

Using TEdgeBrowser Component and Changes to the TWebBrowser Component

Go Up to VCL

RAD Studio 10.4 brings support for working with web content through the Chromium-based Edge WebView2 browser control in VCL applications via the new TEdgeBrowser component.

TEdgeBrowser supersedes TWebBrowser, which uses the Internet Explorer WebBrowser browser control. However TWebBrowser remains in the VCL component set, with some notable changes.

Preparing to use the Edge Browser component

Because TWebBrowser uses the Operating System-supplied Internet Explorer WebBrowser browser control there is no preparation required; it will work wherever Windows has the Internet Explorer control available.

On the contrary, Microsoft Edge is not an Operating System component at the time. Because of this you need to ensure that these items are installed on your computer before you can run an application that uses it:

- The Microsoft Edge Chromium-based browser currently available from <https://www.microsoftedgeinsider.com/download> (Canary channel version whilst the WebView2 SDK is in pre-release, currently the minimum required version is 82.0.430.0).
- The WebView2 control, currently available through NuGet at <https://www.nuget.org/packages/Microsoft.Web.WebView2> or via GetIt package manager.

This applies to compiling and executing the application on your development machine, and also on the

Contents

Preparing to use the Edge Browser component

Installing the Edge WebView2 package via GetIt

Manually Installing the Edge WebView2 package from NuGet

Project Configuration for WebView2 loader DLLs

Using the Edge Browser component

Migrating from TWebBrowser to TEdgeBrowser

Methods

Properties

Events

Changes in TWebBrowser

TWebBrowser Dual Personality Capabilities

Methods

Read-only Properties

Published Read/Write Properties

Events

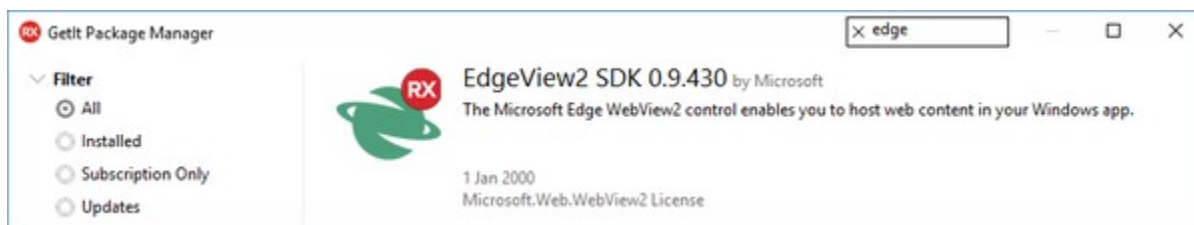
New Edgde-related Demos

end user's machine.

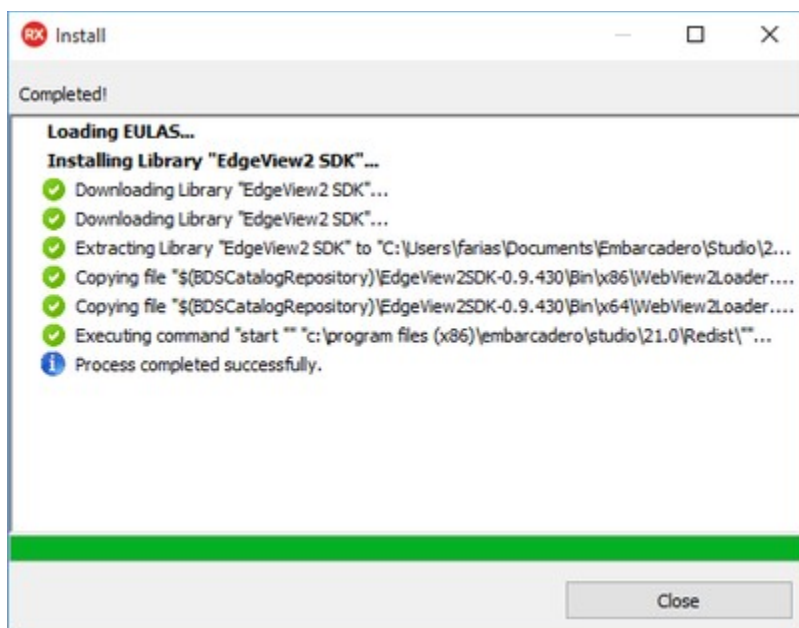
With both these installed, the TEdgeBrowser will function and use the Edge WebView2 browser control to render web content.

Installing the Edge WebView2 package via GetIt

To install the Microsoft WebView2 package, open the GetIt package manager window in the RAD Studio IDE and search for the corresponding entry:



After you accept the license, GetIt will download the package and copy the two WebView2 loader DLLs, for 32 and 64 bit, to the corresponding Redist folders, as the GetIt log indicates:



You can copy those files to your target EXE directories, or to a folder on your path, or follow the steps in the section below "Project Configuration for WebView2 loader DLLs".

Manually Installing the Edge WebView2 package from NuGet

The Edge WebView2 browser control is available from NuGet at <https://www.nuget.org/packages/Microsoft.Web.WebView2> under the name Microsoft.Web.WebView2. The component is currently written against version 0.9.430.

On this page, <https://www.nuget.org/packages/Microsoft.Web.WebView2/0.9.430>, you can click the Download package link to download microsoft.web.webview2.0.9.430.nupkg.

A .nupkg file is really just a .zip file with a different file extension so you can rename it to microsoft.web.webview2.0.9.430.zip and extract the content with the Windows File Explorer, or use 7zip to extract the content and retrieve the required file.

The file you need from the package is WebView2Loader.dll. If you are building a Win32 app you should take build\x86\WebView2Loader.dll, and if you are building a Win64 app you should take build\x64\WebView2Loader.dll.

Project Configuration for WebView2 loader DLLs

Wherever you choose to place the WebView2 loader DLLs you will need to set up a post-build event to copy it into your project's output directory.

For a Win32 target the post-build event should look something like:

```
copy /Y "common_DLL_folder_path\x86\WebView2Loader.dll"  
$(OUTPUTDIR)
```

For a Win64 target use:

```
copy /Y "common_DLL_folder_path\x64\WebView2Loader.dll"  
$(OUTPUTDIR)
```

Using the Edge Browser component

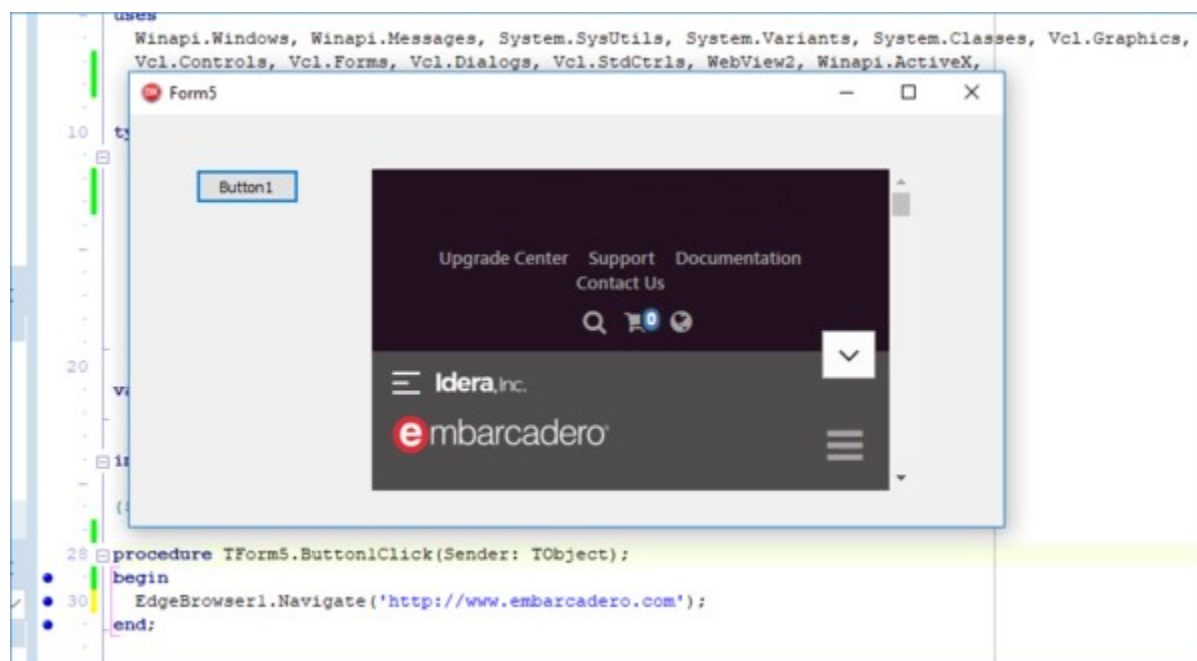
you can use the TEdgeBrowser component in a similar way you use a TWebBrowser, because a number of TEdgeBrowser's methods and properties are much the same to those of TWebBrowser's methods and properties.

Drop the component on a VCL form and size it as appropriate.

To navigate to a URL simply pass that URL to the Navigate method.

Note that the `WebView2` control is created asynchronously. You can instigate the creation either by calling the `Navigate` method, or by calling the `CreateWebView` method. At any point you can tell if the control has been successfully created by using the `WebViewCreated` property.

This is a minimal example of a running application with a single line of code and not other configuration settings:



Additionally, you can use an `OnCreateWebViewCompleted` event handler which gets called when control creation completes, either successfully or unsuccessfully.

If there is no `OnCreateWebViewCompleted` event handler and the `WebView2` browser control cannot be initiated, then an `EEdgeError` exception raises on the main thread, but synchronised onto the main thread from a worker thread.

This is therefore difficult to catch without using a global exception handler, for example, a `TApplicationEvents` component's `OnException` event handler, so an `OnCreateWebViewCompleted` event handler is recommended to simplify this situation.

There are a number of additional events that you can set up event handlers for, which mostly surface events from the underlying `WebView2` browser control. For example, when you navigate to a URL the `OnNavigationStarting` event will fire one or more times based on the target page content. When navigation to a URL concludes the `OnNavigationCompleted` event fires.

Migrating from TWebBrowser to TEdgeBrowser

In order to move from the traditional `TWebBrowser` component to the new Edge-based, `TEdgeBrowser` component you will have to assess which `TWebBrowser` methods, properties, and events you currently use.

There is some commonality between the exposed interface of TWebBrowser and TEdgeBrowser but it is reasonably limited.

The table below shows the closest 'equivalents' in class members for you to change from and to.

It should be noted that the underlying browser control used by TEdgeBrowser, the WebView2 browser control, exposes many fewer events with a narrower focus than the Internet Explorer WebBrowser browser control underlying the TWebBrowser VCL component.

Methods

TWebBrowser method	TEdgeBrowser method
GoBack	GoBack
GoForward	GoForward
GoHome	No equivalent
GoSearch	No equivalent
Navigate	Navigate
Navigate2	Navigate
Refresh	Refresh
Refresh2	Refresh
Stop	Stop
Quit	CloseBrowserProcess
ClientToWindow	No equivalent
PutProperty	No equivalent
GetProperty	No equivalent
QueryStatusWB	No equivalent

ExecWB	No equivalent
ShowBrowserBar	No equivalent

Properties

TWebBrowser property	TEdgeBrowser property
AddressBar	No equivalent
Application	No equivalent
Busy	No direct equivalent
Container	No equivalent
ControllInterface	DefaultInterface is the ICoreWebView2 interface
DefaultInterface	DefaultInterface is the ICoreWebView2 interface
Document	No equivalent
FullName	No direct equivalent, but you can use BrowserProcessID and some Windows APIs
FullScreen	No equivalent
HWND	No direct equivalent but you can use Handle and some Windows APIs
LocationName	DocumentTitle
LocationURL	LocationURL
MenuBar	No equivalent
Name	No equivalent
Offline	No equivalent

Parent	No equivalent
Path	No direct equivalent, but you can use BrowserProcessID and some Windows APIs
ReadyState	No direct equivalent
RegisterAsBrowser	No equivalent
RegisterAsDropTarget	No equivalent
Resizable	No direct equivalent
Silent	The inverse of DefaultScriptDialogsEnabled along with the OnScriptDialogOpening event
StatusBar	StatusBarEnabled
TheaterMode	No equivalent
ToolBar	No equivalent
TopLevelContainer	No equivalent
Type_	No equivalent
Visible	Visible

Events

TWebBrowser event	TEdgeBrowser event
OnBeforeNavigate2	OnNavigationStarting
OnBeforeScriptExecute	No equivalent
OnClientToHostWindow	No equivalent
OnCommandStateChange	No direct equivalent but you can use OnHistoryChanged

OnDocumentComplete	OnNavigationCompleted
OnDownloadBegin	No equivalent
OnDownloadComplete	No equivalent
OnFileDownload	No equivalent
OnFullScreen	No equivalent
OnMenuBar	No equivalent
OnNavigateComplete2	OnNavigationCompleted
OnNavigateError	OnNavigationComplete, noting the IsSuccess parameter
OnNewProcess	No equivalent
OnNewWindow2	OnNewWindowRequested
OnNewWindow3	OnNewWindowRequested
OnPrintTemplateInstantiation	No equivalent
OnPrintTemplateTeardown	No equivalent
OnPrivacyImpactedStateChange	No equivalent
OnProgressChange	No equivalent
OnPropertyChange	No equivalent
OnQuit	No equivalent
OnRedirectXDomainBlocked	No equivalent
OnShowScriptError	No equivalent

OnSetPhishingFilterStatus	No equivalent
OnSetSecureLockIcon	No equivalent
OnStatusBar	No equivalent
OnStatusTextChange	No equivalent
OnTheaterMode	No equivalent
OnThirdPartyUrlBlocked	No equivalent
OnTitleChange	OnDocumentTitleChanged
OnToolBar	No equivalent
OnUpdatePageStatus	No equivalent
OnVisible	No equivalent
OnWebWorkerFinsihed	No equivalent
OnWebWorkerStarted	No equivalent
OnWindowClosing	No equivalent
OnWindowSetHeight	No equivalent
OnWindowSetLeft	No equivalent
OnWindowSetResizable	No equivalent
OnWindowSetTop	No equivalent
OnWindowSetWidth	No equivalent
OnWindowStateChanged	No equivalent

Changes in TWebBrowser

The VCL TWebBrowser component still performs the same job as always, which is to render web content using the Internet Explorer WebBrowser browser control.

However, in addition to this default behaviour TWebBrowser has two new properties; the first property can be used to ask it to apply the newer Edge (Chromium) WebView2 browser control instead. This is achieved by the implementation using an embedded TEdgeBrowser component.

By using the property to use the Edge WebView2 browser control you can immediately benefit from a current, maintained browser, which will have security fixes made available by Microsoft.

However, the flip side of this is that the WebView2 browser control is not an integral part of every Windows installation.

The property SelectedEngine indicates which browser control you desire or require. The supported values are:

- TWebBrowser.TSelectedEngine.IEOnly: the TWebBrowser functions as it always has, employing the IE WebBrowser control.
- TWebBrowser.TSelectedEngine.EdgeIfAvailable: the TWebBrowser uses the Edge WebView2 browser control if possible, otherwise it falls back to the traditional IE WebBrowser control.
- TWebBrowser.TSelectedEngine.EdgeOnly: the TWebBrowser attempts to use the Edge WebView2 browser control, however if this is not possible then no browsing is possible as there is no fallback option.

Some feedback is available via the ActiveEngine property, which offers these values:

- TWebBrowser.TActiveEngine.None: no browser control is in use; if EdgeOnly was requested via SelectedEngine then the Edge browser could not be instantiated.
- TWebBrowser.TActiveEngine.NoneYet: no browser control is in use yet, but an attempt is being made to initialize the Edge WebView2 browser control.
- TWebBrowser.TActiveEngine.IE: the Internet Explorer WebBrowser browser control is in use.
- TWebBrowser.TActiveEngine.Edge: the Edge WebView2 browser control is in use.

Attention: Just as with the TEdgeBrowser component, in order for TWebBrowser to successfully use the Edge WebView2 browser control there are two requirements:

- The Edge (Chromium) browser is installed.
- The Edge WebView2 control loader DLL (part of the Microsoft WebView2 SDK) is available, for example in the directory your project executable is output to.

To install the Microsoft Edge WebView2 control please follow the instructions on the [TEdgeBrowser documentation page](http://docwiki.embarcadero.com/RADStudio/Denali/en/Using_TEdgeBrowser_Component_and_Changes_to_the_TWebBrowser_Component#Preparing_to_use_the_Edge_Browser_component%7C) (http://docwiki.embarcadero.com/RADStudio/Denali/en/Using_TEdgeBrowser_Component_and_Changes_to_the_TWebBrowser_Component#Preparing_to_use_the_Edge_Browser_component%7C).

TWebBrowser Dual Personality Capabilities

When using TWebBrowser in its 'Edge personality' mode (SelectedEngine was set to either EdgelfAvailable or EdgeOnly, and ActiveEngine has been changed to Edge) its various properties, methods, and events have been updated to operate as appropriately as they can.

Note that many of these properties, methods, and events are directly from the Internet Explorer WebBrowser control interfaces, so there are a number of them that now either return a default value, or raise an exception when running in Edge mode.

The following lists point out the behavior when running in Edge mode:

Methods

GoBack	Goes to previous page in browser history
GoForward	Goes to next page in browser history
GoHome	Raises an ENotImplemented exception
GoSearch	Raises an ENotImplemented exception
Navigate	The overload with only a URL parameter navigates to that URL, but the other overloads raise an ENotImplemented exception
Refresh	Reloads the current page
Refresh2	Reloads the current page
Stop	Stops the current navigation activity
Quit	Terminates the browser control's supporting Edge process
ClientToWindow	Raises an ENotImplemented exception
PutProperty	Raises an ENotImplemented exception

GetProperty	Raises an ENotImplemented exception
Navigate2	The overload with only a URL parameter navigates to that URL, but the other overloads raise an ENotImplemented exception
QueryStatusWB	Raises an ENotImplemented exception
ExecWB	Raises an ENotImplemented exception
ShowBrowserBar	Raises an ENotImplemented exception

Read-only Properties

ControllInterface	Returns nil
DefaultInterface	NReturns nil
Application	Returns nil
Parent	Returns nil
Container	Returns nil
Document	Returns nil
TopLevelContainer	Returns True
type_	Returns an empty string
LocationName	Returns the current <u>document title</u>
LocationURL	Returns the <u>current URL</u>
Busy	Returns True when navigation is active, otherwise False
Name	Returns an empty string
HWND	Returns the window handle of the WebView2 browser control

FullName	Returns the full executable path of the Edge browser process being used by the WebView2 control
Path	Returns the full path of the folder of the Edge browser process being used by the WebView2 control
ReadyState	Returns one of a subset of the possible values, based on the state of the Edge control: READYSTATE_UNINITIALIZED, READYSTATE_LOADING or READYSTATE_COMPLETE

Published Read/Write Properties

Visible	Sets and returns the embedded Edge component's Visible property
StatusBar	Sets and returns the embedded Edge component's <u>StatusBarEnabled</u> property
StatusText	Ignores set value and returns an empty string
ToolBar	Ignores set value and returns 0
MenuBar	Ignores set value and returns False
FullScreen	Ignores set value and returns False
Offline	Ignores set value and returns False
Silent	Sets and returns the inverse value to/from the embedded Edge component's <u>DefaultScriptDialogsEnabled</u> property
RegisterAsBrowser	Ignores set value and returns False
RegisterAsDropTarget	Ignores set value and returns False
TheaterMode	Ignores set value and returns False
AddressBar	Ignores set value and returns False
Resizable	Ignores set value and returns True

Events

OnStatusTextChange	Not called
OnProgressChange	Not called
OnCommandStateChange	Fired when the Edge component's <u>OnHistoryChanged</u> event fires
OnDownloadBegin	Not called
OnDownloadComplete	Not called
OnTitleChange	Fired when the Edge component's <u>OnDocumentTitleChanged</u> event fires
OnPropertyChange	Not called
OnBeforeNavigate2	Fired when the Edge component's <u>OnNavigationStarting</u> event fires
OnNewWindow2	Fired when the Edge component's <u>OnNewWindowRequested</u> event fires
OnNavigateComplete2	Fired when the Edge component's <u>OnNavigationCompleted</u> event fires
OnDocumentComplete	Fired when the Edge component's <u>OnNavigationCompleted</u> event fires
OnQuit	Not called
OnVisible	Not called
OnToolBar	Not called
OnMenuBar	Not called
OnStatusBar	Not called
OnFullScreen	Not called

OnTheaterMode	Not Called
OnWindowSetResizable	Not called
OnWindowSetLeft	Not called
OnWindowSetTop	Not called
OnWindowSetWidth	Not called
OnWindowSetHeight	Not called
OnWindowClosing	Not called
OnClientToHostWindow	Not called
OnSetSecureLockIcon	Not called
OnFileDownload	Not called
OnNavigateError	Not called
OnPrintTemplateInstantiation	Not called
OnPrintTemplateTeardown	Not called
OnUpdatePageStatus	Not called
OnPrivacyImpactedStateChange	Not called
OnNewWindow3	Fired when the Edge component's OnNewWindowRequested event fires
OnSetPhishingFilterStatus	Not called
OnWindowStateChanged	Not called
OnNewProcess	Not called

OnThirdPartyUrlBlocked	Not called
OnRedirectXDomainBlocked	Not called
OnBeforeScriptExecute	Not called
OnWebWorkerStarted	Not called
OnWebWorkerFinsihed	Not called
OnShowScriptError	Not called

New Edgde-related Demos

There are new demos installed with the product showcasin the new Edge and the updated TWebBrowser component. You can find them in the following folders:

- Demos\Object Pascal\VCL\WebBrowser\Edge
- Demos\Object Pascal\VCL\WebBrowser\InternetExplorer
- Demos\CPP\VCL\WebBrowser\Edge

Retrieved from "http://docwiki.embarcadero.com/RADStudio/Sydney/e/index.php?title=Using_TEdgeBrowser_Component_and_Changes_to_the_TWebBrowser_Component&oldid=271416"

This page was last edited on 21 May 2020, at 21:33.