2f, **Key_0** = 0x30,

Key_1 = 0x31, **Key_2** = 0x32, **Key_3** = 0x33, **Key_4** = 0x34,

Key_5 = 0x35, **Key_6** = 0x36, **Key_7** = 0x37, **Key_8** = 0x38,

Key_9 = 0x39, **Key_Colon** = 0x3a, **Key_Semicolon** = 0x3b, **Key_Less** = 0x3c,

Key_Equal = 0x3d, **Key_Greater** = 0x3e, **Key_Question** = 0x3f, **Key_At** = 0x40,

Key_A = 0x41, **Key_B** = 0x42, **Key_C** = 0x43, **Key_D** = 0x44,

Key_E = 0x45, **Key_F** = 0x46, **Key_G** = 0x47, **Key_H** = 0x48,

Key_I = 0x49, **Key_J** = 0x4a, **Key_K** = 0x4b, **Key_L** = 0x4c,

Key_M = 0x4d, **Key_N** = 0x4e, **Key_O** = 0x4f, **Key_P** = 0x50,
Key_Q = 0x51, **Key_R** = 0x52, **Key_S** = 0x53, **Key_T** = 0x54,
Key_U = 0x55, **Key_V** = 0x56, **Key_W** = 0x57, **Key_X** = 0x58,
Key_Y = 0x59, **Key_Z** = 0x5a, **Key_BracketLeft** = 0x5b, **Key_Backslash** = 0x5c,
Key_BracketRight = 0x5d, **Key_AsciiCircum** = 0x5e, **Key_Underscore** = 0x5f,
Key_QuoteLeft = 0x60,
Key_BraceLeft = 0x7b, **Key_Bar** = 0x7c, **Key_BraceRight** = 0x7d, **Key_AsciiTilde**
= 0x7e,
Key_nobreakspace = 0x0a0, **Key_exclamdown** = 0x0a1, **Key_cent** = 0x0a2,
Key_sterling = 0x0a3,
Key_currency = 0x0a4, **Key_yen** = 0x0a5, **Key_brokenbar** = 0x0a6, **Key_-**
section = 0x0a7,
Key_diaeresis = 0x0a8, **Key_copyright** = 0x0a9, **Key_ordfeminine** = 0x0aa,
Key_guillemotleft = 0x0ab,
Key_notsign = 0x0ac, **Key_hyphen** = 0x0ad, **Key_registered** = 0x0ae, **Key_-**
macron = 0x0af,
Key_degree = 0x0b0, **Key_plusminus** = 0x0b1, **Key_twosuperior** = 0x0b2,
Key_threesuperior = 0x0b3,
Key_acute = 0x0b4, **Key_mu** = 0x0b5, **Key_paragraph** = 0x0b6, **Key_periodcentered**
= 0x0b7,
Key_cedilla = 0x0b8, **Key_onesuperior** = 0x0b9, **Key_masculine** = 0x0ba,
Key_guillemotright = 0x0bb,
Key_onequarter = 0x0bc, **Key_onehalf** = 0x0bd, **Key_threequarters** = 0x0be,
Key_questiondown = 0x0bf,
Key_Agrave = 0x0c0, **Key_Aacute** = 0x0c1, **Key_Acircumflex** = 0x0c2, **Key_-**
Atilde = 0x0c3,
Key_Adiaeresis = 0x0c4, **Key_Aring** = 0x0c5, **Key_AE** = 0x0c6, **Key_Ccedilla**
= 0x0c7,
Key_Egrave = 0x0c8, **Key_Eacute** = 0x0c9, **Key_Ecircumflex** = 0x0ca, **Key_-**
Ediaeresis = 0x0cb,
Key_Igrave = 0x0cc, **Key_Iacute** = 0x0cd, **Key_Icircumflex** = 0x0ce, **Key_-**
Idiaeresis = 0x0cf,
Key_ETH = 0x0d0, **Key_Ntilde** = 0x0d1, **Key_Ograve** = 0x0d2, **Key_Oacute** =
0x0d3,
Key_Ocircumflex = 0x0d4, **Key_Otilde** = 0x0d5, **Key_Odiaeresis** = 0x0d6, **Key_-**
multiply = 0x0d7,
Key_Ooblique = 0x0d8, **Key_Ugrave** = 0x0d9, **Key_Uacute** = 0x0da, **Key_-**
Ucircumflex = 0x0db,
Key_Udiaeresis = 0x0dc, **Key_Yacute** = 0x0dd, **Key_THORN** = 0x0de, **Key_-**
ssharp = 0x0df,
Key_division = 0x0f7, **Key_ydiaeresis** = 0x0ff, **Key_AltGr** = 0x01001103, **Key_-**
Multi_key = 0x01001120,

Key_Codeinput = 0x01001137, **Key_SingleCandidate** = 0x0100113c, **Key_-MultipleCandidate** = 0x0100113d, **Key_PreviousCandidate** = 0x0100113e,
Key_Mode_switch = 0x0100117e, **Key_Kanji** = 0x01001121, **Key_Muhenkan** = 0x01001122, **Key_Henkan** = 0x01001123,
Key_Romaji = 0x01001124, **Key_Hiragana** = 0x01001125, **Key_Katakana** = 0x01001126, **Key_Hiragana_Katakana** = 0x01001127,
Key_Zenkaku = 0x01001128, **Key_Hankaku** = 0x01001129, **Key_Zenkaku_-Hankaku** = 0x0100112a, **Key_Touroku** = 0x0100112b,
Key_Massyo = 0x0100112c, **Key_Kana_Lock** = 0x0100112d, **Key_Kana_Shift** = 0x0100112e, **Key_Eisu_Shift** = 0x0100112f,
Key_Eisu_toggle = 0x01001130, **Key_Hangul** = 0x01001131, **Key_Hangul_-Start** = 0x01001132, **Key_Hangul_End** = 0x01001133,
Key_Hangul_Hanja = 0x01001134, **Key_Hangul_Jamo** = 0x01001135, **Key_-Hangul_Romaja** = 0x01001136, **Key_Hangul_Jeonja** = 0x01001138,
Key_Hangul_Banja = 0x01001139, **Key_Hangul_PreHanja** = 0x0100113a, **Key_-Hangul_PostHanja** = 0x0100113b, **Key_Hangul_Special** = 0x0100113f,
Key_Dead_Grave = 0x01001250, **Key_Dead_Acute** = 0x01001251, **Key_Dead_-Circumflex** = 0x01001252, **Key_Dead_Tilde** = 0x01001253,
Key_Dead_Macron = 0x01001254, **Key_Dead_Breve** = 0x01001255, **Key_Dead_-Abovedot** = 0x01001256, **Key_Dead_Diaeresis** = 0x01001257,
Key_Dead_Abovering = 0x01001258, **Key_Dead_Doubleacute** = 0x01001259, **Key_Dead_Caron** = 0x0100125a, **Key_Dead_Cedilla** = 0x0100125b,
Key_Dead_Ogonek = 0x0100125c, **Key_Dead_Iota** = 0x0100125d, **Key_Dead_-Voiced_Sound** = 0x0100125e, **Key_Dead_Semivoiced_Sound** = 0x0100125f,
Key_Dead_Belowdot = 0x01001260, **Key_Dead_Hook** = 0x01001261, **Key_-Dead_Horn** = 0x01001262, **Key_Back** = 0x01000061,
Key_Forward = 0x01000062, **Key_Stop** = 0x01000063, **Key_Refresh** = 0x01000064, **Key_VolumeDown** = 0x01000070,
Key_VolumeMute = 0x01000071, **Key_VolumeUp** = 0x01000072, **Key_BassBoost** = 0x01000073, **Key_BassUp** = 0x01000074,
Key_BassDown = 0x01000075, **Key_TrebleUp** = 0x01000076, **Key_TrebleDown** = 0x01000077, **Key_MediaPlay** = 0x01000080,
Key_MediaStop = 0x01000081, **Key_MediaPrevious** = 0x01000082, **Key_MediaNext** = 0x01000083, **Key_MediaRecord** = 0x01000084,
Key_HomePage = 0x01000090, **Key_Favorites** = 0x01000091, **Key_Search** = 0x01000092, **Key_Standby** = 0x01000093,
Key_OpenUrl = 0x01000094, **Key_LaunchMail** = 0x010000a0, **Key_LaunchMedia** = 0x010000a1, **Key_Launch0** = 0x010000a2,
Key_Launch1 = 0x010000a3, **Key_Launch2** = 0x010000a4, **Key_Launch3** = 0x010000a5, **Key_Launch4** = 0x010000a6,
Key_Launch5 = 0x010000a7, **Key_Launch6** = 0x010000a8, **Key_Launch7** = 0x010000a9, **Key_Launch8** = 0x010000aa,

```

Key_Launch9 = 0x010000ab, Key_LaunchA = 0x010000ac, Key_LaunchB =
0x010000ad, Key_LaunchC = 0x010000ae,
Key_LaunchD = 0x010000af, Key_LaunchE = 0x010000b0, Key_LaunchF =
0x010000b1, Key_MediaLast = 0x0100ffff,
Key_Select = 0x01010000, Key_Yes = 0x01010001, Key_No = 0x01010002,
Key_Cancel = 0x01020001,
Key_Printer = 0x01020002, Key_Execute = 0x01020003, Key_Sleep = 0x01020004,
Key_Play = 0x01020005,
Key_Zoom = 0x01020006, Key_Context1 = 0x01100000, Key_Context2 = 0x01100001,
Key_Context3 = 0x01100002,
Key_Context4 = 0x01100003, Key_Call = 0x01100004, Key_Hangup = 0x01100005,
Key_Flip = 0x01100006,
Key_unknown = 0x01ffffff }

```

Static Public Member Functions

- static uint **MapUnityKey** (KeyCode k)

Static Public Attributes

- static Dictionary< KeyCode, QtKey > **KeyMap** = new Dictionary<KeyCode, QtKey>()
- static Dictionary< KeyCode, string > **AsciiMap** = new Dictionary<KeyCode, string>()

4.13.1 Detailed Description

Class to map keyboard input from Unity to Web core.

The documentation for this class was generated from the following file:

- UWKKeys.cs

4.14 UWKPopup Class Reference

Minimal WebPopup using uWebKit and Unity GUI (uWebKit Pro)

Public Member Functions

- void **GetCenterPos** (ref Vector2 pos)
- void **Center** ()

Public Attributes

- bool **Valid**
- [UWKView](#) **View**
- float **X**
- float **Y**
- int **Width** = 1024
- int **Height** = 600
- bool **Visible** = true

4.14.1 Detailed Description

Minimal WebPopup using uWebKit and Unity GUI (uWebKit Pro)

The documentation for this class was generated from the following file:

- UWKPopup.cs

4.15 UWKView Class Reference

[UWKView](#) encapsulates a WebKit WebView and is used to view and interact with the view's content.

Public Member Functions

- void **DrawGUI** (int x, int y)
- void **DrawGUI** (int x, int y, Rect r)
- void [DrawGUI](#) (int x, int y, int width, int height)
Main 2D drawing method for a [UWKView](#), draws the view at the given X, Y coords with width and height dimensions.
- void **BringToFront** ()
- void [BringToFront](#) (bool inGUI)
Brings the view to the top of the stack, the top view receives mouse and keyboard input.
- void [PushToBack](#) ()
Pushes this view to the back of the view stack.
- void [DrawTextIME](#) (int x, int y)
Draws the text IME for Chinese, Japanese, Korean languages.
- void [Remove](#) ()
Remove a view and deregister it from the WebCore.
- void [Hide](#) ()
Hides the view.
- void [Show](#) ()
Show the view.

- void [Show](#) (bool show)
Hide or show the specified view depending on bool argument.
- void [EvaluateJavaScript](#) (string script, CommandProcessEventHandler resultHandler)

Evaluates the given Javascript on the loaded page. if you specify your own return value handler you must consume the return string with `Plugin.GetString` as shown in the `evalResult` above, otherwise you may run out of buffer allocations.

- void **EvaluateJavaScript** (string script)
- void [OnGUIMouse](#) (int xOffset, int yOffset)
Handles GUI mouse input including position, buttons, and scroll wheel for 2D UnityGUI uWebKit integration.
- void **OnGUIMouse** (float xOffset, float yOffset)
- void **OnMouseDown** (int x, int y, int button)
- void **OnMouseUp** (int x, int y, int button)
- void **OnScrollWheel** (float scroll)
- void **SetMousePos** (int x, int y)
- void [ProcessKey](#) (Event e)
Processes the key event.
- void **ProcessKeyboard** (Event e)
- void **OnWebGUI** (int x, int y)
- void **OnWebGUI** (int x, int y, int width, int height)
- void [OnWebGUI](#) (int x, int y, int width, int height, float transparency)
Handles most GUI tasks of a uWebKit view in a UnityGUI context this method should be called from the `OnGUI` method of an associated `MonoBehavior`.
- void [Forward](#) ()
Navigates the page forward in navigation history (if possible)
- void [Back](#) ()
Navigates the page backwards in navigation history (if possible)
- void [LoadTextAssetHTML](#) (TextAsset text)
Loads the specified text asset which contains HTML This can be used to load local web content.
- void [LoadHTML](#) (string HTML)
Loads the specified HTML string directly in the view, can be used for generating web content on the fly.
- void [LoadURL](#) (string url)
Navigate the view to the specified URL (`http://`, `file://`, etc)
- void [UpdateView](#) ()
Updates the view's texture data in response to a `UPVW` command from the webcore.
- void [captureTexture](#) ()
iOS only, captures a cached view for use as a texture mapped to a 3D model

Static Public Member Functions

- static void [InvalidateAllViews](#) ()
Invalidates all views, generally in the response to an issue with a `UWKProcess` restart.
- static void [RestartAllViews](#) ()
Restarts all existing views, once the `UWKProcess` has restarted.

Public Attributes

- string **Name**
The View's unique name as registered with the [UWKCore](#).
- string **URL**
The current URL loaded into the [UWKView](#), note that the `LoadHTML` and `LoadURL` methods should be used for changing the view's URL.
- string **Title**
The active page's title string.
- Texture2D **Icon**
The active page's icon (as seen in browser tab, etc)
- bool **JSPopup** = false
Whether this view is a Javascript popup.
- bool **TextInputActive** = false
- Rect **TextInputRect**
- string **TextInputType** = ""
- TextureSet **TSet**
The TextureSet tied to this view, see [UWKTextureSet.cs](#).
- int **ContentWidth**
ContentWidth and ContentHeight are read only variables based on the dimensions of the content loaded into the WebView, if Width < ContentWidth or Height < ContentHeight the web view will contain scrollbars.
- int **ContentHeight**
- bool **ResizeToContents** = false
If you would like the [UWKView](#) to resize to the actual size of the page contents, set `ResizeToContents` to true.
- URLChangedDelegate **URLChanged**
- TitleChangedDelegate **TitleChanged**
- ViewCreatedDelegate **ViewCreated**
- LoadFinishedDelegate **LoadFinished**
- LoadProgressDelegate **LoadProgress**

Static Public Attributes

- static int **sFrontWindow** = 10000

Properties

- Texture2D **MainTexture** [get]
A convenience property for quick access to the [UWKView](#)'s backbuffer texture Note that when using `SmartRects`, the backbuffer may have regions that are out of date and require the subregion textures to display properly, see [UWKTextureSet](#).
- bool **SmartRects** [get, set]

Sets whether or not the [UWKView](#) uses SmartRects (dirty rectangles). SmartRects can greatly speed updates of large pages which feature small areas that are changing whilst most of the page remains static. SmartRects are designed to work with 2D web content should be disabled when running web content mapped in 3D or if transparency is required SmartRects can be toggled on the fly.

- bool [Visible](#) [get, set]
Controls whether the view is drawn.
- bool [AlphaMask](#) [get, set]
Controls whether the view is alpha masked (body { background-color: transparent;}).
- bool [Valid](#) [get, set]
A [UWKView](#) is invalid until the WebCore creates and initializes it. uWebKit uses an event pattern to notify the view once it is valid. Calls that interact with the view should be avoided until the view has been validated.
- int [Width](#) [get, set]
The Width of the WebView, note that the view is clamped to ≥ 128 and ≤ 1024 . This may be increased in a future version of uWebKit, however as we need to internally resize to a pow2 texture, the next jump would be 2048.
- int [Height](#) [get, set]
The Height of the WebView, see notes for the Width property.
- bool [Active](#) [get, set]
Controls whether the view is active, deactivate views save CPU as they are not updated in native code and thus their texture data also doesn't need to be updated, this can be used to cull web content for instance when a 3D object becomes invisible.
- float [Transparency](#) [get, set]
Gets or sets the transparency from 0.0 invisible to 1.0 fully opaque.
- Rect [MobileRect](#) [get, set]
Gets or sets the rectangle where the view is drawn on iOS.
- int [windowId](#) [get, set]
Gets or sets the window identifier which is used to control which view receives mouse and keyboard input (see [BringToFront](#))

4.15.1 Detailed Description

[UWKView](#) encapsulates a WebKit WebView and is used to view and interact with the view's content.

4.15.2 Member Function Documentation

4.15.2.1 void UWKView.Back ()

Navigates the page backwards in navigation history (if possible)

4.15.2.2 void UWKView.BringToFront (bool inGUI)

Brings the view to the top of the stack, the top view receives mouse and keyboard input.

4.15.2.3 void UWKView.captureTexture ()

iOS only, captures a cached view for use as a texture mapped to a 3D model

4.15.2.4 void UWKView.DrawGUI (int x, int y, int width, int height)

Main 2D drawing method for a [UWKView](#), draws the view at the given X, Y coords with width and height dimensions.

4.15.2.5 void UWKView.DrawTextIME (int x, int y)

Draws the text IME for Chinese, Japanese, Korean languages.

4.15.2.6 void UWKView.EvaluateJavaScript (string script, CommandProcessEventHandler resultHandler)

Evaluates the given Javascript on the loaded page. if you specify your own return value handler you must consume the return string with Plugin.GetString as shown in the eval-Result above, otherwise you may run out of buffer allocations.

4.15.2.7 void UWKView.Forward ()

Navigates the page forward in navigation history (if possible)

4.15.2.8 void UWKView.Hide ()

Hides the view.

4.15.2.9 static void UWKView.InvalidateAllViews () [static]

Invalidates all views, generally in the response to an issue with a UWKProcess restart.

4.15.2.10 void UWKView.LoadHTML (string HTML)

Loads the specified HTML string directly in the view, can be used for generating web content on the fly.

4.15.2.11 void UWKView.LoadTextAssetHTML (TextAsset text)

Loads the specified text asset which contains HTML This can be used to load local web content.

4.15.2.12 void UWKView.LoadURL (string *url*)

Navigate the view to the specified URL (<http://>, <file://>, etc)

4.15.2.13 void UWKView.OnGUIMouse (int *xOffset*, int *yOffset*)

Handles GUI mouse input including position, buttons, and scroll wheel for 2D UnityGUI uWebKit integration.

4.15.2.14 void UWKView.OnWebGUI (int *x*, int *y*, int *width*, int *height*, float *transparency*)

Handles most GUI tasks of a uWebKit view in a UnityGUI context this method should be called from the OnGUI method of an associated MonoBehaviour.

4.15.2.15 void UWKView.ProcessKey (Event *e*)

Processes the key event.

4.15.2.16 void UWKView.PushToBack ()

Pushes this view to the back of the view stack.

4.15.2.17 void UWKView.Remove ()

Remove a view and deregister it from the WebCore.

4.15.2.18 static void UWKView.RestartAllViews () [static]

Restarts all existing views, once the UWKProcess has restarted.

4.15.2.19 void UWKView.Show ()

Show the view.

4.15.2.20 void UWKView.Show (bool *show*)

Hide or show the specified view depending on bool argument.

4.15.2.21 void UWKView.UpdateView ()

Updates the view's texture data in response to a UPVW command from the webcore.

4.15.3 Member Data Documentation

4.15.3.1 int UWKView.ContentWidth

ContentWidth and ContentHeight are read only variables based on the dimensions of the content loaded into the WebView, if Width < ContentWidth or Height < ContentHeight the web view will contain scrollbars.

4.15.3.2 Texture2D UWKView.Icon

The active page's icon (as seen in browser tab, etc)

4.15.3.3 bool UWKView.JSPopup = false

Whether this view is a Javascript popup.

4.15.3.4 string UWKView.Name

The View's unique name as registered with the [UWKCore](#).

4.15.3.5 bool UWKView.ResizeToContents = false

If you would like the [UWKView](#) to resize to the actual size of the page contents, set ResizeToContents to true.

4.15.3.6 string UWKView.Title

The active page's title string.

4.15.3.7 TextureSet UWKView.TSet

The TextureSet tied to this view, see UWKTextureSet.cs.

4.15.3.8 string UWKView.URL

The current URL loaded into the [UWKView](#), note that the LoadHTML and LoadURL methods should be used for changing the view's URL.

4.15.4 Property Documentation

4.15.4.1 bool UWKView.Active [get, set]

Controls whether the view is active, deactive views save CPU as they are not updated in native code and thus their texture data also doesn't need to be updated, this can be used to cull web content for instance when a 3D object becomes invisible.

4.15.4.2 bool UWKView.AlphaMask [get, set]

Controls whether the view is alpha masked (body { background-color: transparent;}).

4.15.4.3 int UWKView.Height [get, set]

The Height of the WebView, see notes for the Width property.

4.15.4.4 Texture2D UWKView.MainTexture [get]

A convenience property for quick access to the UWKView's backbuffer texture Note that when using SmartRects, the backbuffer may have regions that are out of date and require the subregion textures to display properly, see UWKTextureSet.

4.15.4.5 Rect UWKView.MobileRect [get, set]

Gets or sets the rectangle where the view is drawn on iOS.

4.15.4.6 bool UWKView.SmartRects [get, set]

Sets whether or not the [UWKView](#) uses SmartRects (dirty rectangles). SmartRects can greatly speed updates of large pages which feature small areas that are changing whilst most of the page remains static. SmartRects are designed to work with 2D web content should be disabled when running web content mapped in 3D or if transparency is required SmartRects can be toggled on the fly.

4.15.4.7 float UWKView.Transparency [get, set]

Gets or sets the transparency from 0.0 invisible to 1.0 fully opaque.

4.15.4.8 bool UWKView.Valid [get, set]

A [UWKView](#) is invalid until the WebCore creates and initializes it. uWebKit uses an event pattern to notify the view once it is valid. Calls that interact with the view should be avoided until the view has been validated.

4.15.4.9 bool UWKView.Visible [get, set]

Controls whether the view is drawn.

4.15.4.10 int UWKView.Width [get, set]

The Width of the WebView, note that the view is clamped to ≥ 128 and ≤ 1024 . This may be increased in a future version of uWebKit, however as we need to internally resize to a pow2 texture, the next jump would be 2048.

4.15.4.11 int UWKView.windowId [get, set]

Gets or sets the window identifier which is used to control which view receives mouse and keyboard input (see BringToFront)

The documentation for this class was generated from the following file:

- UWKView.cs

Index

- Active
 - UWKView, [34](#)
- AllocateString
 - UWK::Plugin, [16](#)
- AlphaMask
 - UWKView, [35](#)
- Args
 - UWK::BridgeEventArgs, [9](#)
- Back
 - UWKView, [31](#)
- Bind
 - UWK::BridgeObject, [10](#)
- BringToFront
 - UWKView, [31](#)
- captureTexture
 - UWKView, [31](#)
- ClearCommands
 - UWK::Plugin, [16](#)
- ClearCookies
 - UWKCore, [22](#)
- ClosePopup
 - UWKCore, [22](#)
- ContentWidth
 - UWKView, [34](#)
- CreateView
 - UWKCore, [22](#)
- DrawGUI
 - UWKView, [32](#)
- DrawTextIME
 - UWKView, [32](#)
- EvaluateJavaScript
 - UWKView, [32](#)
- Forward
 - UWKView, [32](#)
- fourcc
 - UWK::Command, [13](#)
- GetBytes
 - UWK::Plugin, [16](#)
- GetSPParam
 - UWK::Command, [12](#)
- GetString
 - UWK::Plugin, [16](#)
- GetView
 - UWKCore, [22](#)
- Go
 - UWK::Process, [18](#)
- Height
 - UWKView, [35](#)
- Hide
 - UWKView, [32](#)
- Icon
 - UWKView, [34](#)
- id
 - UWK::Command, [13](#)
- Init
 - UWK::Command, [12](#)
 - UWK::Plugin, [16](#)
 - UWKCore, [22](#)
- InvalidateAllViews
 - UWKView, [32](#)
- Invoke
 - UWK::BridgeObject, [10](#)
- iParams
 - UWK::Command, [13](#)
- JSPopup
 - UWKView, [34](#)
- KillIt
 - UWK::Process, [18](#)
- LoadHTML
 - UWKView, [32](#)
- LoadTextAssetHTML
 - UWKView, [32](#)
- LoadURL

- UWKView, 32
- LOGCB
 - UWK, 6
- MainTexture
 - UWKView, 35
- MethodName
 - UWK::BridgeEventArgs, 9
- MobileRect
 - UWKView, 35
- Name
 - UWK::BridgeObject, 11
 - UWKView, 34
- NewCommand
 - UWK::Command, 12
- numIParams
 - UWK::Command, 13
- numSParams
 - UWK::Command, 13
- OnGUIMouse
 - UWKView, 33
- OnWebGUI
 - UWKView, 33
- Post
 - UWK::Command, 12
- PostCommand
 - UWK::Plugin, 16
- PROCESSCB
 - UWK, 6
- ProcessCommand
 - UWK::Plugin, 16
- ProcessInbound
 - UWK::Plugin, 17
- ProcessKey
 - UWKView, 33
- ProcessReturn
 - UWK::Plugin, 17
- ProcessUp
 - UWKCore, 22
- Properties
 - UWK::BridgeObject, 11
- PushToBack
 - UWKView, 33
- Release
 - UWK::TextureSet, 20
- Remove
 - UWKView, 33
- RemoveView
 - UWKCore, 22
- ResizeToContents
 - UWKView, 34
- RestartAllViews
 - UWKView, 33
- retCode
 - UWK::Command, 13
- SetSParam
 - UWK::Command, 13
- Show
 - UWKView, 33
- Shutdown
 - UWK::Plugin, 16
- SmartRects
 - UWKView, 35
- Source
 - UWK, 6
- SpanSParams
 - UWK::Command, 13
- src
 - UWK::Command, 13
- Stop
 - UWK::Process, 18
- SUCCESSCB
 - UWK, 6
- Title
 - UWKView, 34
- Transparency
 - UWKView, 35
- TSet
 - UWKView, 34
- Update
 - UWK::Plugin, 16
 - UWK::TextureSet, 20
- UpdateTexture
 - UWK::Plugin, 17
- UpdateView
 - UWKView, 33
- URL
 - UWKView, 34
- uWebKitStandard, 20
- UWK, 5
 - LOGCB, 6
 - PROCESSCB, 6
 - Source, 6
 - SUCCESSCB, 6

- UWK::BridgeEventArgs, 9
 - Args, 9
 - MethodName, 9
- UWK::BridgeObject, 10
 - Bind, 10
 - Invoke, 10
 - Name, 11
 - Properties, 11
- UWK::Command, 11
 - fourcc, 13
 - GetSPParam, 12
 - id, 13
 - Init, 12
 - iParams, 13
 - NewCommand, 12
 - numIParams, 13
 - numSPParams, 13
 - Post, 12
 - retCode, 13
 - SetSPParam, 13
 - SpanSPParams, 13
 - src, 13
- UWK::CommandHandler, 14
- UWK::CommandProcessEventArgs, 14
- UWK::Plugin, 15
 - AllocateString, 16
 - ClearCommands, 16
 - GetBytes, 16
 - GetString, 16
 - Init, 16
 - PostCommand, 16
 - ProcessCommand, 16
 - ProcessInbound, 17
 - ProcessReturn, 17
 - Shutdown, 16
 - Update, 16
 - UpdateTexture, 17
- UWK::Process, 17
 - Go, 18
 - KillIt, 18
 - Stop, 18
- UWK::SubBuffer, 18
- UWK::TextureInterop, 19
- UWK::TextureSet, 19
 - Release, 20
 - Update, 20
- UWKCore, 20
 - ClearCookies, 22
 - ClosePopup, 22
 - CreateView, 22
 - GetView, 22
 - Init, 22
 - ProcessUp, 22
 - RemoveView, 22
- UWKKeys, 23
- UWKPopup, 27
- UWKView, 28
 - Active, 34
 - AlphaMask, 35
 - Back, 31
 - BringToFront, 31
 - captureTexture, 31
 - ContentWidth, 34
 - DrawGUI, 32
 - DrawTextIME, 32
 - EvaluateJavaScript, 32
 - Forward, 32
 - Height, 35
 - Hide, 32
 - Icon, 34
 - InvalidateAllViews, 32
 - JSPopup, 34
 - LoadHTML, 32
 - LoadTextAssetHTML, 32
 - LoadURL, 32
 - MainTexture, 35
 - MobileRect, 35
 - Name, 34
 - OnGUIMouse, 33
 - OnWebGUI, 33
 - ProcessKey, 33
 - PushToBack, 33
 - Remove, 33
 - ResizeToContents, 34
 - RestartAllViews, 33
 - Show, 33
 - SmartRects, 35
 - Title, 34
 - Transparency, 35
 - TSet, 34
 - UpdateView, 33
 - URL, 34
 - Valid, 35
 - Visible, 35
 - Width, 36
 - windowId, 36
- Valid
 - UWKView, 35
- Visible

UWKView, [35](#)

Width

UWKView, [36](#)

windowId

UWKView, [36](#)