

## Coherent UI for Unity3D

1.8.1



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Brief overview of Coherent UI	1
1.2	Differences between Desktop and Mobile version	1
<b>2</b>	<b>Installation</b>	<b>3</b>
2.1	Prerequisites	3
2.2	Package contents and structure	3
2.3	Usage of the package	4
2.4	Switching between build targets	5
2.5	Trial version registration	6
<b>3</b>	<b>Samples</b>	<b>7</b>
3.1	Hello, Coherent UI	7
3.2	Facebook integration	9
3.3	Menu And HUD	10
3.4	Binding Demo	12
3.5	Archive resource	16
3.6	Mobile Input	17
3.7	Oculus	19
3.8	Mobile Surface	19
3.9	Live Game Views	20
3.10	IME Sample	20
<b>4</b>	<b>Programmer's API</b>	<b>23</b>
4.1	CoherentUISystem properties for Desktop	23
4.2	CoherentUISystem properties for Mobile	23
4.3	CoherentUIView properties for Desktop	24
4.4	CoherentUIView properties for Mobile	25
4.5	CoherentUILiveGameView properties for Desktop	25
<b>5</b>	<b>Important points</b>	<b>27</b>
5.1	Binding	27
5.2	CoherentMethod attribute for .NET scripts	27

5.3	Namespaces	29
5.4	Subclassing CoherentUIView and UnityViewListener	29
5.5	Coherent UI system lifetime	30
5.6	Customizing initialization of the Coherent UI System	30
5.7	Update cycle	31
5.8	Input forwarding - Desktop	31
5.9	Input forwarding - Mobile	31
5.10	Mobile Preview	32
5.11	Custom file handler	32
5.12	UI Resources	33
5.13	Click-to-focus Views - Desktop only	33
5.14	View Filters - Desktop only	34
5.15	Hit testing - Desktop only	34
5.16	Mobile Surface Views	35
5.17	Logging	35
5.18	Live Game Views	35
<b>6</b>	<b>Requirements</b>	<b>37</b>
<b>7</b>	<b>Namespace Documentation</b>	<b>39</b>
7.1	Package Coherent	39
7.2	Package Coherent.UI	39
7.2.1	Enumeration Type Documentation	41
7.2.1.1	CertificateStatus	41
7.2.1.2	CursorTypes	41
7.2.1.3	Direct3DInterfaceType	41
7.2.1.4	DownloadErrorType	42
7.2.1.5	FileSelectionMode	42
7.2.1.6	JavaScriptMessageType	42
7.2.1.7	MediaStreamType	42
7.2.1.8	RenderingDeviceType	42
7.2.1.9	ScriptCallErrorType	42
7.2.1.10	SystemErrorType	42
7.2.1.11	TextInputControlType	42
7.2.1.12	UserAgentType	42
7.2.1.13	ViewErrorType	43
7.2.1.14	ViewType	43
7.3	Package Coherent.UI.Binding	43
7.3.1	Enumeration Type Documentation	44
7.3.1.1	PropertyBindingFlags	44
7.3.1.2	ValueType	44

<b>8</b>	<b>Class Documentation</b>	<b>45</b>
8.1	Coherent.UI.BoundEventHandle Struct Reference	45
8.1.1	Detailed Description	45
8.2	Coherent.UI.Binding.BoundObject Struct Reference	45
8.2.1	Detailed Description	45
8.2.2	Member Function Documentation	45
8.2.2.1	BindMethods	45
8.3	Coherent.UI.BrowserView Class Reference	46
8.3.1	Detailed Description	46
8.3.2	Member Function Documentation	46
8.3.2.1	GetHistory	46
8.3.2.2	GetViewType	46
8.3.2.3	GoBack	46
8.3.2.4	GoForward	46
8.4	Coherent.UI.BrowserViewListener Class Reference	47
8.4.1	Detailed Description	49
8.4.2	Member Function Documentation	49
8.4.2.1	CreateSurface	49
8.4.2.2	DestroySurface	50
8.4.2.3	OnBindingsReleased	51
8.4.2.4	OnCallback	51
8.4.2.5	OnCaretRectChanged	51
8.4.2.6	OnCertificateError	51
8.4.2.7	OnCursorChanged	52
8.4.2.8	OnDraw	52
8.4.2.9	OnError	52
8.4.2.10	OnFailLoad	52
8.4.2.11	OnFileSelectRequest	52
8.4.2.12	OnFinishLoad	53
8.4.2.13	OnGetAuthCredentials	53
8.4.2.14	OnIMEShouldCancelComposition	53
8.4.2.15	OnJavaScriptMessage	53
8.4.2.16	OnNavigateTo	54
8.4.2.17	OnReadyForBindings	55
8.4.2.18	OnRequestMediaStream	55
8.4.2.19	OnScriptMessage	55
8.4.2.20	OnStartLoading	55
8.4.2.21	OnStopLoading	55
8.4.2.22	OnTextInputTypeChanged	56
8.4.2.23	OnURLRequest	57

8.4.2.24	OnViewCreated	57
8.4.2.25	Release	57
8.4.3	Event Documentation	57
8.4.3.1	BindingsReleased	57
8.4.3.2	Callback	57
8.4.3.3	CaretRectChanged	57
8.4.3.4	CertificateError	57
8.4.3.5	CursorChanged	58
8.4.3.6	Draw	58
8.4.3.7	Error	58
8.4.3.8	FailLoad	58
8.4.3.9	FileSelectRequest	58
8.4.3.10	FinishLoad	58
8.4.3.11	GetAuthCredentials	58
8.4.3.12	IMEShouldCancelComposition	58
8.4.3.13	JavaScriptMessage	58
8.4.3.14	NavigateTo	58
8.4.3.15	ReadyForBindings	58
8.4.3.16	RequestMediaStream	58
8.4.3.17	ScriptMessage	59
8.4.3.18	StartLoading	59
8.4.3.19	StopLoading	59
8.4.3.20	TextInputTypeChanged	59
8.4.3.21	URLRequest	59
8.4.3.22	ViewCreated	59
8.5	Coherent.UI.BrowserViewListenerBase Class Reference	59
8.5.1	Detailed Description	59
8.6	Coherent.UI.Binding.CallbackArguments Class Reference	59
8.6.1	Detailed Description	60
8.6.2	Member Data Documentation	60
8.6.2.1	Arguments	60
8.7	Coherent.UI.Certificate Class Reference	60
8.7.1	Detailed Description	60
8.7.2	Property Documentation	60
8.7.2.1	Fingerprint	60
8.7.2.2	FingerprintCA	61
8.7.2.3	Issuer	61
8.7.2.4	SerialNumber	61
8.7.2.5	Subject	61
8.7.2.6	ValidEnd	61

8.7.2.7	ValidStart	61
8.8	Coherent.UI.CertificatePrincipal Class Reference	61
8.8.1	Detailed Description	62
8.8.2	Property Documentation	62
8.8.2.1	CommonName	62
8.8.2.2	CountryName	62
8.8.2.3	DomainComponents	62
8.8.2.4	LocalityName	62
8.8.2.5	OrganizationNames	62
8.8.2.6	OrganizationUnitNames	62
8.8.2.7	StateOrProvinceName	62
8.8.2.8	StreetAddresses	62
8.9	Coherent.UI.ChildViewInfo Class Reference	62
8.9.1	Detailed Description	63
8.9.2	Property Documentation	63
8.9.2.1	Listener	63
8.9.2.2	NewViewInfo	63
8.10	Coherent.UI.Binding.CoherentProperty Class Reference	63
8.10.1	Detailed Description	63
8.10.2	Constructor & Destructor Documentation	63
8.10.2.1	CoherentProperty	63
8.10.2.2	CoherentProperty	63
8.11	Coherent.UI.Binding.CoherentType Class Reference	63
8.11.1	Detailed Description	64
8.12	Coherent.UI.Download Class Reference	64
8.12.1	Detailed Description	64
8.12.2	Member Function Documentation	64
8.12.2.1	Cancel	64
8.12.2.2	Destroy	65
8.12.2.3	GetDownloadedBytesSoFar	65
8.12.2.4	GetFilename	65
8.12.2.5	GetId	65
8.12.2.6	GetReferrer	65
8.12.2.7	GetTotalBytes	65
8.12.2.8	GetURL	65
8.12.2.9	HasFailed	66
8.12.2.10	IsCancelled	66
8.12.2.11	IsComplete	66
8.13	Coherent.UI.EventListener Class Reference	66
8.13.1	Detailed Description	66

8.13.2	Member Function Documentation	66
8.13.2.1	OnDownloadComplete	66
8.13.2.2	OnDownloadFailed	67
8.13.2.3	OnDownloadStarted	67
8.13.2.4	OnError	67
8.13.2.5	SystemReady	67
8.14	Coherent.UI.EventModifiersState Class Reference	67
8.14.1	Detailed Description	67
8.15	Coherent.UI.EventMouseModifiersState Class Reference	67
8.15.1	Detailed Description	67
8.16	Coherent.UI.FileHandler Class Reference	67
8.16.1	Detailed Description	68
8.16.2	Member Function Documentation	68
8.16.2.1	ReadFile	68
8.16.2.2	WriteFile	68
8.17	Coherent.UI.HTTPHeader Class Reference	68
8.17.1	Detailed Description	68
8.17.2	Property Documentation	68
8.17.2.1	Content	68
8.17.2.2	Field	69
8.18	Coherent.UI.ILogHandler Class Reference	69
8.18.1	Detailed Description	69
8.18.2	Member Function Documentation	69
8.18.2.1	Assert	69
8.18.2.2	WriteLog	69
8.19	Coherent.UI.ImageData Class Reference	69
8.19.1	Detailed Description	70
8.19.2	Member Function Documentation	70
8.19.2.1	Destroy	70
8.19.2.2	Update	70
8.20	Coherent.UI.Binding.InvalidValueCastException Class Reference	70
8.20.1	Detailed Description	70
8.21	Coherent.UI.KeyEventData Class Reference	70
8.21.1	Detailed Description	71
8.21.2	Property Documentation	71
8.21.2.1	IsAutoRepeat	71
8.21.2.2	IsNumPad	71
8.21.2.3	KeyCode	71
8.21.2.4	Modifiers	71
8.21.2.5	Type	71



8.22 Coherent.UI.MediaStreamDevice Class Reference . . . . .	71
8.22.1 Detailed Description . . . . .	71
8.22.2 Property Documentation . . . . .	72
8.22.2.1 DeviceId . . . . .	72
8.22.2.2 Name . . . . .	72
8.22.2.3 Type . . . . .	72
8.23 Coherent.UI.MediaStreamRequest Class Reference . . . . .	72
8.23.1 Detailed Description . . . . .	72
8.23.2 Member Function Documentation . . . . .	72
8.23.2.1 Respond . . . . .	72
8.23.3 Property Documentation . . . . .	72
8.23.3.1 Devices . . . . .	72
8.24 Coherent.UI.MouseEventData Class Reference . . . . .	73
8.24.1 Detailed Description . . . . .	73
8.24.2 Property Documentation . . . . .	73
8.24.2.1 Button . . . . .	73
8.24.2.2 Modifiers . . . . .	73
8.24.2.3 MouseModifiers . . . . .	73
8.24.2.4 Type . . . . .	73
8.24.2.5 WheelX . . . . .	73
8.24.2.6 WheelY . . . . .	74
8.24.2.7 X . . . . .	74
8.24.2.8 Y . . . . .	74
8.25 Coherent.UI.ResourceData Class Reference . . . . .	74
8.25.1 Detailed Description . . . . .	74
8.25.2 Member Function Documentation . . . . .	74
8.25.2.1 GetBuffer . . . . .	74
8.25.2.2 GetSize . . . . .	74
8.25.2.3 SignalFailure . . . . .	75
8.25.2.4 SignalSuccess . . . . .	75
8.26 Coherent.UI.ResourceResponse Class Reference . . . . .	75
8.26.1 Detailed Description . . . . .	75
8.26.2 Member Function Documentation . . . . .	75
8.26.2.1 GetBuffer . . . . .	75
8.26.2.2 SignalFailure . . . . .	75
8.26.2.3 SignalSuccess . . . . .	76
8.27 Coherent.UI.SurfaceResponse Class Reference . . . . .	76
8.27.1 Detailed Description . . . . .	76
8.27.2 Member Function Documentation . . . . .	76
8.27.2.1 Signal . . . . .	76

8.28 Coherent.UI.SystemError Class Reference . . . . .	76
8.28.1 Detailed Description . . . . .	76
8.28.2 Property Documentation . . . . .	77
8.28.2.1 Error . . . . .	77
8.28.2.2 ErrorCode . . . . .	77
8.29 Coherent.UI.SystemSettings Class Reference . . . . .	77
8.29.1 Detailed Description . . . . .	77
8.29.2 Property Documentation . . . . .	78
8.29.2.1 CachePath . . . . .	78
8.29.2.2 CookiesResource . . . . .	78
8.29.2.3 CustomUserAgentString . . . . .	78
8.29.2.4 D3DObject . . . . .	78
8.29.2.5 FocusWindow . . . . .	78
8.29.2.6 HostDirectory . . . . .	78
8.29.2.7 HTML5LocalStoragePath . . . . .	78
8.29.2.8 MinidumpFolder . . . . .	78
8.30 Coherent.UI.SystemSettingsBase Class Reference . . . . .	78
8.30.1 Detailed Description . . . . .	79
8.30.2 Property Documentation . . . . .	79
8.30.2.1 AllowCookies . . . . .	79
8.30.2.2 CustomUserAgentString . . . . .	79
8.30.2.3 D3DInterfaceType . . . . .	79
8.30.2.4 DebuggerPort . . . . .	79
8.30.2.5 DisableWebSecurity . . . . .	80
8.30.2.6 EnableProxy . . . . .	80
8.30.2.7 EnableSupportForProprietaryCodecs . . . . .	80
8.30.2.8 ForceDisablePluginFullscreen . . . . .	80
8.30.2.9 UserAgent . . . . .	80
8.30.2.10 WriteMinidumps . . . . .	80
8.31 Coherent.UI.TouchEventData Class Reference . . . . .	80
8.31.1 Detailed Description . . . . .	81
8.31.2 Property Documentation . . . . .	81
8.31.2.1 HasRadiusInformation . . . . .	81
8.31.2.2 Id . . . . .	81
8.31.2.3 RadiusX . . . . .	81
8.31.2.4 RadiusY . . . . .	81
8.31.2.5 Type . . . . .	81
8.31.2.6 X . . . . .	81
8.31.2.7 Y . . . . .	81
8.32 Coherent.UI.UISystem Class Reference . . . . .	81

8.32.1 Detailed Description . . . . .	82
8.32.2 Member Function Documentation . . . . .	82
8.32.2.1 CreateView . . . . .	82
8.32.2.2 CreateView . . . . .	82
8.32.2.3 FetchSurfaces . . . . .	82
8.32.2.4 Uninitialize . . . . .	82
8.32.2.5 Update . . . . .	83
8.33 Coherent.UI.Binding.UnsupportedPrimitiveTypeException Class Reference . . . . .	83
8.33.1 Detailed Description . . . . .	83
8.34 Coherent.UI.URLRequest Class Reference . . . . .	83
8.34.1 Detailed Description . . . . .	84
8.34.2 Member Function Documentation . . . . .	84
8.34.2.1 AppendExtraHeaderValue . . . . .	84
8.34.2.2 Deny . . . . .	84
8.34.2.3 GetExtraHeader . . . . .	84
8.34.2.4 GetExtraHeaders . . . . .	84
8.34.2.5 GetMethod . . . . .	84
8.34.2.6 GetReferrer . . . . .	85
8.34.2.7 GetURL . . . . .	85
8.34.2.8 RemoveExtraHeader . . . . .	85
8.34.2.9 SetExtraHeader . . . . .	85
8.34.2.10 SetExtraHeaders . . . . .	85
8.34.2.11 SetMethod . . . . .	85
8.34.2.12 SetReferrer . . . . .	85
8.34.2.13 SetURL . . . . .	86
8.35 Coherent.UI.Binding.Value Class Reference . . . . .	86
8.35.1 Detailed Description . . . . .	87
8.35.2 Constructor & Destructor Documentation . . . . .	87
8.35.2.1 Value . . . . .	87
8.35.2.2 Value . . . . .	87
8.35.2.3 Value . . . . .	87
8.35.2.4 Value . . . . .	88
8.35.2.5 Value . . . . .	88
8.35.2.6 Value . . . . .	88
8.35.2.7 Value . . . . .	88
8.35.2.8 Value . . . . .	88
8.35.2.9 Value . . . . .	88
8.35.2.10 Value . . . . .	89
8.35.2.11 Value . . . . .	90
8.35.2.12 Value . . . . .	90

8.35.3 Member Function Documentation . . . . .	90
8.35.3.1 Equals . . . . .	90
8.35.3.2 GetHashCode . . . . .	90
8.35.3.3 operator bool . . . . .	90
8.35.3.4 operator byte . . . . .	90
8.35.3.5 operator Dictionary< string, Value > . . . . .	91
8.35.3.6 operator double . . . . .	91
8.35.3.7 operator float . . . . .	91
8.35.3.8 operator int . . . . .	91
8.35.3.9 operator sbyte . . . . .	92
8.35.3.10 operator string . . . . .	93
8.35.3.11 operator uint . . . . .	93
8.35.3.12 operator Value[] . . . . .	93
8.35.3.13 operator!= . . . . .	93
8.35.3.14 operator== . . . . .	94
8.35.4 Member Data Documentation . . . . .	95
8.35.4.1 Type . . . . .	95
8.36 Coherent.UI.Binding.ValueObject Class Reference . . . . .	95
8.36.1 Detailed Description . . . . .	95
8.37 Coherent.UI.View Class Reference . . . . .	95
8.37.1 Detailed Description . . . . .	98
8.37.2 Member Function Documentation . . . . .	98
8.37.2.1 AuthCredentialsResponse . . . . .	98
8.37.2.2 BindCall . . . . .	98
8.37.2.3 CreateImageData . . . . .	98
8.37.2.4 Destroy . . . . .	98
8.37.2.5 DownloadUrl . . . . .	98
8.37.2.6 DrawSurface . . . . .	99
8.37.2.7 ExecuteScript . . . . .	99
8.37.2.8 FetchMouseOnUIQuery . . . . .	99
8.37.2.9 FetchSurface . . . . .	99
8.37.2.10 GetClickThroughAlphaThreshold . . . . .	99
8.37.2.11 GetCurentViewPath . . . . .	99
8.37.2.12 GetHeight . . . . .	99
8.37.2.13 GetLastFailedPath . . . . .	100
8.37.2.14 GetLastLoadedPath . . . . .	100
8.37.2.15 GetLastRequestedPath . . . . .	100
8.37.2.16 GetTargetFramerate . . . . .	100
8.37.2.17 GetViewType . . . . .	100
8.37.2.18 GetWidth . . . . .	100

8.37.2.19 HasMouseQueryFinished . . . . .	101
8.37.2.20 IMEActivate . . . . .	101
8.37.2.21 IMECancelComposition . . . . .	101
8.37.2.22 IMEConfirmComposition . . . . .	101
8.37.2.23 IMEIsActive . . . . .	101
8.37.2.24 IMESetComposition . . . . .	101
8.37.2.25 InterceptURLRequests . . . . .	102
8.37.2.26 IsMouseOnView . . . . .	102
8.37.2.27 IsOnDemand . . . . .	102
8.37.2.28 IssueMouseOnUIQuery . . . . .	102
8.37.2.29 IsTransparent . . . . .	103
8.37.2.30 KeyEvent . . . . .	103
8.37.2.31 KillFocus . . . . .	103
8.37.2.32 Load . . . . .	103
8.37.2.33 MouseActivate . . . . .	103
8.37.2.34 MouseEvent . . . . .	103
8.37.2.35 PeekFrameReady . . . . .	103
8.37.2.36 Redraw . . . . .	103
8.37.2.37 RegisterForEvent . . . . .	104
8.37.2.38 Reload . . . . .	105
8.37.2.39 ReplyToJavaScriptMessage . . . . .	105
8.37.2.40 RequestFrame . . . . .	105
8.37.2.41 RequestFrame . . . . .	105
8.37.2.42 Resize . . . . .	105
8.37.2.43 SetClickThroughAlphaThreshold . . . . .	106
8.37.2.44 SetFocus . . . . .	106
8.37.2.45 SetScriptError . . . . .	106
8.37.2.46 SetTargetFramerate . . . . .	106
8.37.2.47 SupportsClickThrough . . . . .	106
8.37.2.48 TouchEvent . . . . .	106
8.37.2.49 TriggerEvent . . . . .	107
8.37.2.50 UnbindCall . . . . .	107
8.37.2.51 UnbindObject . . . . .	107
8.37.2.52 UnregisterFromEvent . . . . .	107
8.37.2.53 UsesSharedMemory . . . . .	107
8.38 Coherent.UI.ViewError Class Reference . . . . .	107
8.38.1 Detailed Description . . . . .	108
8.38.2 Property Documentation . . . . .	108
8.38.2.1 Error . . . . .	108
8.38.2.2 ErrorCode . . . . .	108

8.39 Coherent.UI.ViewInfo Class Reference . . . . .	108
8.39.1 Detailed Description . . . . .	109
8.39.2 Property Documentation . . . . .	109
8.39.2.1 ClickThroughAlphaThreshold . . . . .	109
8.39.2.2 ControlTimeOnDemand . . . . .	109
8.39.2.3 ForceSoftwareRendering . . . . .	109
8.39.2.4 Height . . . . .	109
8.39.2.5 IsOnDemand . . . . .	109
8.39.2.6 IsTransparent . . . . .	109
8.39.2.7 SupportClickThrough . . . . .	109
8.39.2.8 TargetFrameRate . . . . .	109
8.39.2.9 UsesSharedMemory . . . . .	109
8.39.2.10 Width . . . . .	110
8.40 Coherent.UI.ViewListener Class Reference . . . . .	110
8.40.1 Detailed Description . . . . .	112
8.40.2 Member Function Documentation . . . . .	112
8.40.2.1 CreateSurface . . . . .	112
8.40.2.2 DestroySurface . . . . .	113
8.40.2.3 OnBindingsReleased . . . . .	113
8.40.2.4 OnCallback . . . . .	113
8.40.2.5 OnCaretRectChanged . . . . .	113
8.40.2.6 OnCertificateError . . . . .	113
8.40.2.7 OnCursorChanged . . . . .	114
8.40.2.8 OnDraw . . . . .	114
8.40.2.9 OnError . . . . .	114
8.40.2.10 OnFailLoad . . . . .	114
8.40.2.11 OnFileSelectRequest . . . . .	115
8.40.2.12 OnFinishLoad . . . . .	115
8.40.2.13 OnGetAuthCredentials . . . . .	115
8.40.2.14 OnIMEShouldCancelComposition . . . . .	115
8.40.2.15 OnJavaScriptMessage . . . . .	115
8.40.2.16 OnNavigateTo . . . . .	116
8.40.2.17 OnReadyForBindings . . . . .	116
8.40.2.18 OnRequestMediaStream . . . . .	116
8.40.2.19 OnScriptMessage . . . . .	116
8.40.2.20 OnStartLoading . . . . .	116
8.40.2.21 OnStopLoading . . . . .	117
8.40.2.22 OnTextInputTypeChanged . . . . .	117
8.40.2.23 OnURLRequest . . . . .	117
8.40.2.24 OnViewCreated . . . . .	117

8.40.2.25 Release	117
8.40.3 Event Documentation	117
8.40.3.1 BindingsReleased	117
8.40.3.2 Callback	117
8.40.3.3 CaretRectChanged	118
8.40.3.4 CertificateError	118
8.40.3.5 CursorChanged	118
8.40.3.6 Draw	118
8.40.3.7 Error	118
8.40.3.8 FailLoad	118
8.40.3.9 FileSelectRequest	118
8.40.3.10 FinishLoad	118
8.40.3.11 GetAuthCredentials	118
8.40.3.12 IMEShouldCancelComposition	118
8.40.3.13 JavaScriptMessage	118
8.40.3.14 NavigateTo	118
8.40.3.15 ReadyForBindings	119
8.40.3.16 RequestMediaStream	119
8.40.3.17 ScriptMessage	119
8.40.3.18 StartLoading	119
8.40.3.19 StopLoading	119
8.40.3.20 TextInputTypeChanged	119
8.40.3.21 URLRequest	119
8.40.3.22 ViewCreated	119
8.41 Coherent.UI.ViewListenerBase Class Reference	119
8.41.1 Detailed Description	120
8.41.2 Member Enumeration Documentation	120
8.41.2.1 MessageLevel	120





# Chapter 1

## Introduction

This guide describes the features of the [Coherent UI](#) integration in Unity3D. A basic understanding of the Unity3D engine, as well as HTML/JavaScript is assumed. Having basic knowledge of the C++ API and design of [Coherent UI](#) may be advantageous. To familiarize yourself with [Coherent UI](#), please read the main documentation file or visit [Coherent UI](#) website.

### 1.1 Brief overview of Coherent UI

[Coherent UI](#) is a modern user interface middleware solution that allows you to integrate HTML pages built with CSS and JavaScript in your game. The communication between your game and the HTML engine is done through the UI System component. Each HTML page is called a *View*. The *View* component allows you to perform operations on the page, such as resizing, navigating to a different URL, sending input events, executing custom JavaScript code and so on. You can create a view through the `CreateView` method of the UI System component. It requires you to supply some initialization parameters, such as width, height, initial URL, etc. It also requires an instance of a *View Listener*. [Coherent UI](#) is highly asynchronous by design, meaning that when you change the URL of a *View*, for example, the function will return immediately and you will receive a notification when the URL has actually changed. The `ViewListener` is the class that receives such notifications for a specific view - when the URL was changed, the page you're trying to open requires authentication details, etc.

### 1.2 Differences between Desktop and Mobile version

[Coherent UI](#) can be divided conceptually in two libraries - [Coherent UI Desktop](#) (for Windows & MacOSX) and [Coherent UI Mobile](#) (for iOS and Android). Due to platform limitations the two have a different subset of features. Namely the Mobile version has some limitations while the Desktop version is fully featured.

#### Warning

The current version of [Coherent UI](#) for Android has known issues when using the software keyboard alongside with Unity's one. Keyboards other than the default may not work as expected (e.g. Swype).

Mobile limitations on [Coherent UI Mobile](#) include:

- You are only able to create 2D views on-top of your game for HUDs or in-game browsers. Views splatted on 3D surfaces in the game world are not supported due to platform limitations on iOS and Android.
- Input management must be implemented through minor changes in the HTML & JS and is not pixel perfect but HTML element-based.
- Bound objects are currently missing from the binding
- No on-demand views and frame-rate control

- When writing scripts, the IDE's autocompletion will use the API for the Desktop version. There are a few methods that are available only for Desktop and others only for Mobile, so you won't see the mobile ones in your code editor. The classes for the mobile version are named the same as the desktop version but they're in a different namespace, *Coherent.UI.Mobile*. You can inspect the objects in that namespace to check the availability of the method you're looking for. In MonoDevelop you can inspect an object by using the "Go to declaration (F12)" feature.

Other than that the API has been kept 100% compatible between the Desktop & Mobile versions. The Core binding, Resource management and View management are the same. *Coherent UI Mobile* supports both device builds and simulator ones.

## Chapter 2

# Installation

[Coherent UI](#) for Unity3D is distributed in a **unitypackage** file. You can import this package in your project by either double-clicking on it, or by importing it through Unity in the *Project* window.

### Note

**Versions 1.5.3 and prior:** After importing the package, you have to **install** the assets provided. This is done by the *Assets* → [Coherent UI](#) → *Install Coherent UI* menu entry.

**Versions 1.5.4 and later:** The install step required for previous versions was removed in [Coherent UI](#) 1.5.4. The new structure places the UI Resources inside the `WebPlayerTemplates` folder since it's the only one that Unity ignores when compiling scripts. The files placed there are not actual web player templates but a collection of HTML/CSS/JS files that are used by the samples. These files won't interfere with actual Web Player Templates as long as there is no file named `index.html`. The default UI Resources path is no longer set (the installation step in previous versions did that) and the default file handler will try to find the assets in the `WebPlayerTemplates` folder when ran in the **Editor**. For built games you **must** set the UI Resources path using the *Edit* → *Project Settings* → [Coherent UI](#) → *Select UI Folder* entry.

## 2.1 Prerequisites

- Unity3D 3.5.7 or later is required
- Only Pro version of Unity is supported
- [Coherent UI Mobile](#) supports iOS 5.1 and above.
- [Coherent UI Mobile](#) supports Android 3.0 and above (API level 11).
- Mobile versions of [Coherent UI](#) SDK for Unity3D work with the free version, but the *Preview* feature will not work.
- Java Development Kit (JDK) is required for exporting to Android. Referring to the folder you installed JDK in as `<JDK>`, you need to have the `<JDK>/bin` folder in your PATH environment variable and set the JAVA\_HOME environment variable to `<JDK>`.

### Note

[Coherent UI](#) SDK has different requirements than the [requirements for Unity3D](#). Unity supports older hardware via fallbacks, which may prevent [Coherent UI](#) SDK from running on older graphics cards. Please make sure that the requirements of [Coherent UI](#) SDK match your client requirements for the project.

## 2.2 Package contents and structure

The package has the following structure:

- *CoherentUI* - contains the UI debugger, samples, and documentation.
- *CoherentUI/Editor* - contains editor classes for displaying the properties of *Coherent UI* components plus a post-build step class. These classes provide utility functionality.
- *Plugins* - contains the *Coherent UI* libraries. These are automatically copied when building.
- *Standard Assets/Scripts/CoherentUI* - contains the *Coherent UI* integration classes. You'll find a Detail folder inside which contains classes that are internal for the implementation and are used by the "public" classes. The interface you should be using is outside the Detail folder.
- *StreamingAssets* - contains assets that need to be copied as-is in the build directory. These include the *Coherent UI* host process executable, the libraries it needs, locales and UI resources.

## 2.3 Usage of the package

After importing the *CoherentUI.unitypackage* in your project, the two main scripts you'll be using are *Standard Assets/Scripts/CoherentUI/CoherentUISystem* and *Standard Assets/Scripts/CoherentUI/CoherentUIView*. The easiest way to use *Coherent UI* is to drag the *CoherentUIView* component onto an object and hitting Play - that's it! Everything will be up and running. Actually, you can do the same for most of the usage scenarios - just drag the component and then configure it in the inspector. Here's a bit more detail about the two scripts.

The first script, *CoherentUISystem*, defines initialization parameters of the *Coherent UI* System and should be placed no more than once in your project. The UI system is meant to be initialized in the first scene and live throughout the game's lifetime. You need to add this component to your scene only if you need custom initialization of the *Coherent UI* System. For the most cases, using only *CoherentUIView*s is enough - they will automatically create an instance of the *Coherent UI* System for you with reasonable default parameters. Check *CoherentUISystem Lifetime* for details.

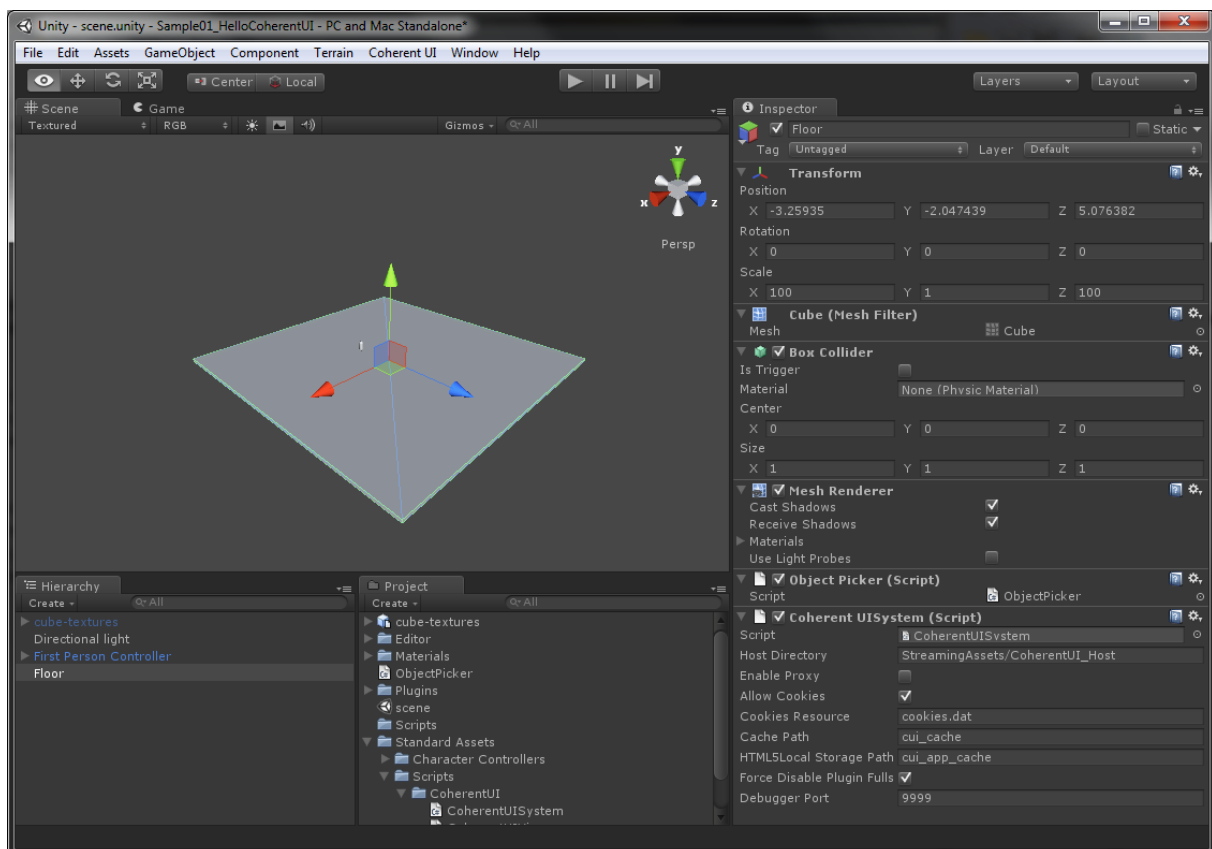


Figure 2.1: CoherentUISystem properties

The second script, `CoherentUIView`, will represent a single HTML page. This is the component that renders your CSS and JavaScript animations and makes your game alive. This component can be placed on any object that is renderable and serves as its material. When placing it on an object all the needed components are automatically created, hidden from you, and the rendered output is bound to the mainTexture of a new material that is created at runtime. This material is set as the gameObject's renderer material so that you see the page rendered on your object.

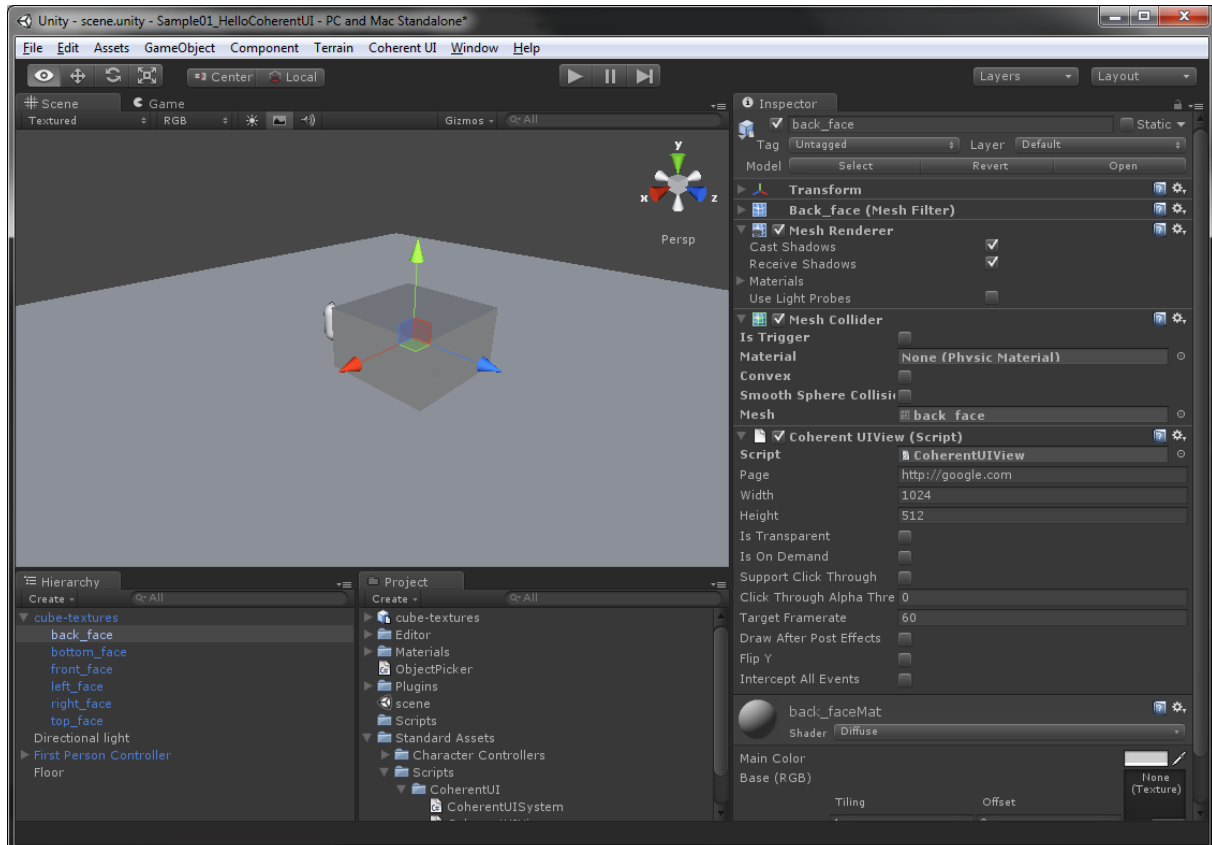


Figure 2.2: CoherentUIView properties

You can also place views on cameras. When doing so, you get your page rendered on the whole viewport of the camera. This way you can easily add a heads-up display for your game. Don't forget to mark your HUD as transparent and make your HTML page transparent! There's even an option for drawing the HUD after the post-effects if you like (The option is available for non-HUDs, too, but it results in a no-op because it doesn't make sense).

## 2.4 Switching between build targets

*Coherent UI Mobile* has different properties and a restricted subset of features compared to the desktop versions. To switch between the desktop features and the mobile ones you must switch the current active target in Unity. This is done by clicking on **File->Build Settings** selecting the target platform and clicking "Switch Platform". After doing so the properties of the Views will change to reflect their capabilities specific to the selected platform. Even if the current target is iOS or Android and hence you'll build with *Coherent UI Mobile* you can still preview your game in the Editor. The preview will use the desktop version of *Coherent UI*. Keep in mind that in this case you might see some differences compared to the the mobile device after the build as the desktop version has more features. You should check often how your UI behaves and looks on the device itself (or the simulator).

## 2.5 Trial version registration

The trial version of [Coherent UI](#) for Unity3D requires a simple registration. The registration requires:

- the email you used for downloading [Coherent UI](#)
- internet connection
- outgoing connections to port 3000 to be allowed in your firewall

The registration is started automatically when you run the game. It shows a simple form that asks for your email.

You can register [Coherent UI](#) on multiple machines, provided that they run different operating systems.

Games exported with the trial version of [Coherent UI](#) also require registration. This means if you want to run your built game on a different computer, you'll have to register [Coherent UI](#) on that computer, too. When building a game using the trial version of [Coherent UI](#) the activator is packaged as well and ran when the game is started. For manual activation, copy the *CoherentUI/Activator* folder from your Unity3D project to the machine and run:

```
Activator.exe --unity3d --host \  
    "<FULL PATH TO GAME DATA FOLDER>/StreamingAssets/CoherentUI_Host/windows"
```

on Windows and:

```
Activator.app/Contents/MacOS/Activator --unity3d --host \  
    "<FULL PATH TO GAME APP>/Contents/Data/StreamingAssets/CoherentUI_Host/macosx"
```

on Mac OS X.

## Chapter 3

# Samples

### Note

Some samples show Desktop-specific features while others focus on Mobile targets

The samples provide a starting point for you. They are located in `Assets/CoherentUI/Samples` and the required scripts are already configured.

The samples are based on a simple scene we've set up for you. The scene consists of a light, a floor, a cube, and a FPS controller so you can move around. There are a few key points in the base sample you should be aware of.

- The FPS controller's `MouseLook` script has been modified a bit - you can toggle if it's active using the **Alt** key. This was done for convenience so you can navigate over the page, projected on the cube easily, without looking around. The `CharacterMotor` script has also been modified for the same reason.
- The cube's faces all have `MeshCollider` components. They are needed by Unity to produce texture coordinates in the raycast hit info. The texture coordinates, in turn, are needed to calculate the position of the hit point on the [Coherent UI View](#).
- The scenes that have interactive [Coherent UI](#) surfaces placed on objects use the *Click-to-focus* feature of the [Coherent UI View](#). It allows the user to forward input to the view when she clicks on it and stop forwarding it when she clicks outside.

### Note

In the case where the mesh collider does not coincide with the actual geometry of the object, the raycast and coordinate calculation must take place in the user code. In the samples we're only using simple geometry that is the same as the mesh collider so no further work is needed.

## 3.1 Hello, Coherent UI

Just hit play - and you'll see a rotating cube and a sign, saying "Hello, Coherent UI". The sample demonstrates the easiest possible setup for [Coherent UI](#) for Unity3D. The scene is made of a cube in front of the camera with a script that rotates it, and a [Coherent UI](#) component attached to the main camera. The component's URL is set to a local resource using our special `coui://` protocol and the "Transparent" checkbox is ticked. You can preview the page displayed in your browser to see what you can expect.

### Note

The **coui** protocol uses the folder selected by the *Edit* → *Project Settings* → [Coherent UI](#) → *Select UI folder* menu item as a root for a virtual file system.

You can also add a *CoherentUIView* component to a 3D object if you like. To do so, locate the *CoherentUIView* script under **Standard Assets/Scripts/Coherent UI/CoherentUIView** in the *Project* window and then add it to the object of your choice. In Unity3D 4.x you can do that from the inspector windows by clicking "Add component".

Here's a sample where a *Coherent UI View* component is added to a cube. Make sure you tick the "Click to focus" checkbox for the view to get input forwarding easily.

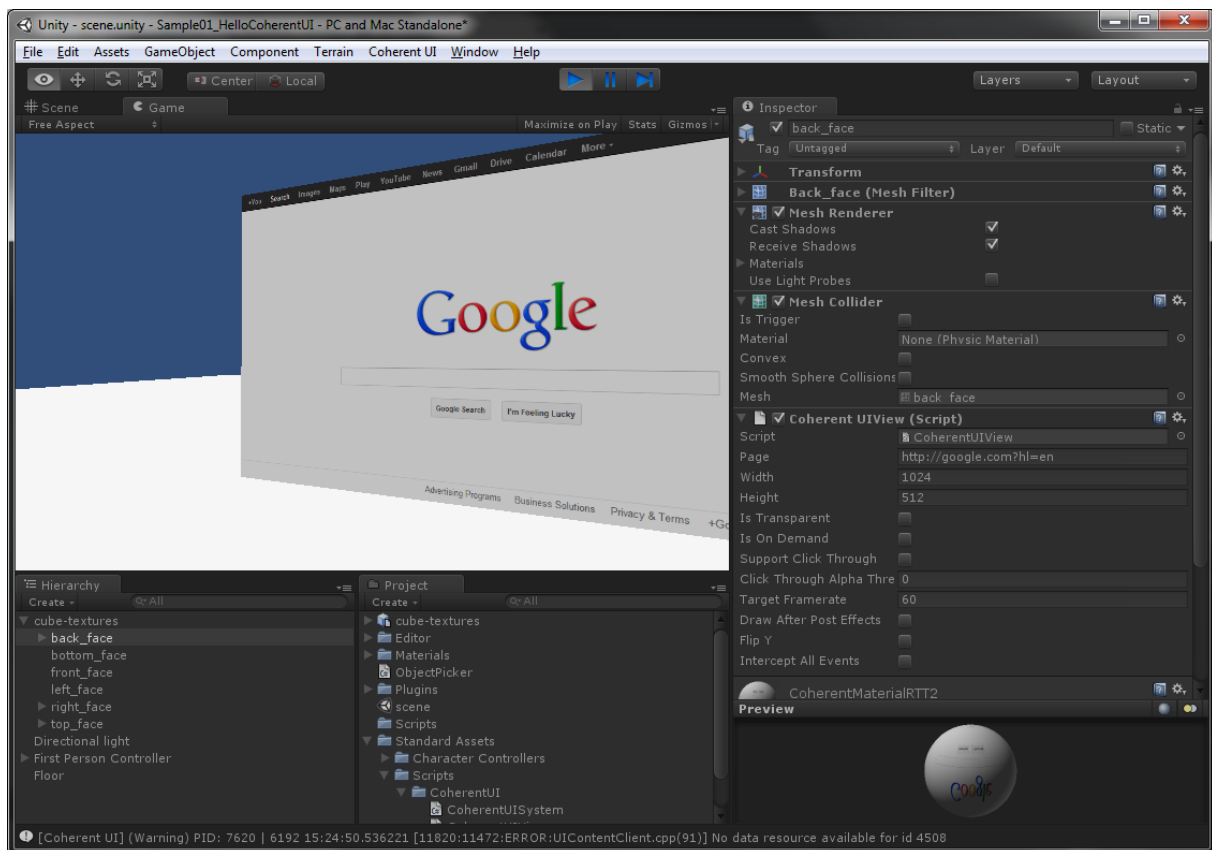


Figure 3.1: CoherentUIView on a texture

You can also try pages that show dialog boxes, such as

- [http://www.w3schools.com/js/tryit.asp?filename=tryjs\\_alert](http://www.w3schools.com/js/tryit.asp?filename=tryjs_alert)
- [http://www.w3schools.com/js/tryit.asp?filename=tryjs\\_confirm](http://www.w3schools.com/js/tryit.asp?filename=tryjs_confirm)
- [http://www.w3schools.com/js/tryit.asp?filename=tryjs\\_prompt](http://www.w3schools.com/js/tryit.asp?filename=tryjs_prompt)
- <http://www.httpwatch.com/httpgallery/authentication/> (scroll down and click Display Image)

If you do not handle the Listener events for dialog boxes by yourself they will be handled automatically by *Coherent UI*.



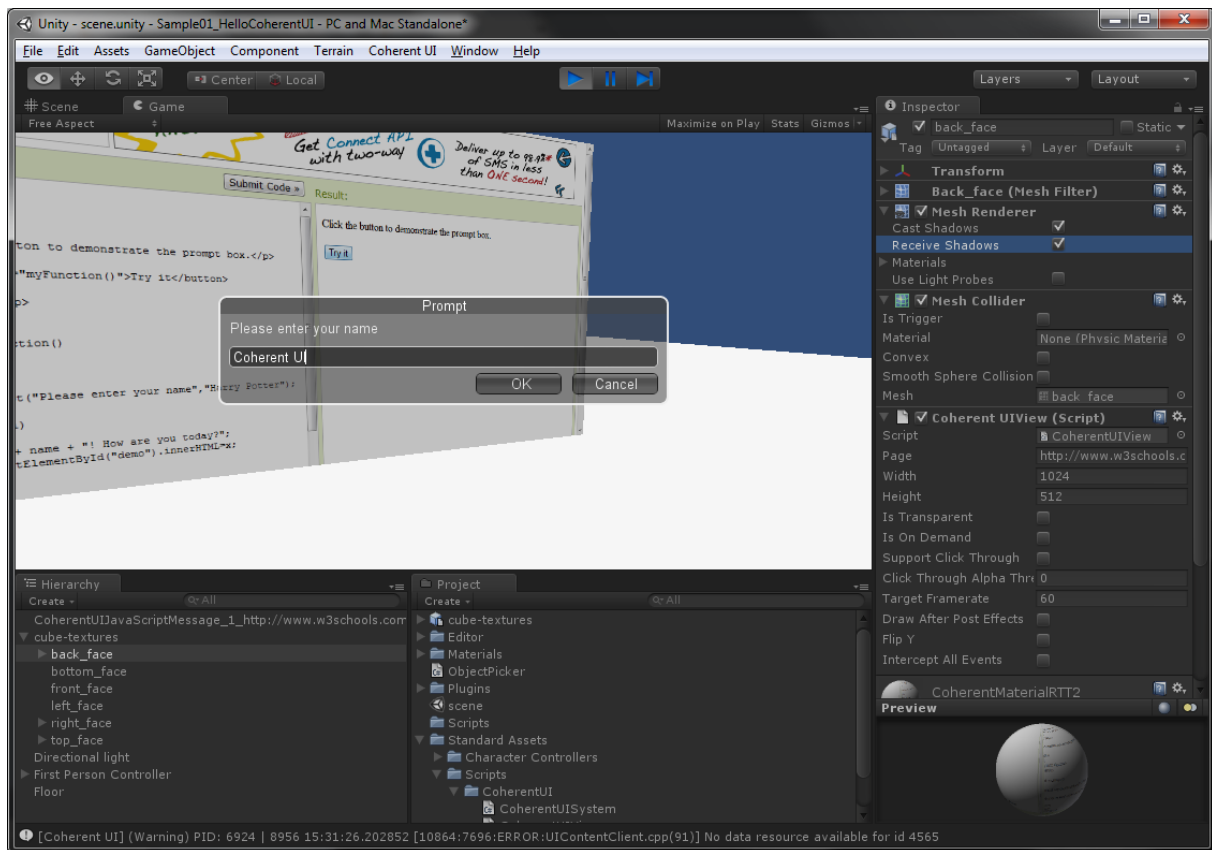


Figure 3.2: CoherentUIView showing a prompt

## 3.2 Facebook integration

This sample uses a facebook application that shows some of your photos in a rotating circle. You'll see how to customize the `View` and `ViewListener` behaviour to suit your needs.

### Note

If you're having trouble displaying the local resources, make sure you have selected the `UIResources` folder located in the root of the project using the `Edit` → `Project Settings` → `Coherent UI` → `Select UI folder` menu in the editor. This folder will be used for resources addressed with `coui://` links.

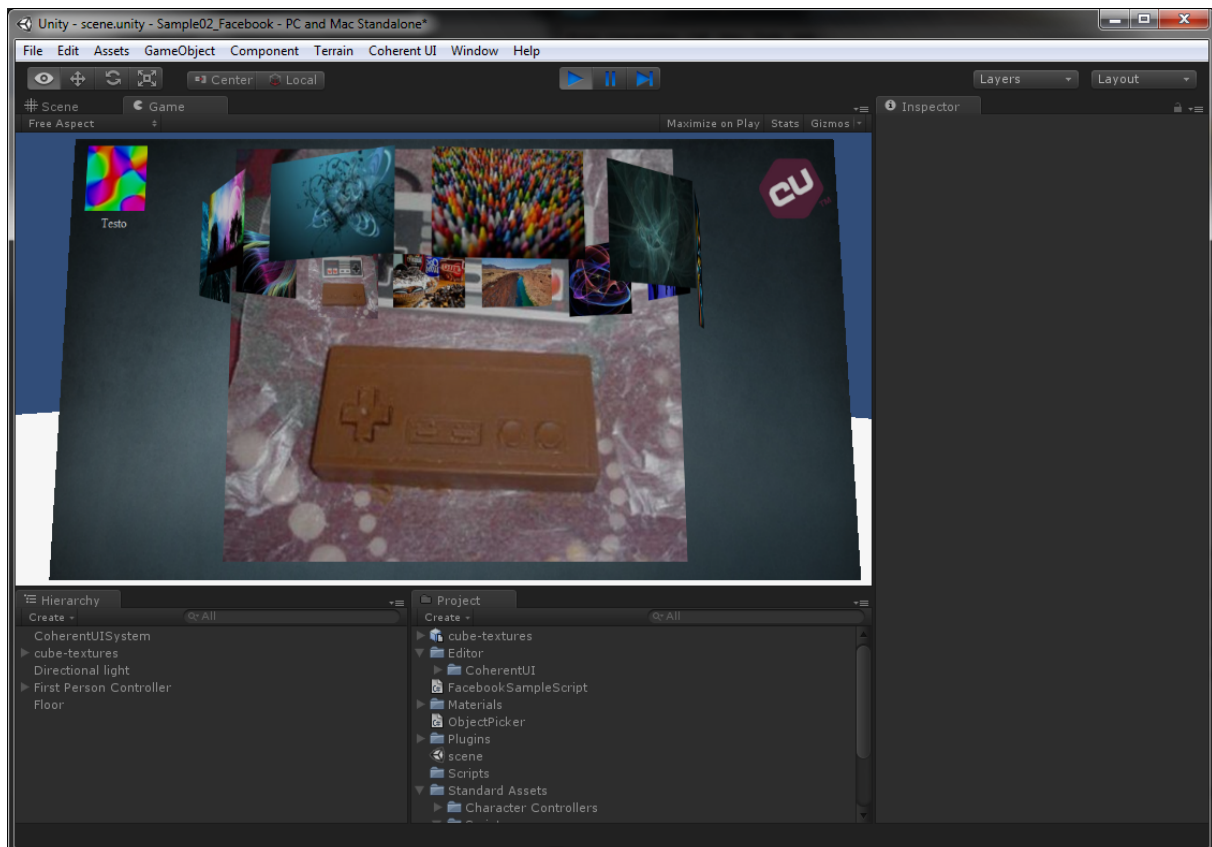


Figure 3.3: CoherentUIView Facebook integration sample

You will find a new script - `FacebookSampleScript`. We need to modify the default behaviour of `UnityViewListener` for this sample. The key points are:

- In its `Start` method the `FacebookSampleScript` obtains the `CoherentUIView` component in the game object.
- In the `Start` method the script also attaches to the `CoherentUIView.OnViewCreated` event to enable intercepting URL requests and to the `UnityViewListener.URLRequest` event to redirect the Facebook login URL to the local HTML page.

This is the easiest way to handle events provided by the `UnityViewListener`. Another option is to derive the `UnityViewListener` and `CoherentUIView` but that is unnecessary.

The `FacebookSampleScript.OnURLRequestHandler` checks if the requested URL is the one that was registered in the *Facebook Application settings* (this might be any made up URL). If that's the case, we redirect it to the local page that uses our Facebook application. You will find the page we're redirecting to in the sample project folder under *FacebookSample/facebook.html*.

Now that we've explained what the class does, the only thing left is to hit play. You can press **Alt** to stop moving/looking so you can browse in the Facebook application.

### 3.3 Menu And HUD

This sample shows a game menu and HUD built with *Coherent UI*.

**Note**

If you're having trouble displaying the local resources, make sure you have selected the *UIResources* folder located in the root of the project using the *Edit* → *Project Settings* → *Coherent UI* → *Select UI folder* menu in the editor. This folder will be used for resources addressed with `coui://` links.

Double-click the *Menu* scene in the Project window (located in *Scenes*) to make it active, if it isn't already. In the *Menu* scene we have a *MenuScript* component attached to the camera. *MenuScript* does three things:

- sends the mouse position to the view every `Update()`
- registers the handlers for clicking on a menu button
- loads the game when the "New Game" button is clicked

**Note**

You have to add the *game* scene to the build settings of your project so Unity3D can load it.

In the game, we have another component for controlling the view - HUD, this time implemented in *UnityScript*. Each frame the component updates the compass orientation, based on its transformation.

In addition to updating the compass, the HUD component takes care of disabling the *CharacterMotor* component when the focus is on a *Click-to-focus* view. The HUD script attaches to *CoherentUISystem::OnViewFocused* event and changes the *canControl* property of the selected *CharacterMotor* whenever a *Click-to-focus* view gains or loses focus. This allows typing in the focused view without moving the character. Here is the HUD component in code:

```
#pragma strict
#if UNITY_STANDALONE || UNITY_STANDALONE_WIN || UNITY_STANDALONE_OSX || UNITY_EDITOR
import Coherent.UI.Binding; // to use View.TriggerEvent with extra arguments

private var View : Coherent.UI.View;
private var CurrentDirection : float;

// CharacterMotor component to be disabled when a Click-To-Focus view has gained focus
public var characterMotor : CharacterMotor;

function Start () {
    var viewComponent = GetComponent(typeof(CoherentUIView)) as CoherentUIView;

    viewComponent.OnViewCreated += this.ViewCreated;
    viewComponent.OnViewDestroyed += this.ViewDestroyed;

    CurrentDirection = 0;

    var uiSystem = Component.FindObjectOfType(typeof(CoherentUISystem)) as CoherentUISystem;
    // get notified when a Click-to-focus view gains or loses focus
    uiSystem.OnViewFocused += this.ViewFocused;
}

function ViewCreated(view : Coherent.UI.View) {
    View = view;
    var viewComponent = GetComponent(typeof(CoherentUIView)) as CoherentUIView;
    Debug.LogWarning(String.Format("View {0} created", viewComponent.Page));
}

function ViewDestroyed() {
    View = null;
}

function ViewFocused(focused : boolean) {
    if (characterMotor) {
        // enable or disable the character movements
        characterMotor.canControl = !focused;
    }
}

function Update () {
    if (View != null)
    {
        var direction = this.transform.rotation.eulerAngles.y;
        if (Mathf.Abs(direction - CurrentDirection) > 2)
        {

```

```

        View.TriggerEvent("SetAbsoluteCompassRotation", direction);
        CurrentDirection = direction;
    }
}
#endif

```

In the game scene we have one more component - `ObjectPicker`, that takes care of directing the input to the correct `CoherentUIView`. First it checks whether the mouse is on a HUD element and if it is not, then raycasts and checks whether the mouse is over the browser window. Make sure the "Support Click Through" checkbox is ticked on the HUD `CoherentUIView` - this allows you to detect whether the mouse is over a transparent area in the View. If the property is left unchecked the input will never reach any objects in the world, because the system will think that the mouse is always over the HUD.

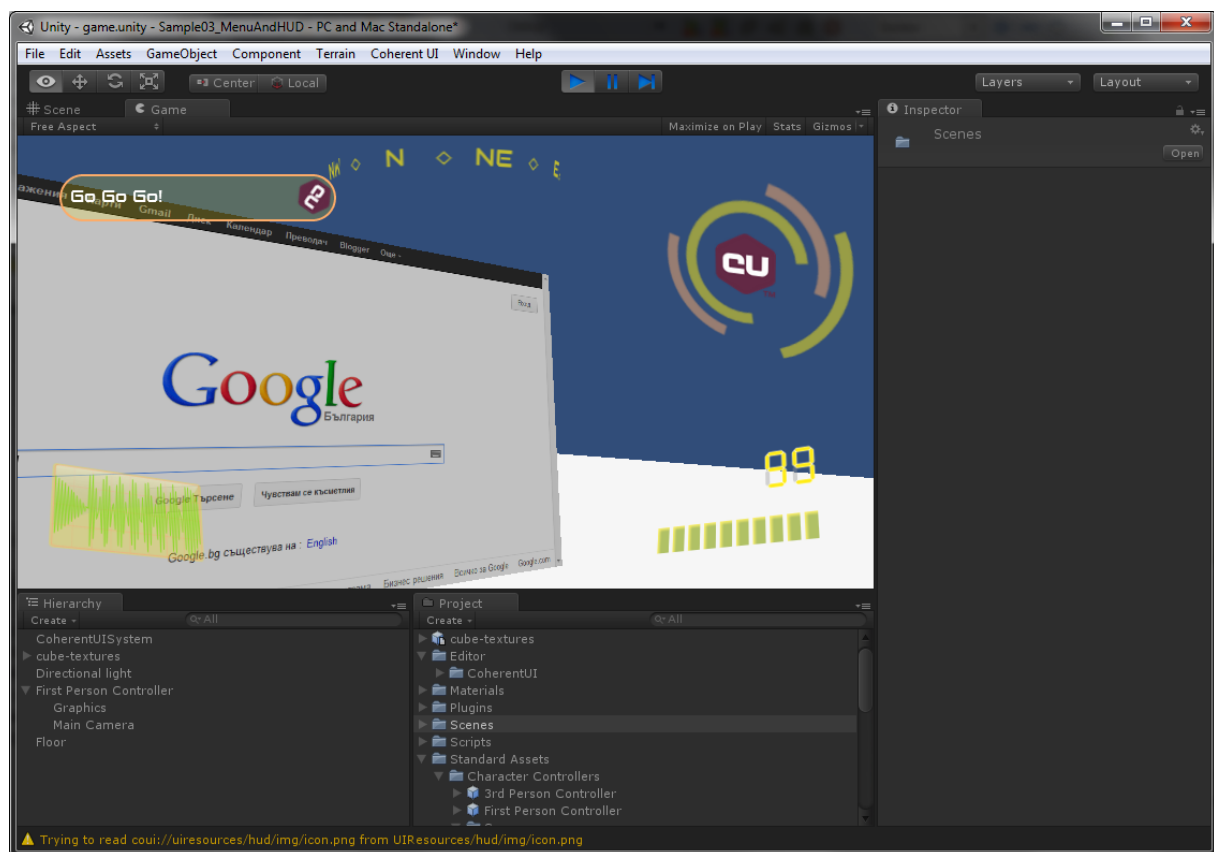


Figure 3.4: CoherentUIView HUD sample in action

### 3.4 Binding Demo

This sample will demonstrate communication between Unity3D and JavaScript used by the HTML page loaded in a [Coherent UI View](#).

#### Note

If you're having trouble displaying the local resources, make sure you have selected the `UIResources` folder located in the root of the project using the `Edit → Project Settings → Coherent UI → Select UI folder` menu in the editor. This folder will be used for resources addressed with `coui://` links.

The sample shows a simple options dialog. The options are passed from JavaScript to Unity3D and back to exercise the binding. This walkthrough will give you a brief overview of the [Coherent UI Binding](#) fundamentals but be sure

to check out the [Binding](#) section in this guide, as well as the **Binding for .NET** section in the general reference document.

#### Note

This sample guide is written for C# Unity3D scripts

First, when passing custom object types between Unity3D and JavaScript, you must first inform [Coherent UI](#) about the data in the type. You can do that using the `CoherentType` attribute:

```
using Coherent.UI.Binding;

// all properties / fields for Options will be visible to Coherent UI
[CoherentType(PropertyBindingFlags.All)]
public struct GameOptions
{
    public string Backend;
    public uint Width;
    public uint Height;

    public string Username
    {
        get {
            return System.Security.Principal.WindowsIdentity.GetCurrent().Name.ToString();
        }
    }

    // rename the NetPort property to NetworkPort
    [CoherentProperty("NetworkPort")]
    public uint NetPort { get; set; }
}
```

Now the `GameOptions` structure will correspond to a JavaScript object having the same properties. Note that you can rename a property using the `CoherentProperty` attribute. In this case, the "NetPort" property will correspond to "NetworkPork" in JavaScript.

After exposing the properties, we need to register some event handlers. This can be done in two ways. The first way to bind event handlers is to add a handler for the `ReadyForBindings` event of the `UnityViewListener` (manual binding). The second way (.NET only) is to add a `CoherentMethod` attribute to the method you want to bind (automatic binding). See the [CoherentMethod](#) section in this guide for more details.

#### Note

The sample provides two scripts for binding - `ManualBinding.cs` and `AutomaticBinding.cs`. Make sure that only one of those scripts is active when exploring the sample.

We'll explore the manual binding first. Start by registering a handler for `ReadyForBindings`:

```
m_View = GetComponent<CoherentUIView>();
m_View.Listener.ReadyForBindings += HandleReadyForBindings;
```

The handler would look like this:

```
void HandleReadyForBindings (int frameId, string path, bool isMainFrame)
{
    if (isMainFrame)
    {
        // bind ApplyOptions to "ApplyOptions" in JavaScript
        m_View.View.BindCall("ApplyOptions", (Action<GameOptions>)this.ApplyOptions);
        m_View.View.BindCall("GetLatency", (Func<int>)this.GetNetworkLatency);
        m_View.View.BindCall("GetGameTime", (Func<int>)this.GetGameTime);

        m_View.View.BindCall("GetMath", (Func<BoundObject>)(() => {
            return BoundObject.BindMethods(new MyMath());
        }));

        // triggered by the view when it has loaded
        m_View.View.RegisterForEvent("ViewReady", (Action)this.ViewReady);
    }
}
```

Now, when JavaScript calls `engine.call("ApplyOptions", options)`, Unity3D will execute its handler - namely the `(Action<GameOptions>)this.ApplyOptions` method registered above.

The `options` structure passed as a parameter looks like this:

```
function onApplyButton() {
    var options = {};
    options.__Type = "GameOptions";
    options.Backend = $("#backend").text();
    options.Width = Number($("#gameWidth").val());
    options.Height = Number($("#gameHeight").val());
    options.Username = $("#user").text();
    options.NetworkPort = Number($("#netPort").val());

    // This will call the C++ engine code with the just created structure. It'll be correctly populated
    engine.call("ApplyOptions", options);
}
```

Note that there is one "internal" property - `__Type`. This property defines the mapped type and is essential for correct behavior of the binding provided by [Coherent UI](#).

The "ApplyOptions" handler just bounces the options back to JavaScript:

```
public void ApplyOptions(GameOptions options)
{
    m_View.View.TriggerEvent("gameConsole:Trace", options);
}
```

`gameConsole:Trace` will dump the `options` object in the console.

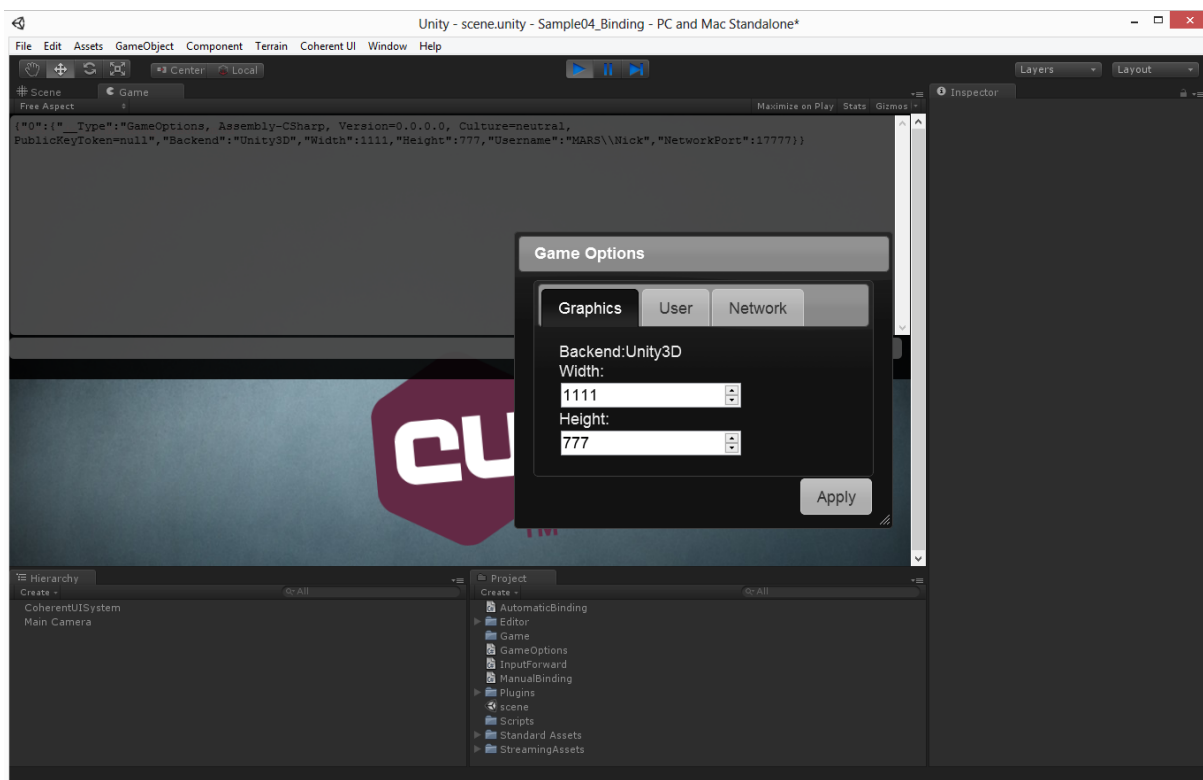


Figure 3.5: Binding sample in action

The automatic binding saves us the first two steps - instead of registering a handler in the `UnityViewListener` callback you can decorate a method with the `CoherentMethod` attribute:

```
[Coherent.UI.CoherentMethod("ApplyOptions", false)]
public void ApplyOptions(GameOptions options)
{
    m_View.View.TriggerEvent("gameConsole:Trace", options);
}
```

The first argument of the attribute is the JavaScript function that we're binding. The second defines whether the function is an *event* (opposed to a *call* - *calls* are single-cast only).

## Returning a value to JavaScript

*Calls* can also return a value. Let's inspect the "GetLatency" binding:

```
// By default, the second argument of CoherentMethod is false
[Coherent.UI.CoherentMethod("GetLatency")]
public int GetNetworkLatency()
{
    // not actual latency :)
    return (int)UnityEngine.Random.Range(0, 1000);
}
```

Since the execution of JavaScript is **not** synchronous you will receive the result in a callback. [Coherent UI](#) provides a promise/future pattern for convenience. This is the "GetLatency" JavaScript call in all its glory:

```
function getLatency() {
    engine.call("GetLatency").then(function() {
        $("#latency").text(arguments[0]);
    });
}
```

As you can see from the script, the value returned by Unity3D will be passed to the callback as an argument.

## Method Binding

This sample also shows how to use binding of .Net methods. For detailed explanation see "Exposing Methods" in the "Binding for .Net" chapter in the API documentation.

The sample exposes an instance of `MyMath` to JavaScript.

```
class MyMath
{
    public double Sum(double[] numbers)
    {
        return numbers.Sum();
    }

    public double Average(double[] numbers)
    {
        return numbers.Average();
    }
}
```

When the game is up and running, we request the `MyMath` instance, store it as a global object and register two console commands that use it.

```
engine.on('Ready', function () {
    //get the exposed MyMath object
    engine.call('GetMath').then(function (math) {

        // store reference to the object
        window.MyMath = math;

        // register commands for the console
        engine.trigger('gameConsole:AddCommand', 'sum', 'computes the sum of its arguments', function (line) {
            var numbers = line.split(/\s+/).map(Number);
            numbers.splice(0, 1);
            window.MyMath.Sum(numbers).then(function (sum) {
                engine.trigger('gameConsole:Trace', 'The sum of', numbers, 'is', sum);
            });
        });

        engine.trigger('gameConsole:AddCommand', 'avg', 'computes the average of its arguments', function (line) {
            var numbers = line.split(/\s+/).map(Number);
            numbers.splice(0, 1);
            window.MyMath.Average(numbers).then(function (avg) {
```

```

        engine.trigger('gameConsole:Trace', 'The average of', numbers, 'is', avg);
    });
    });
});
});

```

To try out the commands press `~` to open the game console and type `sum 40 2`.

## Events

When you need multiple handlers for a JavaScript function you need to register an *event*. This can be done by the `Coherent.UI.View.RegisterForEvent` method for manual binding

```
m_View.View.RegisterForEvent("ViewReady", (Action)this.ViewReady);
```

and passing `true` as a second parameter of the `CoherentMethod` attribute for automatic binding.

```

[Coherent.UI.CoherentMethod("ViewReady", true)]
public void ViewReady()
{
    // show the options
    m_View.View.TriggerEvent("OpenOptions", m_GameOptions);
}

```

Bear in mind that *events* **cannot** return a value.

If you want to handle an event in JavaScript, you need to register a handler using the `engine.on` call.

```

engine.on('OpenOptions', function (options) {
    // Open an options dialog
});

```

An event can be triggered by JavaScript by using

```
engine.trigger("EventName", eventArgs);
```

or by Unity3D using

```
m_View.View.TriggerEvent("EventName", eventArgs);
```

where `m_View` is `CoherentUIView`.

This concludes the binding demo walkthrough.

## 3.5 Archive resource

This sample demonstrates reading resources from an archive through a custom file handler.

The scene is just a simple camera that has an attached `CoherentUIView` component and another one that forwards input to the view. The important component for this sample is *CustomFileHandlerScript*. It sets the `CoherentUISystem` factory function object in its `Awake` method so the system is initialized with the custom handler.

```

void Awake()
{
    CoherentUISystem.FileHandlerFactoryFunc = () => { return new CustomFileHandler("UIResources", "
        ArchiveResource.tar"); };
}

```

In fact, that's all the component has to do - set the `CoherentUISystem.FileHandlerFactoryFunc` prior to any `Start` method being called. The returned `CustomFileHandler` is the class that does the actual reading/writing. The methods the user needs to implement are:



```
public override void ReadFile (string url, Coherent.UI.ResourceResponse response)
{
    ...
}

public override void WriteFile (string url, Coherent.UI.ResourceData resource)
{
    ...
}
```

They will be called for all *coui://* links and for the cookies (if enabled).

#### Note

When reading/writing cookies, you may receive a URL that has a *file://* scheme.

The input URL can be interpreted in any way you see fit for your handler. In the example we try to open the link *coui://UIResources/mainmenu/menu.html*. The *mainmenu/menu.html* is actually compressed in an archive. First we verify that the host part of the URL (*UIResources*) is what we expect for compressed resources and then proceed to search the archive for the resource.

Since **Coherent UI** supports asynchronous reading/writing of resources, when done with the I/O operations, you must use the *ResourceResponse* or *ResourceData* object to signal the outcome. You can do that by using the *SignalSuccess* and *SignalFailure* methods.

#### Note

At the moment the data you read/write must be converted to a native buffer before usage. This is done with the *System.Runtime.InteropServices.Marshal* class. Here's a short snippet that reads a file and then copies the managed buffer in an unmanaged buffer provided by the *ResourceResponse* object:

```
public override void ReadFile (string url, Coherent.UI.ResourceResponse response)
{
    byte[] bytes = File.ReadAllBytes(url);

    IntPtr buffer = response.GetBuffer((uint)bytes.Length);
    if (buffer == IntPtr.Zero)
    {
        response.SignalFailure();
        return;
    }

    Marshal.Copy(bytes, 0, buffer, bytes.Length);

    response.SignalSuccess();
}
```

Make sure you check out the code in *CustomFileHandlerScript* for a complete example.

## 3.6 Mobile Input

#### Warning

Unity3D 3.5 doesn't support native plugins for iOS simulator builds. There are a couple of articles explaining how to workaround this limitation, but the most helpful is <http://tech.enekochan.com/2012/05/28/using-the-xcode-simulator-with-a-unity-3-native-ios-plugin-in/>. We have added a script with a post-process step that tries to fix the project, so it can be used with the iOS simulator. The script is in *Editor/iOSSimulatorProjectFixer.cs*.

This sample demonstrates how you can control what part of the input is forwarded to Unity3D by using JavaScript. The sample starts with an overlay that has 3 buttons on the left side, area for testing input forwarding on the right, and a red box in the world. The buttons enable/disable the input forwarding for touches on the right side of the overlay and the third one bumps the box upwards. When you touch the box a little force is applied and it should move forward as if you pushed it. If you touch the box in the right area of the overlay the force is applied only if input forwarding is enabled.

There are also buttons for toggling the overlay and mouse look controller in the top left corner for convenience.

**Warning**

The HTML code for the sample is not designed for small displays so it may look out of proportion on a phone or a phone simulator. You can try lowering the font size in the accompanying `css` file.

**Note**

The [Coherent](#) UI View used has its input state set to "Transparent". If you set it to "Take all" input is consumed before reaching Unity3D and if you set it to "Take none" it is sent directly to Unity3D.

**Warning**

Unfortunately, at the moment you can't see the actual behavior of the sample in the Unity3D Editor. Input is handled differently for the standalone and mobile versions so it is recommended that you test your project either on a mobile device or an emulator.

Code-wise speaking, when you make a touch on a "Transparent" [Coherent](#) UI View, the `engine.checkClickThrough` function is called. You can check its code in `coherent.js`. Basically it obtains the DOM element from the touch coordinates and begins a series of checks. First, if the element is the *document body* or has the **`coui-noinput`** class, input is directly sent to the client (Unity3D). Otherwise, the element is checked for having a **`coui-inputcallback`** class. If it does, the element's `couiInputCallback` is invoked which determines whether the input is forwarded or consumed; if it doesn't have such class, input is consumed.

The sample enables/disables the input forwarding on the right area by removing/adding the **`coui-noinput`** class to the corresponding DOM element. In this sample, only the DOM element for the right area has the class **`coui-inputcallback`** and its `couiInputCallback` function is set when the engine has signaled it's ready. The function itself doesn't do anything special, it just always returns `true`, meaning that the input is consumed in JavaScript and never sent to the client (Unity3D).

You can also bump the box upwards which is a demonstration for the binding for [Coherent](#) UI Mobile. It's very much the same as binding for .NET/Unity3D standalone so we'll not go in detail here.

Summing up:

- if you want the input forwarded to the client when touching an element, simply add the **`coui-noinput`** class to the element.
- if you want an element to consume input, ensure that it does **not** have the **`coui-noinput`**.
- if you want to have custom processing over an element, ensure that it does **not** have the **`coui-noinput`**, add a **`coui-inputcallback`**, and add a function `couiInputCallback` which ultimately returns `true` if you want to consume the input or `false` if you want to forward it to the client.

Note that for obtaining the element below the touch point we're currently using `document.elementFromPoint`. In the sample, the right area is represented by a `<div>` and there's some text inside it as a child element. Only the `<div>` has the `coui-inputcallback` class. If you touch the text its element will be checked for the `coui-inputcallback` class and since it doesn't have one input will be consumed. Since we want to apply the logic for all touches inside the area this presents a problem. One solution is to add the `coui-inputcallback` class to all child elements and set their `couiInputCallback` functions to the same variable. Another solution is to use the **`pointer-events`** CSS property on the children elements, e.g.:

```
#menu-right > *
{
  pointer-events: none;
}
```

This is how the sample solves the problem. Note that this is an experimental CSS property and prevents elements from being the target of pointer events. This is fine in the sample but may have adverse effects in your code so use it with caution.

## 3.7 Oculus

### Note

It is not possible to create a sample scene for Oculus Rift for Unity3D 3.5 using the official Oculus SDK. So our Oculus Rift sample scene is created with Unity3D 4 and can not be packaged in our asset store packages. You can download a package with the scene from: [https://s3.amazonaws.com/CoherentLabs/-Public/Coherent\\_UI\\_Oculus\\_Sample.unitypackage](https://s3.amazonaws.com/CoherentLabs/-Public/Coherent_UI_Oculus_Sample.unitypackage)

This sample demonstrates how you can create UI elements in your games for the Oculus Rift, using the CoherentUI and OculusVR integrations in Unity. The focus of the sample is on creating a HUD element that can be used with the Oculus headset.

### Note

You should include the Oculus integration package for Unity in order to use Oculus in Unity3D (you need a Unity Pro 4.01 or higher version). For more information about the Oculus SDK and its integration with Unity, please visit <http://www.oculusvr.com/>.

You can get Oculus ready for use in Unity by following these steps:

- go to the official website of Oculus Rift - <http://www.oculusvr.com/> ;
- download OculusUnityIntegration.unitypackage from there;
- open an Unity3D instance and go to Assets -> Import Package -> Custom Package. When the dialog box for choosing packages opens, go to the place with the downloaded package and select it. A new dialog box with a list of the files for import will open. Choose 'Import All'. After completing this step you should be able to see a new menu - 'Oculus'.

Now we are ready to proceed with the sample. As you can see, there is an OVRPlayerController in the *Scene* menu (it can be added from Oculus -> Prefabs -> OVRPlayerController). The controller is comprised of two things: a simple character controller and another OVR prefab - OVRCameraController. The latter is used as an interface between Unity and the two cameras. Double-click it to reveal its children objects. Double-clicking the OVRCameraController will show the cameras and a plane object attached to the right camera. This plane is used as a surface for drawing the HUD. There is an CoherentUIView component attached to it, which uses `coi://UIResources/Menu-AndHUD/hud/hud.html` as a page. It is suggested that for drawing HUD elements you attach the surface to the right camera for optimal results.

### Note

To render a view on top of everything, we put an check mark in 'IsIndependentOfZBuffer'.

### Warning

Whenever using a plane as a surface for views, choose a material for the plane and use a 'Transparent/Diffuse' shader for it.

This is basically all you need to draw the HUD. Hit play to see the result

## 3.8 Mobile Surface

The Mobile Surfaces sample shows how to use [Coherent](#) UI views on surfaces for mobile platforms. The sample shows a texture with a [Coherent](#) UI view on it that is rotating. Every 100 frames the view is changed and the texture is updated by calling the 'UpdateView' method of the view.

### Note

To achieve transparency of the surface *both* - the *IsTransparent* property of the view should be checked **and** the shader of the material should be set to *Transparent/Diffuse*.

### 3.9 Live Game Views

#### Note

Live Game Views are a Desktop-only feature.

The sample shows a simple scene with a Live Game View. The scene consists of a "game world" with a plane as ground, four rotating colored cubes and a light.

On the main Camera a [Coherent](#) UI View is attached that serves as an in-game HUD(heads-up-display). The Live Game View is part of the HUD. A [Coherent](#) UI Live Game View component is also attached to this camera and hence to the UI - it is the "link" between the View and the provider of the 3D images.

A second camera is fixed on the cubes and provides the source data. Everything this camera renders is automatically sent to the UI View linked to it as a dynamic "ImageData" object in the UI JavaScript Code. The HUD.html page draw the Live View with the following code:

```
window.onload = function() {
    var c = document.getElementById("myCanvas");
    c.onEngineImageDataUpdated = function (name, image) {
        var c = document.getElementById("myCanvas");
        var ctx = c.getContext("2d");

        ctx.clearRect(0, 0, c.width, c.height);
        ctx.putImageData(image, 0, 0);
        ctx.fillStyle = "white";
        ctx.font = "16pt Arial";
        ctx.fillText("Live Game View!", 15, 25);
    }
}
```

Every time a new frame is drawn by the "live" camera the "onEngineImageDataUpdated" is called on all "canvas" elements that define it in the page. There you can use the data in whatever fashion you need. In this sample we draw the image into the canvas and write some text on it. When you "Play" the game you'll see the four rotating cubes drawn in the HUD in the lower left corner as the "live" camera sees them.

Note that now the images of the camera are part of the UI DOM, so you can achieve all UI effects with it. The canvas elements can be animated, 3D transformed into the UI, post-effects can be applied on them, they can be interactive etc.

Via Live Game Views in a real game you can trivially have 3D animated unit portraits, animated items in the UI, 3D mini-maps, security cameras etc.

### 3.10 IME Sample

This sample demonstrates the usage of Input Method Editor for languages that require more complex input such as Chinese. To have IME enabled for a particular view you should just put a check on the *Enable IME* property of the **CoherentUIView** and this is all you need to do so as to have support for IME in your view.

Open the IME Sample, change your input language to one that requires IME, for example Chinese and hit play. Try to write something in the dialog that appears and hit 'Enter' or just press the 'Say' button. What you just wrote will appear in the dialog window.

Enabling IME does a few things:

- It subscribes for the **OnCaretRectChanged**, **IMEShouldCancelComposition** and **OnTextInputControlTypeChanged** events.
- **OnTextInputControlTypeChanged** event is fired when the type of text input control changes. The method that subscribes for it determines from the new type of control whether IME composition mode in Unity should be turned off or on.
- **OnCaretRectChanged** event is fired when the caret has changed its position. It has arguments *x* and *y* for the top left position of the caret rectangle and *width* and *height* for its width and height, all of them measured in pixels, relatively to the view. Since in Unity you have to set the position of the candidate list window, this method is used for helping set the candidate window position on the screen.

**Note**

If you want to set the candidate window on a specific screen position or make the calculations on your own, you can subscribe with a method for the **CalculateIMECandidateListPosition** handler, which has the same arguments as `OnCaretRectChanged` and should return the desired position of the candidate window in screen space. Please note that (0, 0) for the IME cursor position in Unity is in the upper left corner in spite of bottom left.



## Chapter 4

# Programmer's API

The Unity API is an extension of the normal .Net API, and it is contained in the main API Reference documentation.

### 4.1 CoherentUISystem properties for Desktop

- **Main Camera** - The main camera in the scene. This property is used only for [Click to focus Views](#).
- **Host Directory** - Path to the directory where the [Coherent](#) UI executable resides. This path can be relative to the resource path of the game (i.e. the *Assets* folder of the Unity3D project or the <YourGameName>\_Data folder of the built game). By default it is *StreamingAssets/CoherentUI*.
- **Proxy** - Enables proxy support by autodetecting the system settings. This detection is usually very slow and this setting should be enabled only when the user is behind a proxy and you're accessing the Internet.
- **Cookies** - Enables support for cookies.
- **Cookies file** - A file that will be used for reading and writing cookies when they are enabled.
- **Cache path** - Path for saving cached data. Leave null for in-memory caching only.
- **Local storage path** - Path for saving HTML5 page local storage data. Leave null to forbid local storage.
- **Disable fullscreen plugins** - Disables fullscreen mode for all plugins (e.g. Flash, Silverlight, etc.)
- **Disable web security** - Disable same origin policy. Use with caution.
- **Debugger port** - The port that will be opened for the debugger to connect and debug your interface. A value of -1 means disabled.
- **Support for Alt Tab** - Enables support for Alt and Tab in fullscreen DirectX9 applications for Windows. Due to restrictions in Dx9, it is necessary to reset Unity's graphics device in windowed mode before initializing UI System and after the system has been initialized and is ready, set reset it to fullscreen mode. If you uncheck this, the resets won't happen in fullscreen mode, but Alt+Tab functionality will not be supported.

The `CoherentUISystem` component also provides a static factory function object (`FileHandlerFactory-Func`) that the user can customize in order to make use of her own `FileHandler`. See the [Custom file handler](#) section for a detailed explanation.

### 4.2 CoherentUISystem properties for Mobile

- **URL cache** - Decides if the device URL cache will be used.
- **Memory cache size** - The maximum size of the in-memory cache.
- **Disk cache size** - The maximum size of the disk cache.

### 4.3 CoherentUIView properties for Desktop

- **URL** - Indicates the URL that will be initially loaded.
- **Width** - The width of the [Coherent](#) UI View.
- **Height** - The height of the [Coherent](#) UI View.
- **Transparent** - Defines if the View supports transparency.
- **Pre-load script** - The script will be executed before any other code in the UI View
- **Smart input** - *Available for transparent Views.* Enables support for queries whether the cursor is over a transparent pixel. A transparent pixel is considered one that has an alpha value below or equal to the *click-through alpha threshold*.
- **Smart input alpha** - A value in the range [0-1] inclusive that determines whether a pixel is transparent; A pixel is transparent if its alpha value is below or equal to the threshold.
- **Lockable focus** - When enabled, the View takes the input focus when clicked and releases it when you click outside the View. See [Click to focus Views](#)
- **Enable IME** - When enabled, this view will have Input Method Editor enabled for languages that have more complex input like Chinese.
- **On Demand ([Coherent UI Pro only](#))** - Provides perfect synchronization of the game frames and [Coherent](#) UI frames. Use this when you need to synchronize the game and the interface, e.g. when displaying name tags over the players. Using standard views may introduce a delay of a few frames.
- **Timer override** - The UI uses the in-game timer (requires On-demand)
- **Max. frame-rate ([Coherent UI Pro only](#))** - Sets the maximum framerate for the View. The view will never exceed this framerate.
- **Draw order** - Defines whether the View is drawn before or after the post-effects. Available for Views attached to cameras.
- **Flip Y** - Flips the drawn image vertically.
- **Auto UI messages** - When enabled, any event triggered in *JavaScript* on the `engine` object is forwarded to the game object containing the View.
- **Enable [[CoherentMethodAttribute](#)]** - Enables the usage of the [Coherent Method attribute](#)
- **Compensate gamma** - The View will have gamma corrected when linear color space is used. Whether you should use this property or not depends on the scene you are making - are there any additional cameras, what kind of rendering path they use, etc. Generally, this property comes handy when you use linear color space and the view is attached to a camera with deferred lighting or if your main camera uses deferred lighting and the camera with the view does not.
- **Always on top** - Indicates whether this value is z-buffer independent. If it is set to true, the view is rendered on top of everything.
- **Show JS dialogs** - Automatically handle JavaScript messages and authentication requests. If enabled, a visual dialog will be shown; otherwise it's up to the programmer to make the appropriate response. If you do not respond to the message, the page will block until you do. Examine `ViewListener.cs` and `CoherentDialog.cs` for a sample reply logic.
- **Software only rendering** - The View will be rendered without hardware acceleration.
- **Match camera size** - The View will be automatically resized to always match the size of the camera
- **Texture Filtering** - The filtering mode used for rendering the views. The two available modes are Point Filtering and Linear Filtering. ([View Filters - Desktop only](#) )



## 4.4 CoherentUIView properties for Mobile

- **URL** - Indicates the URL that will be initially loaded.
- **Width** - The width of the [Coherent](#) UI View in pixels.
- **Height** - The height of the [Coherent](#) UI View in pixels.
- **X Position** - The X of the [Coherent](#) UI View position relative to the upper-left corner of the device screen.
- **Y Position** - The Y of the [Coherent](#) UI View position relative to the upper-left corner of the device screen.
- **Retina rescale** - If set the content will be scaled to fit the View.
- **Enable WebGL Support** - This property is ignored due to vendor API limitations
- **Software only rendering - Android only** The View will be rendered without hardware acceleration. Some devices might render pages incorrectly because of hardware capabilities in which case software rendering helps. Note that advanced transformations and effects are not available with software rendering.
- **Input behavior** - Controls how the view handles user input (touches and gestures). The behavior only influences events that happen in the bounds of the View. If the user touches outside a view the event goes normally to the game.
  - Transparent - the user controls which input events to be taken by the UI and which to pass to the game. This is explained in the Mobile input management section ([Input forwarding - Mobile](#))
  - Take all - the View takes all the input - nothing is passed to the game
  - Take none - the View takes no input - everything goes to the game
- **Transparent** - Defines if the View supports transparency.
- **Smart input** - *Available for transparent Views.* Enables transparent input.
- **Auto UI messages** - When enabled, any event triggered in *JavaScript* on the `engine` object is forwarded to the game object containing the View.
- **Enable [CoherentMethodAttribute]** - Enables the usage of the [Coherent Method attribute](#)

## 4.5 CoherentUILiveGameView properties for Desktop

### Note

Live Game Views are a Desktop-only feature.

- **Name** - This is the string used as an identifier for the Live Game View. In the UI JavaScript code this string can be used to apply specific logic to only certain live views.
- **Width** - The Width of the Live Game View. The ImageData object created in the UI JavaScript DOM will have this width.
- **Height** - The Height of the Live Game View. The ImageData object created in the UI JavaScript DOM will have this height.
- **Source Camera** - The camera that will provide the images for the Live Game View. Everything rendered by this camera will be sent in the UI JavaScript universe and automatically updated every frame. If you omit a Source Camera than the source texture will be used.
- **Source Texture** - A texture to be used as source for the Live Game View images. If both a camera and a texture are specified - than the first frame in the live view will use the texture but all others will be provided by the Camera. If you omit specifying a Source Camera than only the texture will be used. If you change the content of the source texture and want to update the Live View linked to it you can do it by calling the "UpdateFromCurrentTexture" method of the Live Game View component.
- **Read alpha** - Specifies if to read the alpha value drawn by the Source Camera.



## Chapter 5

# Important points

### 5.1 Binding

Binding *C#*, *UnityScript* and *Boo* handlers to JavaScript callbacks is the same as binding for the .Net platform. You have to register handlers when the `UnityViewListener`'s `ReadyForBindings` event is fired. You can do that by using either `Coherent.UI.View.BindCall` (for single-cast delegates) and `Coherent.UI.View.RegisterForEvent` (when you have multiple subscribers for the event). For more details see the general reference documentation, chapter **Binding for .Net**.

```
private void RegisterBindings(int frame, string url, bool isMain)
{
    if (isMain)
    {
        var view = ViewComponent.View;
        if (view != null)
        {
            // When engine.call('NewGame') is executed in JavaScript,
            // the this.NewGame method will be called as well
            view.BindCall("NewGame", (System.Action)this.NewGame);
        }
    }
}
```

The `Coherent.UI.View` can be obtained using the `View` property of the `CoherentUIView` component.

To take advantage of the *Unity3D* component and message system each `CoherentUIView` has the `Intercept-AllEvents` property. If intercepting of all events is enabled, any event triggered in *JavaScript* on the engine object is forwarded to the game object containing the view. This is done by using `SendMessage` with `Coherent.UI.Binding.Value[]` containing all the event arguments.

```
engine.trigger('Event', 42);
// will execute SendMessage('Event', [42])
```

#### Note

Using this method of handling triggered events has some additional overhead over the direct `.Net Coherent.UI.View.RegisterForEvent` method.

### 5.2 CoherentMethod attribute for .NET scripts

Instead of handling the `ReadyForBindings` event and doing `BindCall` or `RegisterForEvent` by yourself, you can use the `CoherentMethod` attribute to decorate methods in components.

**Warning**

This attribute only works if the `CoherentUIView`'s `Enable Binding Attribute` is set to `true`. By default it is **false**.

The decorated methods will be automatically bound to the `View` owned by the `CoherentUIView` component in the Game Object. If the Game Object has no `CoherentUIView` component, the attribute has no effect. The `CoherentMethod` attribute has a string property for the JavaScript event name, and an optional boolean parameter that specifies whether the method is a *call* or *event* handler (*calls* can have only a single handler, while *events* may have many). Here's an example component using the attribute:

```
public class BindingComponent : MonoBehaviour {

    [Coherent.UI.CoherentMethod("NewGame")]
    void MyCallHandler()
    {
        Debug.Log("MyCallHandler called in response to engine.call('NewGame')");
    }

    [Coherent.UI.CoherentMethod("EnemySpotted", true)]
    void MyEventHandler()
    {
        Debug.Log("MyEventHandler called in response to engine.trigger('EnemySpotted')");
    }
}
```

See the [Binding Sample](#) for a complete example.

**Warning**

\* Binding methods using the `CoherentMethod` attribute is easier than doing it manually in `ReadyForBindings`, but presents possible performance penalties during game startup. When the `CoherentUIView` component is created, it searches all the other components in the host Game Object for methods marked with `CoherentMethod` using reflection. This can be a costly operation and to prevent undesirable slowdowns during startup the `Enable Binding Attribute` property for each `CoherentUIView` is set to **false** by default.

\* The `CoherentMethod` attribute currently does **NOT** support dynamically added components. Methods decorated with the attribute are only bound when the `CoherentUIView` component is created, which is usually when the Game Object it is part of is created.

**JavaScript and Unity3D**

Consult the *Binding for .NET* chapter in the general reference document. Check the [Binding Sample](#) and its walk-through in this guide for an example.

Briefly, Unity3D can call JavaScript using *events*; JavaScript can call Unity3D using *events* or *calls*.

- Events

Events can be called by JavaScript

```
engine.trigger("MyEvent", args);
```

or Unity3D

```
coherentUIView.View.TriggerEvent("MyEvent", args);
```

The "MyEvent" will be handled by any registered method in JavaScript

```
engine.on("MyEvent", function() {...});
```

or Unity3D

```
coherentUIView.View.RegisterForEvent("MyEvent", handlerMethod);
```

- Calls

Calls, unlike events, can have only one handler. They can also return values. To execute a *call* from JavaScript use

```
engine.call("MyCall", args);
```

It will be handled by a method registered using

```
coherentUIView.View.BindCall("MyCall", handlerMethod);
```

## 5.3 Namespaces

[Coherent](#) UI classes are placed in the [Coherent.UI](#) namespace for Desktop and [Coherent.UI.Mobile](#) for the Mobile version. You can check the [Coherent](#) UI files - for instance [CoherentUIView](#) and take a look at the beginning of the file at how the namespaces are imported depending on the Unity platform targeted. Exceptions to that rule are classes that cannot be in a namespace because Unity doesn't allow it, such as components that derive from [MonoBehaviour](#).

## 5.4 Subclassing CoherentUIView and UnityViewListener

The default [CoherentUIView](#) component and the [UnityViewListener](#) provide the most common functionality and are usually enough for normal usage of [Coherent UI](#). If you need custom behavior, you need to subclass them.

The class you derive from [UnityViewListener](#) would usually subscribe to various events that aren't handled by default. It is recommended not to override the [OnViewCreated](#) callbacks since the [UnityViewListener](#) class contains important logic that you would have to implement yourself otherwise.

The class you derive from [CoherentUIView](#) would only need to create an instance of your custom View Listener. This can be done by copying the [Update](#) method of [CoherentUIView](#) and editing it appropriately.

Note that when subclassing [CoherentUIView](#) you will no longer be able to view or edit the properties in the Inspector. That's because we're using C# properties in our component instead of fields and they are not automatically shown. To show the properties of a given C# script we need to make a new Editor script (inside the Editor folder of your Assets) that shows the properties for a specific type. We've already done that for [CoherentUIView](#), but you'll have to do it yourself for derived classes. The script contents should be the following:

```
using UnityEngine;
using System.Collections;
using UnityEditor;

[CustomEditor(typeof(<YourType>))]
public class <YourType>ViewEditor : Editor {

    private <YourType> m_Target;
    private CoherentPropertyField[] m_Fields;

    public void OnEnable() {
        m_Target = target as <YourType>;
        m_Fields = CoherentExposeProperties.GetProperties(m_Target);
    }

    public override void OnInspectorGUI() {
        if(m_Target == null)
            return;
        this.DrawDefaultInspector();
        CoherentExposeProperties.Expose(m_Fields);
    }
}
```

Just replace `<YourType>` with the actual name of your class.

### Note

In most cases, subclassing is unnecessary. See [Facebook Sample](#) for an example how to subscribe for `UnityViewListener` events.

## 5.5 Coherent UI system lifetime

Since initialization of the `CoherentUISystem` component is a costly operation, it is designed to be done once in the first scene of your game. The component itself has the same lifetime as the application. Since Unity tears down the state of the game when you load a new scene, the component is marked not be destroyed using the `DontDestroyOnLoad()` function. This makes it persist through scenes and is available using the `Object.FindObjectOfType` function. Getting the system can be done with the following line of code:

```
var uiSystem = Object.FindObjectOfType(typeof(CoherentUISystem)) as CoherentUISystem;
```

## 5.6 Customizing initialization of the Coherent UI System

When using only `CoherentUI` components, the [Coherent UI](#) System will be automatically initialized using the default parameters. These parameters define global system settings such as whether cookies are enabled, local storage and cache paths, debugger port and others. Check the `CoherentUISystem` component for a full list.

The [Coherent UI](#) System can be initialized with parameters other than the defaults in the following ways. Either drag the `CoherentUISystem` component to any object and edit the properties in the Inspector window, or edit the `CoherentUISystem.cs` script located in *Standard Assets/Scripts/CoherentUI* to fit your needs.

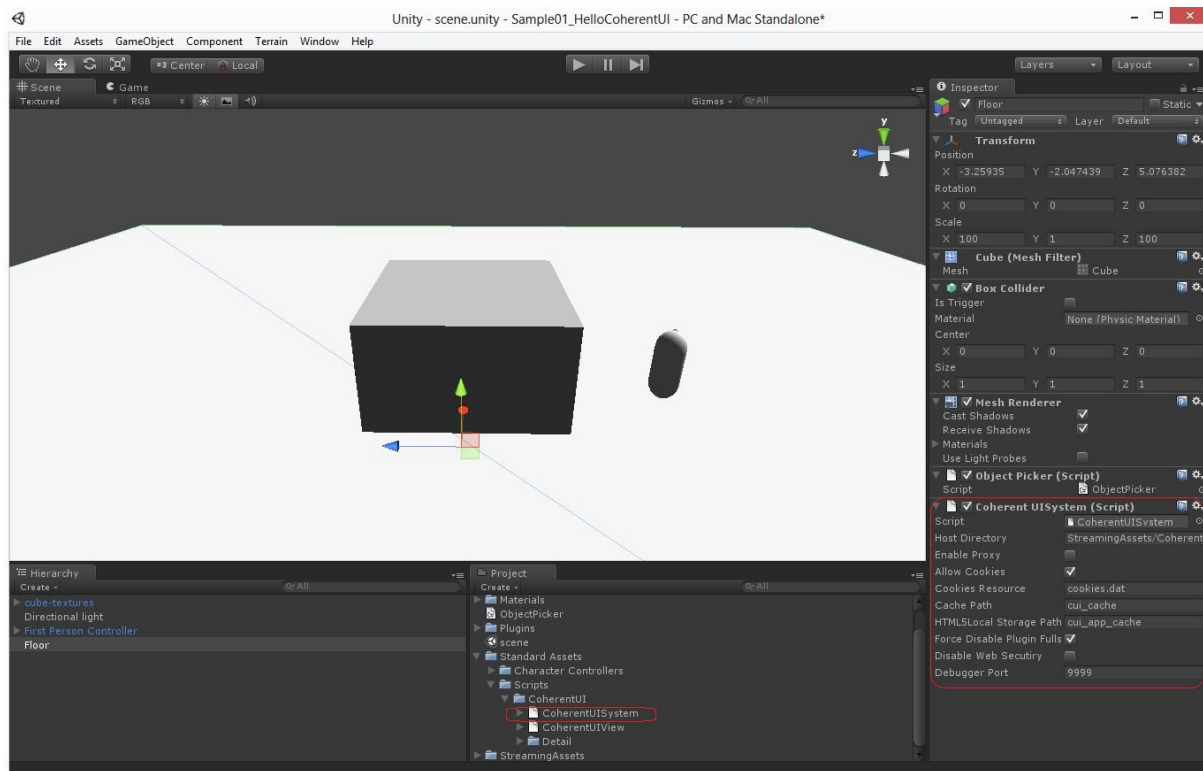


Figure 5.1: Custom CoherentUISystem component

The `CoherentUISystem` component is designed to be only one in the whole game. Adding more than one `CoherentUISystems` to your level will result in undefined behavior.

## 5.7 Update cycle

In a standard C++ or .NET application you need to poll [Coherent UI](#) for surfaces each frame using the `Update` and `FetchSurfaces` API calls. In our Unity integration, this is all hidden from you and you don't have to worry about it. The `CoherentUIViewRenderer` component issues rendering events which are handled by our library. All that's left for you is to drag a `CoherentUIView` component on an object!

## 5.8 Input forwarding - Desktop

A `CoherentUIView` requires focus to receive user input. Usually you'd want to forward input to a single view, but for flexibility [Coherent UI](#) supports multiple focused views. A `CoherentUIView`'s focus is controlled by the `ReceivesInput` property. To avoid confusion with multiple views, the property is set to **false** by default, meaning no input will be forwarded to [Coherent UI](#) unless you explicitly set it in your Game Object. It is **NOT** exposed in the Inspector, as it designed to be modified in code only.

### Note

The forwarding happens in the `CoherentUISystem`'s `LateUpdate` method, allowing you to do all your logic for input focus management in the `Update` methods of your components.

## 5.9 Input forwarding - Mobile

Due to differences in the input management for iOS/Android, [Coherent UI](#) provides a helper class for getting the list of touches for the current frame. The [Coherent](#) touch structure is `InputManager.CoherentTouch` - very much the same as Unity3D's `Touch` structure. The equivalent methods for `Input.touchCount` and `Input.GetTouch(index)` are `InputManager.TouchesCount` and `InputManger.GetTouch(index)`, respectively.

For **iOS** there's practically no difference between the [Coherent](#) and Unity3D touches. For **Android**, however, [Coherent](#) touches contain only touches that reached the game (i.e. the touch wasn't filtered using the techniques described in the next paragraph). The Unity3D supplied touches on Android aren't filtered and you get all of them when using the `Input.touches` array, regardless of whether they are on the UI or not. When using [Coherent UI](#) for mobile, it is recommended to use [Coherent](#) touches instead of Unity3D touches to avoid platform differences between iOS and Android.

On iOS/Android all [Coherent UI](#) Views are composited in 2D on-top of your Unity3D application content. When the user start a touch event or performs a gesture there is a mechanism that decides if that event should be handled by the UI or the application. It works like this: in the properties of the View the user can select one of three input modes for a every View - "Take All", "Take None", "Transparent". Keep in mind that all those modifiers are in effect only for events that happend are in the bounds of the View. If the user touches outside a particular View the event is always handled by the game.

- "Take All" specifies that all events are handled by the View and nothing goes to the game. It is usable if you want to have for instance an in-game browser. All touches on it should be handled by itself and not influence the game.
- "Take None" specifies that the View passes all input to the game. This is usable if you need to just show some non-interactive views or disable their input completely in some situation.
- "Transparent" specifies that the input is either handled by the View or the game - usable for HUDs, Menus etc. For Views with "Transparent" input mode the user is the one in charge of deciding if an element on the page is interactive - hence should receive input or is 'input transparent'.

**Note**

**iOS:** Input "Transparent" views work correctly **ONLY** if you have included *coherent.js* in your HTML page.

**Android:** Input on any view works correctly **ONLY** if you have included *coherent.js* in your HTML page.

Upon touch within a View, *Coherent UI Mobile* inspects the touched element:

- if the element has the CSS class *coui-noinput* it passes input to the game. The element does not accept input.
- if the element has the CSS class *coui-inputcallback*, a method called *couiInputCallback(x, y)* is called on the element with the coordinates of the event. It should return "true" if the user wants the element to handle the input and "false" if the game should handle it. This allows for custom fine-grained control in JavaScript on which elements are interactive.
- if the class *coui-noinput* and *coui-inputcallback* are missing from the element, it is assumed that it is interactive and takes the input.

To summarize: If a View has an Input State set to "Transparent" all elements are by default interactive and take input. You can mark elements with the CSS class *coui-noinput* to make them transparent to input. If you need more advanced logic when deciding if an element is interactive or not you can decorate it with *coui-inputcallback* and implement a method *couiInputCallback(x, y)* on it.

## 5.10 Mobile Preview

The preview for the mobile versions of *Coherent UI* allows you to easily see how your page behaves without testing it on a device or simulator. There are a few notable differences, however. First, the input redirection is based on the Android version, meaning you need to import *coherent.js* in your HTML files for it to work. All other notes for the input regarding Android apply for the preview as well. Another difference is that on the devices the *CoherentUIView* is always shown on top. Due to the specifics of Unity this is not easy to simulate automatically without interfering with some client functionality. That's why it's up to the user to simulate this behaviour in the preview. You should ensure that the camera you have your view on is always drawn last so the views are displayed the same, both in Editor preview and on the device.

## 5.11 Custom file handler

The *CoherentUISystem* component makes use of a static factory function object (*FileHandlerFactoryFunc*) to create the *FileHandler* object that is used reading URLs with the *coui* scheme. The default function returns a handler that reads resources from the path set by *Edit* → *Project Settings* → *Coherent UI* → *Select UI folder* for the Editor and in the Data folder for built games.

The factory function object is public and can be customized. The *FileHandler* it returns is passed to the UI initialization routine in the *Start* method of the *CoherentUISystem* component. That means the user should set the factory function prior to the invocation of the *Start* method of the components - e.g. in the *Awake* method.

**Note**

You can check the execution order of event function in Unity3D [on this page](#)

This is an example usage of a custom file handler:

```
public class CustomFileHandlerScript : MonoBehaviour {

    class CustomFileHandler : Coherent.UI.FileHandler
    {
        public override void ReadFile (string url, Coherent.UI.ResourceResponse response)
        {
            // Implementation here
        }
    }
}
```



```

    public override void WriteFile (string url, Coherent.UI.ResourceData resource)
    {
        // Implementation here
    }
}

//-----

void Awake()
{
    CoherentUISystem.FileHandlerFactoryFunc = () => { return new CustomFileHandler(); };
}
}

```

See [Archive resource demo](#) for an example.

## 5.12 UI Resources

Files for [Coherent UI](#) are by default selected from the folder set by *Edit* → *Project Settings* → [Coherent UI](#) → *Select UI folder*. Resources found there will be used by the editor and will automatically be copied in your game upon build.

The selected UI resources folder is per-user so that different developers working on the game can have their UI folders wherever they want on their machine. You can also set a per-project folder for the UI resources. This is done by extending the `CoherentPostProcessor` class by setting a static const setting named `ProjectUIResources`:

```

public partial class CoherentPostProcessor {
    public static string ProjectUIResources = "relative/path/to/ui/resources";
}

```

The per-project path must be relative to the folder of the project and the extension class should live under the 'Editor' folder in Unity. This feature is very handy also if you build your game on machines that can't start Unity and you use the command line. The per-user setting overrides the per-project one so that developers can still put their resources wherever they want.

### Warning

When using the "Build & run" option for Android, the required resources will **NOT** be packaged because this is done as a post-build step. Unity3D pushes the non-repacked APK first, and then executes the post-build step which results in an APK without UI resources and including unneeded files. To work around that, you should either use the "Build" option, which produces an APK on your computer that can be installed on a device using the ADB tool in the Android SDK, or generate an Eclipse project and run it from the IDE.

## 5.13 Click-to-focus Views - Desktop only

When the `ClickToFocus` property is enabled on a View, it will automatically take **all** the input focus when you click on it and lose it when you click somewhere else. When focused, all mouse and keyboard input will be forwarded to the View. "Click-to-focus" views have their `ReceivesInput` property managed by the `CoherentUISystem` and you should **NOT** set it manually. If you do so, you'll receive a warning message in Unity3D and the input forwarding behavior will be unexpected.

### Warning

There is no way to prevent the standard Unity character controller scripts from moving the character, even when the input event is *used*. You have to manually disable your character controller when you want the user to type in in a view and stands still. For an explanation see the [Menu And HUD](#) sample.

**Note**

"Click-to-focus" views perform raycasts to obtain the object in the 3D world below the cursor. For a raycast to report the correct texture coordinates of the hitpoint, you need to set up a *Mesh Collider* on the objects with [Coherent](#) UI Views placed.

"Click-to-focus" Views are useful in cases when you want keyboard input forwarded to a View regardless of the mouse position, e.g. input fields.

**Note**

To function properly, "Click-to-focus" views need to know which camera is the main camera in the scene. For simple scenes, this can be obtained from the `Camera.main` property in Unity3D. This is what [Coherent](#) UI assumes is the main camera, and obtains it in the `MonoBehaviour.OnEnable` callback, which is executed when a scene is loaded. This is done only if there is no currently set camera for the `m_MainCamera` field so it does not interfere with custom user code. For complex scenes with multiple cameras, however, it is up to you to set the public `m_MainCamera` field to the appropriate camera (also visible in the Inspector window).

Which `CoherentUIViews` receive input is up to your gameplay needs. Here we'll walk you through a simple script that you'll see used in the samples - it forwards input to the closest view under the cursor. First, it sets the `ReceivesInput` property to *all* views to **false**. Then it queries the `CoherentUIView` attached to the main camera (if any) whether the mouse is over a solid or transparent pixel (make sure to set the `SupportClickThrough` property of the HUD view to **true** to support this operation). If the mouse is over a solid pixel, then the HUD is focused and receives input. Otherwise, a raycast is generated that finds the object under the cursor. If that object has a `CoherentUIView` component, that's what gets the focus.

Here's the script itself:

To summarize, you can apply any logic you like for input forwarding - e.g. forward input to objects in the view frustum, HUD only, etc. [Coherent](#) UI supports multiple focused views. View focus can be modified using the `ReceivesInput` property of `CoherentUIView` which is controlled only by the script code.

You can also mark views as "Click to focus" which makes them take all the focus when clicking them (and lose focus when clicking somewhere else). You should take care not to set the `ReceivesInput` property on "Click to focus"-enabled Views as it is automatically managed. Setting the `ReceivesInput` property on such views manually will result in unexpected behavior and will produce a warning message.

## 5.14 View Filters - Desktop only

[Coherent](#) UI provides three types of filtering:

- Point Filtering - applies point filtering to the view texture.
- Linear Filtering - applies linear filtering to the view texture.

## 5.15 Hit testing - Desktop only

**Note**

For an explanation about input forwarding for Mobile check [Input Forwarding Mobile](#)

Forwarding input to Views attached to the camera is straight-forward - you only have to mark your view as an input receiver using the `ReceivesInput` property of `CoherentUIView`. The mouse position will be obtained from the `Input.mousePosition` property.

If you want to forward an input event to a View that's attached on an object in the 3D world, you'll have to do a bit more work. You'll have to use a raycast to find the object below the cursor and then transform the texture coordinates of the hit point into the space of the [Coherent UI View](#). Note that Unity provides texture coordinates only when the object has a Mesh Collider component attached. The coordinates must be transformed from [0, 1] to [view.Width, view.Height]. This can usually be done simply by multiplying the coordinates by the dimensions of the View (which are available as properties). Then, you have to set the resulting coordinates to the CoherentUIView component using the SetMousePosition method. Check [Input Forwarding](#) for an example script that forwards input to the view on the object that is currently below the cursor. Note that in the samples the `MeshCollider` component has the same geometry as the renderable mesh. This may not always be true and in such cases you would have to make a transformation of the coordinates that works for you.

For a sample how hit testing works see the [Menu and HUD](#) sample.

## 5.16 Mobile Surface Views

Displaying CoherentUI view on a surface is straightforward for Desktop platforms. Unfortunately the performance and API restrictions of the current platforms do not allow us to fully support views on surfaces for mobile. *Mobile Surface Views* allow displaying of HTML5 content on a surface in mobile games with the following limitations:

- the view must be explicitly updated using the `MobileSurfaceView.UpdateView` method.
- there is no input for the view. Input could be simulated using JavaScript.

To create a *Mobile Surface View*, simply add the component to a object with a renderer component. The `MobileSurfaceView` component will create a new texture for the renderer and replace the main texture of the material with the view.

See also the [Mobile Surface Sample](#).

## 5.17 Logging

[Coherent UI](#) logs are automatically redirected to the Unity console (or game log for built games) using the `Debug.Log` method. You can control the minimum severity of the [Coherent UI](#) logs when initializing the [Coherent UI System](#).

## 5.18 Live Game Views

### Note

Live Game Views are a Desktop-only feature.

Live Game Views are one of the most powerful yet easy to use fetures in [Coherent UI](#) for Unity 3D. They allow you to have 3D rendered images (by a Unity3D Camera) as part of the UI itself. The images are automatically updated in a high-performance way and made available to the UI JavaScript code.

Via Live Game Views in a real game you can trivially have 3D animated unit portraits, animated items in the UI, 3D mini-maps, security cameras etc..

The steps to get a Live Game View are trivial:

1. Add the [Coherent UI](#) "Live Game View" Component to a GameObject that already has a [Coherent UI View](#).
2. Give the link a Name
3. Set the size
4. Drag a Camera that will send the data

5. (Optional) You can select a texture too in the Component. If you don't select a Camera - this texture will be rendered instead. In this way you can send just textures in the UI.

The "Live Game View" Component represents a link between the View and a Unity Camera or a texture. When you attach it - it will automatically start sending the updated image drawn by the linked Camera to the JavaScript code of the page currently displayed by the View.

Every such link has a Name that is used to identify it inside the page's JS code. You can have as many Live Game Views attached to the same [Coherent](#) UI View as you want.

After you press "Play", the Live Game View will be operational and everything the Camera "sees" will be available in the UI as an ImageData object.

To use it - add a "canvas" element in the page and a "onEngineImageDataUpdated" function to it. This function will be called every time the image is updated and the name will be passed so that you can identify just the one you need.

In the "onEngineImageDataUpdated" function you can do whatever you need with the image - draw it in the canvas, apply filters, write text etc..

This snippet shows a sample UI JavaScript function that draws the image in a "canvas" element and writes text on it:

```
window.onload = function() {  
    var c = document.getElementById("myCanvas");  
    c.onEngineImageDataUpdated = function (name, image) {  
        var c = document.getElementById("myCanvas");  
        var ctx = c.getContext("2d");  
  
        ctx.clearRect(0, 0, c.width, c.height);  
        ctx.putImageData(image, 0, 0);  
        ctx.fillStyle = "white";  
        ctx.font = "16pt Arial";  
        ctx.fillText("Live Game View!", 15, 25);  
    }  
}
```

As you can see we just listen for when a new image has arrived and re-draw the content of the canvas named "myCanvas". The "name" parameter received is the "Name" of the Live Game View as specified in the Unity Inspector and allows us to identify the different links.

#### Note

For a sample scene with Live Game Views check the "Sample\_LiveGameViews" scene available in the [Coherent](#) UI package.

## Chapter 6

# Requirements

To use [Coherent](#) UI SDK in your project, please make sure that the following minimum system requirements are met by your development environment and are compatible with your project requirements for the end-users.

### Software Requirements

#### Windows

- Windows XP (required Service Pack 3)
- Windows Vista (required Service Pack 2)
- Windows 7 (required Service Pack 1)
- Windows 8
- Windows 8.1

Visual Studio 2010, 2012, 2013 is required to build [Coherent](#) UI SDK. DirectX SDK is required for older versions of Visual Studio to build some of the samples.

#### Note

- All Windows systems must have the latest Service Pack provided by Microsoft installed.
- DirectX shared textures require Windows Vista or later.
- Latest drivers provided by the hardware vendor are required.
- For Windows 8 and 8.1 only desktop mode applications are supported
- WinRT is not supported
- The [Coherent](#) UI Samples require the [Microsoft DirectX End-User Runtime \(June 2010\)](#) to work compile (or at least d3dx9\_43.dll and D3DCompiler\_43.dll should be added to the output folder).

#### Linux

Generally any distribution similar to Ubuntu 12.04 LTS.

- glibc 2.14 or later
- libstdc++6 or later
- GTK 2.24.10 or later
- GCC 4.6.3 or later

**Note**

Only the latest video card drivers are supported. OpenGL 3.0 support is required.

**Mac OS X**

- 10.7 or later
- XCode 4.4 or later

**Note**

Limited support for Mac OS 10.6.8 is available, but extensive testing should be performed, and some features might not work.

**Hardware Requirements**

- Intel or AMD CPU architectures are supported (no ARM support)
- Dual-core CPU is recommended
- 512 MB RAM
- 350 MB HDD space for development
- 40-80 MB HDD space for runtime libraries

*Video adapter:*

- Support for shader model 3.0 is required
- DirectX 9.0c or later support
- OpenGL 3.0\* or later support
- Full non-power-of-two textures support
- Dedicated video card recommended
- Certain features of [Coherent](#) UI may be available on earlier versions of OpenGL but we don't test that configuration and results may be unpredictable.
- On configurations with multiple video cards - SLI, Optimus, etc., the faster card should be enabled, and your users should make sure that the same adapter runs both your application and the [Coherent](#) UI rendering processes.

## Chapter 7

# Namespace Documentation

### 7.1 Package Coherent

#### Namespaces

- package [UI](#)

### 7.2 Package Coherent.UI

#### Namespaces

- package [Binding](#)

#### Classes

- class [HTTPHeader](#)  
*Encapsulates a HTTP header field with it's content.*
- class [BrowserView](#)  
*Class that encapsulates a browser [Coherent::UI::View](#)*
- class [BrowserViewListener](#)  
*Interface that allows clients to listen to [Coherent::UI::ViewListener](#)*
- class [BrowserViewListenerBase](#)  
*Interface all browser view listeners inherit. For an easier to use interface inherit instead -. [Coherent::UI::BrowserViewListener](#)*
- class [Certificate](#)  
*Represents an SSL certificate.*
- class [CertificatePrincipal](#)  
*Represents certificate principal.*
- class [ChildViewInfo](#)  
*Helper structure for creating child views.*
- class [Download](#)  
*Encapsulates a download task.*
- class [EventListener](#)  
*Abstract interface to listen to System-related events.*
- class [EventModifiersState](#)  
*The state of the key modifiers when an event happens.*
- class [EventMouseModifiersState](#)

*The state of the mouse modifiers when an event happens.*

- class [FileHandler](#)

*Abstract interface that allows clients to provide their own file-handling functionality.*

- class [ILogHandler](#)

*Interface to allow custom logging.*

- class [ImageData](#)

*This class represents a link to a*

- class [KeyEventData](#)

*A keyboard event.*

- class [MediaStreamDevice](#)

*Represents a media stream device.*

- class [MediaStreamRequest](#)

*Represents a request for media stream.*

- class [MouseEventData](#)

*A mouse event.*

- class [ResourceData](#)

*Abstract interface providing data for storing resources.*

- class [ResourceResponse](#)

*Abstract interface for responding to read resource requests.*

- class [SurfaceResponse](#)

*Interface that signals for creation of rendering surfaces.*

- class [SystemError](#)

*Encapsulates a system-related error.*

- class [SystemSettings](#)

*Encapsulates the settings of the*

- class [SystemSettingsBase](#)

*Encapsulates the settings of the*

- class [TouchEventData](#)

*A touch event.*

- class [UISystem](#)

*Encapsulates basic [UI](#) system functionality.*

- class [URLRequest](#)

*represents a single URL request Allows monitoring and modifications of URL requests*

- class [View](#)

*Class that encapsulates a [UI](#)*

- class [ViewError](#)

*Encapsulates a view-related error.*

- class [ViewInfo](#)

*Encapsulates the options of a*

- class [ViewListener](#)

*Interface that allows clients to listen to*

- class [ViewListenerBase](#)

*Interface all view listeners inherit. For an easier to use interface inherit instead -. [Coherent::UI::ViewListener](#)*

- struct [BoundEventHandle](#)

*Handle for registered delegates*



## Enumerations

- enum [CertificateStatus](#)  
*Enum containing the possible certificate validation errors.*
- enum [CursorTypes](#)  
*Enum containing all the cursor types that a view can signal.*
- enum [Direct3DInterfaceType](#)  
*Types of interface used for DirectX.*
- enum [DownloadErrorType](#) { [DownloadErrorType.DET\\_Success](#), [DownloadErrorType.DET\\_Download-OperationFailed](#) }
- enum [FileSelectionMode](#)  
*Enumeration for file selection modes.*
- enum [JavaScriptMessageType](#)  
*Enum containing the possible types of a javascript message.*
- enum [MediaStreamType](#)  
*Enumeration of media stream types.*
- enum [RenderingDeviceType](#)  
*Types of rendering backend.*
- enum [ScriptCallErrorType](#) { [ScriptCallErrorType.SCE\\_Success](#), [ScriptCallErrorType.SCE\\_ArgumentType](#), [ScriptCallErrorType.SCE\\_NoSuchMethod](#), [ScriptCallErrorType.SCE\\_NoResult](#) }
- enum [SystemErrorType](#) { [SystemErrorType.SE\\_Success](#), [SystemErrorType.SE\\_InternalError](#), [SystemErrorType.SE\\_InvalidKey](#), [SystemErrorType.SE\\_MissingComponent](#) }  
*System-related error-codes.*
- enum [TextInputControlType](#)  
*Enum containing the text input types reported for IME.*
- enum [UserAgentType](#)  
*Predefined values for the used User Agent. If you select "Custom" you also have to set the CustomUserAgentString.*
- enum [ViewErrorType](#) { [ViewErrorType.VE\\_Success](#), [ViewErrorType.VE\\_FailedCreation](#), [ViewErrorType.VE\\_FailedRendering-Creation](#), [ViewErrorType.VE\\_ClickThroughFailed](#), [ViewErrorType.VE\\_InvalidPath](#), [ViewErrorType.VE\\_ChildCompositionFailed](#), [ViewErrorType.VE\\_InvalidCall](#), [ViewErrorType.VE\\_FrameAlreadyInFlight](#), [ViewErrorType.VE\\_FrameNotReady](#), [ViewErrorType.VE\\_FrameNotRequested](#), [ViewErrorType.VE\\_Abnormal-Termination](#), [ViewErrorType.VE\\_QueryNotReady](#) }  
*View-related error codes.*
- enum [ViewType](#)  
*Enum containing the possible view types.*

### 7.2.1 Enumeration Type Documentation

#### 7.2.1.1 enum Coherent.UI.CertificateStatus

Enum containing the possible certificate validation errors.

#### 7.2.1.2 enum Coherent.UI.CursorTypes

Enum containing all the cursor types that a view can signal.

#### 7.2.1.3 enum Coherent.UI.Direct3DInterfaceType

Types of interface used for DirectX.

#### 7.2.1.4 enum Coherent.UI.DownloadErrorType

Enumerator

**DET\_Success** Indicates everything went fine.

**DET\_DownloadOperationFailed** Indicates the download operation failed.

#### 7.2.1.5 enum Coherent.UI.FileSelectionMode

Enumeration for file selection modes.

#### 7.2.1.6 enum Coherent.UI.JavaScriptMessageType

Enum containing the possible types of a javascript message.

#### 7.2.1.7 enum Coherent.UI.MediaStreamType

Enumeration of media stream types.

#### 7.2.1.8 enum Coherent.UI.RenderingDeviceType

Types of rendering backend.

#### 7.2.1.9 enum Coherent.UI.ScriptCallErrorType

Enumerator

**SCE\_Success** Indicates that the call was successful.

**SCE\_ArgumentType** Indicates that a script provided argument has different type than the one expected.

**SCE\_NoSuchMethod** Indicates that there is no such method.

**SCE\_NoResult** Indicates that there is no result for this call.

#### 7.2.1.10 enum Coherent.UI.SystemErrorType

System-related error-codes.

Enumerator

**SE\_Success** Indicates no error.

**SE\_InternalError** Indicates a serious system internal error.

**SE\_InvalidKey** Indicates that the user supplied an invalid activation key for this version of the product.

**SE\_MissingComponent** Indicates that a component required by [Coherent UI](#) is missing.

#### 7.2.1.11 enum Coherent.UI.TextInputControlType

Enum containing the text input types reported for IME.

#### 7.2.1.12 enum Coherent.UI.UserAgentType

Predefined values for the used User Agent. If you select "Custom" you also have to set the CustomUserAgentString.

## 7.2.1.13 enum Coherent.UI.ViewErrorType

View-related error codes.

Enumerator

- VE\_Success** Indicates no error.
- VE\_FailedCreation** Indicates failure to create a view.
- VE\_FailedRenderingCreation** Indicates failure to create the rendering resources associated with a view.
- VE\_ClickThroughFailed** Indicates an error during click-through a query.
- VE\_InvalidPath** Indicates that the view was redirected to an invalid path.
- VE\_ChildCompositionFailed** Indicates that the composition of a child widget failed.
- VE\_InvalidCall** Indicates a call on a type of view that is not called.
- VE\_FrameAlreadyInFlight** Indicates that a frame is already in-flight.
- VE\_FrameNotReady** Indicates that a frame is still being rendered.
- VE\_FrameNotRequested** Indicates that a new frame was not requested.
- VE\_AbnormalTermination** Indicates an abnormal termination of the view.
- VE\_QueryNotReady** Indicates that the mouse query hasn't finished yet.

## 7.2.1.14 enum Coherent.UI.ViewType

Enum containing the possible view types.

## 7.3 Package Coherent.UI.Binding

### Classes

- class [CallbackArguments](#)  
*Holds the arguments for the generic callback not handled by a registered delegate*
- struct [BoundObject](#)  
*Wrapper class for .Net objects exposed to [Coherent UI](#)*
- class [CoherentProperty](#)  
*Specify a property / field or method visible to [Coherent UI](#)*
- class [CoherentType](#)  
*Specify which properties and fields of a type are visible to [Coherent UI](#)*
- class [UnsupportedPrimitiveTypeException](#)  
*Thrown when trying to bind a value of unsupported primitive type such as long*
- class [InvalidValueCastException](#)  
*Thrown when casting a [Value](#) to an incompatible type*
- class [Value](#)  
*Type for representing generic JavaScript values*
- class [ValueObject](#)  
*Class for compound JavaScript objects, behaves like a Dictionary<string, Value>*

## Enumerations

- enum [PropertyBindingFlags](#) { [PropertyBindingFlags.Explicit](#) = 0, [PropertyBindingFlags.Instance](#) = 1, [PropertyBindingFlags.Static](#) = 2, [PropertyBindingFlags.All](#) = 3 }

*Determines the set of the bound properties for this type*

- enum [ValueType](#) { [ValueType.Null](#), [ValueType.Boolean](#), [ValueType.Integer](#), [ValueType.UInteger](#), [ValueType.Number](#), [ValueType.String](#), [ValueType.Array](#), [ValueType.Object](#), [ValueType.SByte](#) = 10, [ValueType.Byte](#) }

*Specifies the type of [Coherent.UI.Binding.Value](#)*

### 7.3.1 Enumeration Type Documentation

#### 7.3.1.1 enum [Coherent.UI.Binding.PropertyBindingFlags](#)

Determines the set of the bound properties for this type

##### Enumerator

**Explicit** Bind only properties and fields with [CoherentProperty](#) attribute

**Instance** Bind only instance properties and fields

**Static** Bind only static properties and fields

**All** Bind all instance and static properties and fields

#### 7.3.1.2 enum [Coherent.UI.Binding.ValueType](#)

Specifies the type of [Coherent.UI.Binding.Value](#)

##### Enumerator

**Null** empty value

**Boolean** boolean value

**Integer** integer value

**UInteger** unsigned integer value

**Number** double value

**String** string value

**Array** array value

**Object** object value

**SByte** signed byte value

**Byte** unsigned byte value

## Chapter 8

# Class Documentation

### 8.1 Coherent.UI.BoundEventHandle Struct Reference

Handle for registered delegates

#### 8.1.1 Detailed Description

Handle for registered delegates

### 8.2 Coherent.UI.Binding.BoundObject Struct Reference

Wrapper class for .Net objects exposed to [Coherent UI](#)

#### Static Public Member Functions

- static [BoundObject BindMethods](#) (object target)  
*expose object's methods to [Coherent UI](#)*

#### 8.2.1 Detailed Description

Wrapper class for .Net objects exposed to [Coherent UI](#)

#### 8.2.2 Member Function Documentation

8.2.2.1 static **BoundObject** Coherent.UI.Binding.BoundObject.BindMethods ( object *target* ) [inline],[static]

expose object's methods to [Coherent UI](#)

##### Parameters

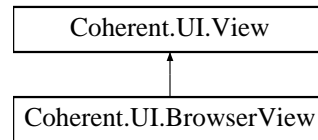
<i>target</i>	instance
---------------	----------

##### Returns

## 8.3 Coherent.UI.BrowserView Class Reference

Class that encapsulates a browser [Coherent::UI::View](#)

Inheritance diagram for Coherent.UI.BrowserView:



### Public Member Functions

- override [ViewType](#) `GetViewType ()`  
*Get The type of this view.*
- virtual void [GoBack \(\)](#)  
*Navigates back to the previous URL, if any.*
- virtual void [GoForward \(\)](#)  
*Navigates to the next URL, if any (i.e. available after navigating back)*
- virtual void [GetHistory \(\)](#)  
*Issues a request for obtaining the navigation history for the view. [BrowserViewListener::OnHistoryObtained](#)*

#### 8.3.1 Detailed Description

Class that encapsulates a browser [Coherent::UI::View](#)

#### 8.3.2 Member Function Documentation

**8.3.2.1** virtual void `Coherent.UI.BrowserView.GetHistory ( )` `[inline]`, `[virtual]`

Issues a request for obtaining the navigation history for the view. [BrowserViewListener::OnHistoryObtained](#)

**8.3.2.2** override `ViewType` `Coherent.UI.BrowserView.GetViewType ( )` `[inline]`, `[virtual]`

Get The type of this view.

Returns

the type of the view

Reimplemented from [Coherent.UI.View](#).

**8.3.2.3** virtual void `Coherent.UI.BrowserView.GoBack ( )` `[inline]`, `[virtual]`

Navigates back to the previous URL, if any.

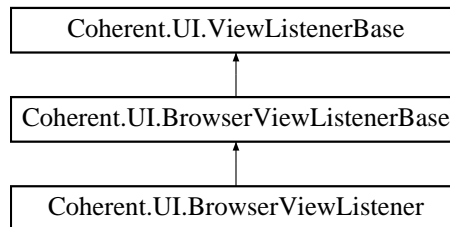
**8.3.2.4** virtual void `Coherent.UI.BrowserView.GoForward ( )` `[inline]`, `[virtual]`

Navigates to the next URL, if any (i.e. available after navigating back)

## 8.4 Coherent.UI.BrowserViewListener Class Reference

Interface that allows clients to listen to [Coherent::UI::ViewListener](#)

Inheritance diagram for Coherent.UI.BrowserViewListener:



### Public Member Functions

- override void [Release](#) ()  
*Called when the listener is no longer needed by the [UI](#) system.*
- override void [OnViewCreated](#) ([View](#) view)  
*Called when the requested*
- override void [OnDraw](#) ([CoherentHandle](#) handle, bool usesSharedMemory, int width, int height)  
*Called when a new surface has been drawn and is ready to use by the client.*
- override void [OnFinishLoad](#) (int frameId, string validatedPath, bool isMainFrame, int statusCode, [HTTPHeader\[\]](#) headers)  
*Called when a frame has been successfully loaded.*
- override void [OnFailLoad](#) (int frameId, string validatedPath, bool isMainFrame, string error)  
*Called when a frame has been failed loading.*
- override void [OnURLRequest](#) (string url, [URLResponse](#) response)  
*Called before an URL request is made. The default implementation allows all requests.*
- override void [OnReadyForBindings](#) (int frameId, string path, bool isMainFrame)  
*Called when a frame is ready for bindings.*
- override void [OnBindingsReleased](#) (int frameId, string path, bool isMainFrame)  
*Called when the bindings for frame are released.*
- override void [OnStartLoading](#) ()  
*Called when a new path has started loading.*
- override void [OnStopLoading](#) ()  
*Called when all load operations have completed.*
- override void [OnNavigateTo](#) (string path)  
*Called when the view starts navigation to a new path.*
- override void [OnError](#) ([ViewError](#) error)  
*Called when an error occurs for this specific*
- override void [OnScriptMessage](#) ([ViewListenerBase.MessageLevel](#) level, string message, string sourceId, int line)  
*Called when a message is sent from a script running in this specific*
- override void [OnCursorChanged](#) ([CursorTypes](#) cursor)  
*Called when the cursor has changed internally in the*
- override void [OnCallback](#) (string eventName, [Binding.CallbackArguments](#) arguments)  
*Called by the [UI](#) when there is no registered handler for this event.*
- override void [OnJavaScriptMessage](#) (string message, string defaultPrompt, string frameUrl, int messageType)  
*Called when the view triggered a javascript message box, i.e. an alert, confirmation dialog or a prompt dialog.*
- override void [OnGetAuthCredentials](#) (bool isProxy, string host, uint port, string realm, string scheme)

- Called when a view requires authentication credentials.*

  - override void [CreateSurface](#) (bool sharedMemory, uint width, uint height, [SurfaceResponse](#) response)
 

*Called when the The format for DirectX9 must be D3DFMT\_A8R8G8B8 The format for DirectX10 and DirectX11 must be B8G8R8A8\_UNORM*
  - override void [DestroySurface](#) (CoherentHandle surface, bool usesSharedMemory)
 

*Called when a surface is unneeded anymore and should be destroyed This function can be called from a thread different than the main UI system thread in order to support client applications with multi-threaded rendering.*
  - override void [OnCertificateError](#) (string url, [CertificateStatus](#) status, [Certificate](#) certificate, [CertificateError-Response](#) response)
 

*Called when there is an error with the certificate of a particular URL. the certificate and response pointers are valid only for the duration of this call*
  - override void [OnRequestMediaStream](#) ([MediaStreamRequest](#) request)
 

*Called when the view requests access to a media stream. Media streams are the audio capture (microphone) and video capture (camera) devices on the system.*
  - override void [OnTextInputTypeChanged](#) ([TextInputControlType](#) type, bool canComposeInline)
 

*Called when the current text input control changes (i.e. the user click an edit-box). Use this method to decide when to allow for IME input. the method will be called ONLY if IME is activated on this View. [View::IMEActivate](#)*
  - override void [OnCaretRectChanged](#) (uint x, uint y, uint width, uint height)
 

*Called when the caret changes during IME composition. You can use this method to correctly position a custom IME control & candidate list.*
  - override void [OnIMEShouldCancelComposition](#) ()
 

*Called when the user must cancel the IME composition due to an event in the*
  - override void [OnFileSelectRequest](#) ([FileSelectRequest](#) request)
 

*Called when the view requests file selection. It could be either single file, directory or multiple files.*

## Events

- CoherentUI\_OnViewCreated [ViewCreated](#)

*Fired when the requested*
- CoherentUI\_OnDraw [Draw](#)

*Fired when a new surface has been drawn and is ready to use by the client.*
- CoherentUI\_OnFinishLoad [FinishLoad](#)

*Fired when a frame has been successfully loaded.*
- CoherentUI\_OnFailLoad [FailLoad](#)

*Fired when a frame has been failed loading.*
- CoherentUI\_OnURLRequest [URLRequest](#)

*Fired before an URL request is made. The default implementation allows all requests.*
- CoherentUI\_OnReadyForBindings [ReadyForBindings](#)

*Fired when a frame is ready for bindings.*
- CoherentUI\_OnBindingsReleased [BindingsReleased](#)

*Fired when the bindings for frame are released.*
- CoherentUI\_OnStartLoading [StartLoading](#)

*Fired when a new path has started loading.*
- CoherentUI\_OnStopLoading [StopLoading](#)

*Fired when all load operations have completed.*
- CoherentUI\_OnNavigateTo [NavigateTo](#)

*Fired when the view starts navigation to a new path.*
- CoherentUI\_OnError [Error](#)

*Fired when an error occurs for this specific*
- CoherentUI\_OnScriptMessage [ScriptMessage](#)

*Fired when a message is sent from a script running in this specific*
- CoherentUI\_OnCursorChanged [CursorChanged](#)



- Fired when the cursor has changed internally in the*
- CoherentUI\_OnCallback [Callback](#)
- Fired by the [UI](#) when there is no registered handler for this event.*
- CoherentUI\_OnJavaScriptMessage [JavaScriptMessage](#)
- Fired when the view triggered a javascript message box, i.e. an alert, confirmation dialog or a prompt dialog.*
- CoherentUI\_OnGetAuthCredentials [GetAuthCredentials](#)
- Fired when a view requires authentication credentials.*
- CoherentUI\_OnCertificateError [CertificateError](#)
- Fired when there is an error with the certificate of a particular URL. the certificate and response pointers are valid only for the duration of this call*
- CoherentUI\_OnRequestMediaStream [RequestMediaStream](#)
- Fired when the view requests access to a media stream. Media streams are the audio capture (microphone) and video capture (camera) devices on the system.*
- CoherentUI\_OnTextInputTypeChanged [TextInputTypeChanged](#)
- Fired when the current text input control changes (i.e. the user click an edit-box). Use this method to decide when to allow for IME input. the method will be called ONLY if IME is activated on this [View](#). [View::IMEActivate](#)*
- CoherentUI\_OnCaretRectChanged [CaretRectChanged](#)
- Fired when the caret changes during IME composition. You can use this method to correctly position a custom IME control & candidate list.*
- CoherentUI\_OnIMEShouldCancelComposition [IMEShouldCancelComposition](#)
- Fired when the user must cancel the IME composition due to an event in the*
- CoherentUI\_OnFileSelectRequest [FileSelectRequest](#)
- Fired when the view requests file selection. It could be either single file, directory or multiple files.*

## Additional Inherited Members

### 8.4.1 Detailed Description

Interface that allows clients to listen to [Coherent::UI::ViewListener](#)

### 8.4.2 Member Function Documentation

8.4.2.1 `override void Coherent.UI.BrowserViewListener.CreateSurface ( bool sharedMemory, uint width, uint height, SurfaceResponse response ) [inline],[virtual]`

Called when the The format for DirectX9 must be D3DFMT\_A8R8G8B8 The format for DirectX10 and DirectX11 must be B8G8R8A8\_UNORM

#### Parameters

<i>sharedMemory</i>	true if the surface should be created in shared memory (4 * width * height bytes); false if a shared texture must be created.
<i>width</i>	the width of the surface in pixels
<i>height</i>	the height of the surface in pixels
<i>response</i>	<ul style="list-style-type: none"> <li>• object to hold the response when the surface is created or fails it's creation - must be signaled</li> </ul>

#### Returns

a handle to the created surface

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.4.2.2** `override void Coherent.UI.BrowserViewListener.DestroySurface ( CoherentHandle surface, bool usesSharedMemory )`  
`[inline], [virtual]`

Called when a surface is unneeded anymore and should be destroyed This function can be called from a thread different than the main [UI](#) system thread in order to support client applications with multi-threaded rendering.

## Parameters

<i>surface</i>	handle to the surface
<i>usesSharedMemory</i>	determines whether the surface parameter is a handle to shared memory or shared texture

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.4.2.3** `override void Coherent.UI.BrowserViewListener.OnBindingsReleased ( int frameId, string path, bool isMainFrame )`  
`[inline], [virtual]`

Called when the bindings for frame are released.

## Parameters

<i>frameId</i>	the id of the frame
<i>path</i>	the path in the frame
<i>isMainFrame</i>	true if this is the main frame of the view

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.4.2.4** `override void Coherent.UI.BrowserViewListener.OnCallback ( string eventName, Binding.CallbackArguments arguments )`  
`[inline], [virtual]`

Called by the [UI](#) when there is no registered handler for this event.

## Parameters

<i>eventName</i>	name of the event
<i>arguments</i>	arguments of the event invocation

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.4.2.5** `override void Coherent.UI.BrowserViewListener.OnCaretRectChanged ( uint x, uint y, uint width, uint height )`  
`[inline], [virtual]`

Called when the caret changes during IME composition. You can use this method to correctly position a custom IME control & candidate list.

## Parameters

<i>the</i>	x position of the selection caret
<i>the</i>	y position of the selection caret
<i>the</i>	width of the selection caret
<i>the</i>	height of the selection caret

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.4.2.6** `override void Coherent.UI.BrowserViewListener.OnCertificateError ( string url, CertificateStatus status, Certificate certificate, CertificateErrorResponse response )`  
`[inline], [virtual]`

Called when there is an error with the certificate of a particular URL. the certificate and response pointers are valid only for the duration of this call

## Parameters

<i>url</i>	the url of the request
<i>status</i>	the error status of the certificate
<i>certificate</i>	the certificate details. This pointer will be valid only for this call
<i>response</i>	object to signal whether to continue loading the URL. This pointer will be valid only for this call

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.4.2.7** `override void Coherent.UI.BrowserViewListener.OnCursorChanged ( CursorTypes cursor ) [inline], [virtual]`

Called when the cursor has changed internally in the

Parameters

<i>cursor</i>	the new cursor
---------------	----------------

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.4.2.8** `override void Coherent.UI.BrowserViewListener.OnDraw ( CoherentHandle handle, bool usesSharedMemory, int width, int height ) [inline], [virtual]`

Called when a new surface has been drawn and is ready to use by the client.

Parameters

<i>handle</i>	a handle to one of the buffers created by CreateSurface. May be a shared memory buffer or a shared texture depending on the way the <a href="#">View</a> was created. The handle is valid only during this call
<i>usesSharedMemory</i>	determines whether the handle parameter is a handle to shared memory or shared texture
<i>width</i>	the width of the surface
<i>height</i>	the height of the surface

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.4.2.9** `override void Coherent.UI.BrowserViewListener.OnError ( ViewError error ) [inline], [virtual]`

Called when an error occurs for this specific

Parameters

<i>error</i>	error description
--------------	-------------------

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.4.2.10** `override void Coherent.UI.BrowserViewListener.OnFailLoad ( int frameId, string validatedPath, bool isMainFrame, string error ) [inline], [virtual]`

Called when a frame has been failed loading.

Parameters

<i>frameId</i>	the id of the frame
<i>validatedPath</i>	the path in the frame
<i>isMainFrame</i>	true if this is the main frame of the <a href="#">View</a>
<i>error</i>	error message for the failure

Reimplemented from [Coherent.UI.ViewListenerBase](#).

8.4.2.11 `override void Coherent.UI.BrowserViewListener.OnFileSelectRequest ( FileSelectRequest request )`  
`[inline],[virtual]`

Called when the view requests file selection. It could be either single file, directory or multiple files.

Parameters

<i>request</i>	contains the file selection params for the request
----------------	--

Reimplemented from [Coherent.UI.ViewListenerBase](#).

8.4.2.12 `override void Coherent.UI.BrowserViewListener.OnFinishLoad ( int frameId, string validatedPath, bool isMainFrame, int statusCode, HTTPHeader[] headers )`  
`[inline],[virtual]`

Called when a frame has been successfully loaded.

Parameters

<i>frameId</i>	the id of the loaded frame
<i>validatedPath</i>	the path loaded in the frame
<i>isMainFrame</i>	true if this is the main frame of the <a href="#">View</a>
<i>statusCode</i>	the status code of the response
<i>headers</i>	an array of header fields
<i>headersCount</i>	the count of items in the headers array

Reimplemented from [Coherent.UI.ViewListenerBase](#).

8.4.2.13 `override void Coherent.UI.BrowserViewListener.OnGetAuthCredentials ( bool isProxy, string host, uint port, string realm, string scheme )`  
`[inline],[virtual]`

Called when a view requires authentication credentials.

Parameters

<i>isProxy</i>	whether the request came from a server or a proxy
<i>host</i>	the host which triggered the request
<i>port</i>	the port at which the request was triggered
<i>realm</i>	realm of the authentication challenge. Encoded in UTF-8
<i>scheme</i>	the authentication scheme used, e.g. "basic" or "digest". Encoded in ASCII

Reimplemented from [Coherent.UI.ViewListenerBase](#).

8.4.2.14 `override void Coherent.UI.BrowserViewListener.OnIMEShouldCancelComposition ( )`  
`[inline],[virtual]`

Called when the user must cancel the IME composition due to an event in the

Reimplemented from [Coherent.UI.ViewListenerBase](#).

8.4.2.15 `override void Coherent.UI.BrowserViewListener.OnJavaScriptMessage ( string message, string defaultPrompt, string frameUrl, int messageType )`  
`[inline],[virtual]`

Called when the view triggered a javascript message box, i.e. an alert, confirmation dialog or a prompt dialog.

Parameters

<i>message</i>	the JavaScript message
<i>defaultPrompt</i>	the default value of the prompt text box, in case the message type is prompt
<i>frameUrl</i>	the URL which created the message
<i>messageType</i>	the type of the message (alert/confirm/prompt)

Reimplemented from [Coherent.UI.ViewListenerBase](#).

#### 8.4.2.16 `override void Coherent.UI.BrowserViewListener.OnNavigateTo ( string path ) [inline],[virtual]`

Called when the view starts navigation to a new path.

Parameters

<i>path</i>	URL that the view is navigating to
-------------	------------------------------------

Reimplemented from [Coherent.UI.ViewListenerBase](#).

#### 8.4.2.17 `override void Coherent.UI.BrowserViewListener.OnReadyForBindings ( int frameId, string path, bool isMainFrame ) [inline],[virtual]`

Called when a frame is ready for bindings.

Parameters

<i>frameId</i>	the id of the frame
<i>path</i>	the path in the frame
<i>isMainFrame</i>	true if this is the main frame of the view

Reimplemented from [Coherent.UI.ViewListenerBase](#).

#### 8.4.2.18 `override void Coherent.UI.BrowserViewListener.OnRequestMediaStream ( MediaStreamRequest request ) [inline],[virtual]`

Called when the view requests access to a media stream. Media streams are the audio capture (microphone) and video capture (camera) devices on the system.

Parameters

<i>request</i>	contains the available media streams for the request
----------------	--

Reimplemented from [Coherent.UI.ViewListenerBase](#).

#### 8.4.2.19 `override void Coherent.UI.BrowserViewListener.OnScriptMessage ( ViewListenerBase.MessageLevel level, string message, string sourceId, int line ) [inline],[virtual]`

Called when a message is sent from a script running in this specific

Parameters

<i>level</i>	message level
<i>message</i>	the text of the message
<i>sourceId</i>	id of the script (usually file name)
<i>line</i>	the number of the line in which the message was sent

Reimplemented from [Coherent.UI.ViewListenerBase](#).

#### 8.4.2.20 `override void Coherent.UI.BrowserViewListener.OnStartLoading ( ) [inline],[virtual]`

Called when a new path has started loading.

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.4.2.21** `override void Coherent.UI.BrowserViewListener.OnStopLoading ( ) [inline],[virtual]`

Called when all load operations have completed.

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.4.2.22** `override void Coherent.UI.BrowserViewListener.OnTextInputTypeChanged ( TextInputType type, bool canComposeInline ) [inline],[virtual]`

Called when the current text input control changes (i.e. the user click an edit-box). Use this method to decide when to allow for IME input. the method will be called ONLY if IME is activated on this [View](#). [View::IMEActivate](#)

Parameters

<i>type</i>	the type of the currently focused text input control by the user
<i>canCompose-Inline</i>	if the IME composition could be performed in-line in the control

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.4.2.23** `override void Coherent.UI.BrowserViewListener.OnURLRequest ( string url, URLResponse response ) [inline],[virtual]`

Called before an URL request is made. The default implementation allows all requests.

Parameters

<i>url</i>	the request URL
<i>response</i>	response whether to allow the request and may be redirect it. This pointer will be valid only for this call

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.4.2.24** `override void Coherent.UI.BrowserViewListener.OnViewCreated ( View view ) [inline],[virtual]`

Called when the requested

Parameters

<i>view</i>	the instance of the view containing all manipulation methods
-------------	--

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.4.2.25** `override void Coherent.UI.BrowserViewListener.Release ( ) [inline],[virtual]`

Called when the listener is no longer needed by the [UI](#) system.

Reimplemented from [Coherent.UI.ViewListenerBase](#).

### 8.4.3 Event Documentation

**8.4.3.1** `CoherentUI_OnBindingsReleased` [Coherent.UI.BrowserViewListener.BindingsReleased](#)

Fired when the bindings for frame are released.

**8.4.3.2** `CoherentUI_OnCallback` [Coherent.UI.BrowserViewListener.Callback](#)

Fired by the [UI](#) when there is no registered handler for this event.

**8.4.3.3 CoherentUI\_OnCaretRectChanged Coherent.UI.BrowserViewListener.CaretRectChanged**

Fired when the caret changes during IME composition. You can use this method to correctly position a custom IME control & candidate list.

**8.4.3.4 CoherentUI\_OnCertificateError Coherent.UI.BrowserViewListener.CertificateError**

Fired when there is an error with the certificate of a particular URL. the certificate and response pointers are valid only for the duration of this call

**8.4.3.5 CoherentUI\_OnCursorChanged Coherent.UI.BrowserViewListener.CursorChanged**

Fired when the cursor has changed internally in the

**8.4.3.6 CoherentUI\_OnDraw Coherent.UI.BrowserViewListener.Draw**

Fired when a new surface has been drawn and is ready to use by the client.

**8.4.3.7 CoherentUI\_OnError Coherent.UI.BrowserViewListener.Error**

Fired when an error occurs for this specific

**8.4.3.8 CoherentUI\_OnFailLoad Coherent.UI.BrowserViewListener.FailLoad**

Fired when a frame has been failed loading.

**8.4.3.9 CoherentUI\_OnFileSelectRequest Coherent.UI.BrowserViewListener.FileSelectRequest**

Fired when the view requests file selection. It could be either single file, directory or multiple files.

**8.4.3.10 CoherentUI\_OnFinishLoad Coherent.UI.BrowserViewListener.FinishLoad**

Fired when a frame has been successfully loaded.

**8.4.3.11 CoherentUI\_OnGetAuthCredentials Coherent.UI.BrowserViewListener.GetAuthCredentials**

Fired when a view requires authentication credentials.

**8.4.3.12 CoherentUI\_OnIMEShouldCancelComposition Coherent.UI.BrowserViewListener.IMEShouldCancelComposition**

Fired when the user must cancel the IME composition due to an event in the

**8.4.3.13 CoherentUI\_OnJavaScriptMessage Coherent.UI.BrowserViewListener.JavaScriptMessage**

Fired when the view triggered a javascript message box, i.e. an alert, confirmation dialog or a prompt dialog.

**8.4.3.14 CoherentUI\_OnNavigateTo Coherent.UI.BrowserViewListener.NavigateTo**

Fired when the view starts navigation to a new path.



**8.4.3.15 CoherentUI\_OnReadyForBindings Coherent.UI.BrowserViewListener.ReadyForBindings**

Fired when a frame is ready for bindings.

**8.4.3.16 CoherentUI\_OnRequestMediaStream Coherent.UI.BrowserViewListener.RequestMediaStream**

Fired when the view requests access to a media stream. Media streams are the audio capture (microphone) and video capture (camera) devices on the system.

**8.4.3.17 CoherentUI\_OnScriptMessage Coherent.UI.BrowserViewListener.ScriptMessage**

Fired when a message is sent from a script running in this specific

**8.4.3.18 CoherentUI\_OnStartLoading Coherent.UI.BrowserViewListener.StartLoading**

Fired when a new path has started loading.

**8.4.3.19 CoherentUI\_OnStopLoading Coherent.UI.BrowserViewListener.StopLoading**

Fired when all load operations have completed.

**8.4.3.20 CoherentUI\_OnTextInputTypeChanged Coherent.UI.BrowserViewListener.TextInputTypeChanged**

Fired when the current text input control changes (i.e. the user click an edit-box). Use this method to decide when to allow for IME input. the method will be called ONLY if IME is activated on this [View](#). [View::IMEActivate](#)

**8.4.3.21 CoherentUI\_OnURLRequest Coherent.UI.BrowserViewListener.URLRequest**

Fired before an URL request is made. The default implementation allows all requests.

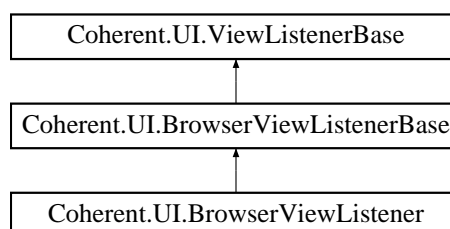
**8.4.3.22 CoherentUI\_OnViewCreated Coherent.UI.BrowserViewListener.ViewCreated**

Fired when the requested

## 8.5 Coherent.UI.BrowserViewListenerBase Class Reference

Interface all browser view listeners inherit. For an easier to use interface inherit instead -. [Coherent::UI::BrowserViewListener](#)

Inheritance diagram for Coherent.UI.BrowserViewListenerBase:



## Additional Inherited Members

### 8.5.1 Detailed Description

Interface all browser view listeners inherit. For an easier to use interface inherit instead -. [Coherent::UI::BrowserViewListener](#)

## 8.6 Coherent.UI.Binding.CallbackArguments Class Reference

Holds the arguments for the generic callback not handled by a registered delegate

### Public Attributes

- [Value\[\] Arguments](#)  
*Arguments provided by JavaScript*

### 8.6.1 Detailed Description

Holds the arguments for the generic callback not handled by a registered delegate

### 8.6.2 Member Data Documentation

#### 8.6.2.1 Value [] Coherent.UI.Binding.CallbackArguments.Arguments

Arguments provided by JavaScript

## 8.7 Coherent.UI.Certificate Class Reference

Represents an SSL certificate.

Inherits IDisposable.

### Properties

- `byte[]` [SerialNumber](#) [get]  
*Serial number of the certificate in DER encoding.*
- `byte[]` [Fingerprint](#) [get]  
*the fingerprint of the certificate*
- `byte[]` [FingerprintCA](#) [get]  
*the fingerprint of the certificate authority*
- [CertificatePrincipal Subject](#) [get]  
*Subject of the certificate.*
- [CertificatePrincipal Issuer](#) [get]  
*Issuer of the certificate.*
- `uint` [ValidStart](#) [get]  
*The starting date from which the certificate becomes valid in seconds since the UNIX epoch 01.01.1970.*
- `uint` [ValidEnd](#) [get]  
*The date after which the certificate is no longer valid in seconds since the UNIX epoch 01.01.1970.*

### 8.7.1 Detailed Description

Represents an SSL certificate.

### 8.7.2 Property Documentation

**8.7.2.1** `byte [] Coherent.UI.Certificate.Fingerprint` `[get]`

the fingerprint of the certificate

the fingerprint in binary as 20 bytes

**8.7.2.2** `byte [] Coherent.UI.Certificate.FingerprintCA` `[get]`

the fingerprint of the certificate authority

the fingerprint in binary as 20 bytes

**8.7.2.3** **CertificatePrincipal** `Coherent.UI.Certificate.Issuer` `[get]`

Issuer of the certificate.

**8.7.2.4** `byte [] Coherent.UI.Certificate.SerialNumber` `[get]`

Serial number of the certificate in DER encoding.

**8.7.2.5** **CertificatePrincipal** `Coherent.UI.Certificate.Subject` `[get]`

Subject of the certificate.

**8.7.2.6** `uint Coherent.UI.Certificate.ValidEnd` `[get]`

The date after which the certificate is no longer valid in seconds since the UNIX epoch 01.01.1970.

**8.7.2.7** `uint Coherent.UI.Certificate.ValidStart` `[get]`

The starting date from which the certificate becomes valid in seconds since the UNIX epoch 01.01.1970.

## 8.8 Coherent.UI.CertificatePrincipal Class Reference

Represents certificate principal.

Inherits IDisposable.

### Properties

- `string[] StreetAddresses` `[get]`  
*The list of street addresses of the principal*
- `string[] OrganizationNames` `[get]`  
*The list of organization names of the principal*
- `string[] OrganizationUnitNames` `[get]`

- The list of organization unit names of the principal*
- string[] [DomainComponents](#) [get]
- The list of domain components storage of the principal*
- string [CommonName](#) [get]
- Null terminated string with the common name of the subject.*
- string [LocalityName](#) [get]
- Null terminated string with the locality name of the subject.*
- string [StateOrProvinceName](#) [get]
- Null terminated string with the state or province name of the subject.*
- string [CountryName](#) [get]
- Null terminated string with the country name of the subject.*

### 8.8.1 Detailed Description

Represents certificate principal.

### 8.8.2 Property Documentation

8.8.2.1 string [Coherent.UI.CertificatePrincipal.CommonName](#) [get]

Null terminated string with the common name of the subject.

8.8.2.2 string [Coherent.UI.CertificatePrincipal.CountryName](#) [get]

Null terminated string with the country name of the subject.

8.8.2.3 string [] [Coherent.UI.CertificatePrincipal.DomainComponents](#) [get]

The list of domain components storage of the principal

8.8.2.4 string [Coherent.UI.CertificatePrincipal.LocalityName](#) [get]

Null terminated string with the locality name of the subject.

8.8.2.5 string [] [Coherent.UI.CertificatePrincipal.OrganizationNames](#) [get]

The list of organization names of the principal

8.8.2.6 string [] [Coherent.UI.CertificatePrincipal.OrganizationUnitNames](#) [get]

The list of organization unit names of the principal

8.8.2.7 string [Coherent.UI.CertificatePrincipal.StateOrProvinceName](#) [get]

Null terminated string with the state or province name of the subject.

8.8.2.8 string [] [Coherent.UI.CertificatePrincipal.StreetAddresses](#) [get]

The list of street addresses of the principal

## 8.9 Coherent.UI.ChildViewInfo Class Reference

Helper structure for creating child views.

Inherits IDisposable.

### Properties

- [ViewInfo NewViewInfo](#) [get, set]  
*An existing*
- [ViewListenerBase Listener](#) [get, set]  
*View*

### 8.9.1 Detailed Description

Helper structure for creating child views.

### 8.9.2 Property Documentation

**8.9.2.1 ViewListenerBase Coherent.UI.ChildViewInfo.Listener** [get], [set]

*View*

**8.9.2.2 ViewInfo Coherent.UI.ChildViewInfo.NewViewInfo** [get], [set]

*An existing*

## 8.10 Coherent.UI.Binding.CoherentProperty Class Reference

Specify a property / field or method visible to [Coherent UI](#)

Inherits Attribute.

### Public Member Functions

- [CoherentProperty](#) ()  
*Make a property / field or method visible to [Coherent UI](#)*
- [CoherentProperty](#) (string name)  
*Make a property / field or method visible to [Coherent UI](#)*

### 8.10.1 Detailed Description

Specify a property / field or method visible to [Coherent UI](#)

### 8.10.2 Constructor & Destructor Documentation

**8.10.2.1 Coherent.UI.Binding.CoherentProperty.CoherentProperty ( )** [inline]

Make a property / field or method visible to [Coherent UI](#)

8.10.2.2 Coherent.UI.Binding.CoherentProperty.CoherentProperty ( *string name* ) [inline]

Make a property / field or method visible to [Coherent UI](#)

## Parameters

<i>name</i>	name of the property / field / method
-------------	---------------------------------------

## 8.11 Coherent.UI.Binding.CoherentType Class Reference

Specify which properties and fields of a type are visible to [Coherent UI](#)

Inherits Attribute.

### 8.11.1 Detailed Description

Specify which properties and fields of a type are visible to [Coherent UI](#)

## 8.12 Coherent.UI.Download Class Reference

Encapsulates a download task.

Inherits IDisposable.

### Public Member Functions

- virtual void [Destroy](#) ()  
*Must be called when the client is ready with the download to free the resources associated with it If the download is not yet complete this implies a Cancel operation Calling any method after Finish is undefined.*
- virtual uint [GetId](#) ()  
*An opaque id of the download. Ids are unique per-system per-session and can be used to distinguish downloads pointing to the same resource.*
- virtual bool [Cancel](#) ()  
*Cancels a download task.*
- virtual string [GetURL](#) ()  
*The download url.*
- virtual string [GetFilename](#) ()  
*The name of the downloaded file.*
- virtual string [GetReferrer](#) ()  
*The referrer of the download.*
- virtual uint [GetTotalBytes](#) ()  
*The estimated size of the download.*
- virtual uint [GetDownloadedBytesSoFar](#) ()  
*The bytes downloaded so far. NB. Downloads are completed asynchronously, this is an estimation and might not be up-to-date.*
- virtual bool [IsComplete](#) ()  
*Tells if the download is complete.*
- virtual bool [IsCancelled](#) ()  
*Tells if the download has been cancelled.*
- virtual bool [HasFailed](#) ()  
*Tells if the download has failed.*

### 8.12.1 Detailed Description

Encapsulates a download task.

## 8.12.2 Member Function Documentation

### 8.12.2.1 `virtual bool Coherent.UI.Download.Cancel ( ) [inline],[virtual]`

Cancels a download task.

#### Returns

true if the task has been cancelled. Calling Cancel on an already finished or stopped task will return false

### 8.12.2.2 `virtual void Coherent.UI.Download.Destroy ( ) [inline],[virtual]`

Must be called when the client is ready with the download to free the resources associated with it If the download is not yet complete this implies a Cancel operation Calling any method after Finish is undefined.

### 8.12.2.3 `virtual uint Coherent.UI.Download.GetDownloadedBytesSoFar ( ) [inline],[virtual]`

The bytes downloaded so far. NB. Downloads are completed asynchronously, this is an estimation and might not be up-to-date.

#### Returns

the bytes downloaded so far

### 8.12.2.4 `virtual string Coherent.UI.Download.GetFilename ( ) [inline],[virtual]`

The name of the downloaded file.

#### Returns

the name of the file being downloaded

### 8.12.2.5 `virtual uint Coherent.UI.Download.GetId ( ) [inline],[virtual]`

An opaque id of the download. Ids are unique per-system per-session and can be used to distinguish downloads pointing to the same resource.

#### Returns

a unique id of the download

### 8.12.2.6 `virtual string Coherent.UI.Download.GetReferrer ( ) [inline],[virtual]`

The referrer of the download.

#### Returns

the referrer

### 8.12.2.7 `virtual uint Coherent.UI.Download.GetTotalBytes ( ) [inline],[virtual]`

The estimated size of the download.

#### Returns

the estimated size in bytes



8.12.2.8 `virtual string Coherent.UI.Download.GetURL ( ) [inline],[virtual]`

The download url.

Returns

the url to download

8.12.2.9 `virtual bool Coherent.UI.Download.HasFailed ( ) [inline],[virtual]`

Tells if the download has failed.

Returns

if the download has failed

8.12.2.10 `virtual bool Coherent.UI.Download.IsCancelled ( ) [inline],[virtual]`

Tells if the download has been cancelled.

Returns

if the download has been cancelled

8.12.2.11 `virtual bool Coherent.UI.Download.IsComplete ( ) [inline],[virtual]`

Tells if the download is complete.

Returns

if the download is complete

## 8.13 Coherent.UI.EventListener Class Reference

Abstract interface to listen to System-related events.

Inherits IDisposable.

### Public Member Functions

- virtual void [SystemReady](#) ()  
*Called when the system is fully initialized. Creating Views is now permitted.*
- virtual void [OnError](#) ([SystemError](#) arg0)  
*Called when an error in the system occurs.*
- virtual bool [OnDownloadStarted](#) ([Download](#) downloadItem)  
*Called when the system tries to start a new download. Return true to allow for the download to begin or false to cancel it.*
- virtual void [OnDownloadComplete](#) ([Download](#) downloadItem, System.IntPtr data, uint size)  
*Called when a download is complete.*
- virtual void [OnDownloadFailed](#) ([Download](#) downloadItem, [DownloadErrorType](#) error)  
*Called when a download has failed.*

### 8.13.1 Detailed Description

Abstract interface to listen to System-related events.

### 8.13.2 Member Function Documentation

**8.13.2.1** `virtual void Coherent.UI.EventListener.OnDownloadComplete ( Download downloadItem, System.IntPtr data, uint size ) [inline],[virtual]`

Called when a download is complete.

**8.13.2.2** `virtual void Coherent.UI.EventListener.OnDownloadFailed ( Download downloadItem, DownloadErrorType error ) [inline],[virtual]`

Called when a download has failed.

**8.13.2.3** `virtual bool Coherent.UI.EventListener.OnDownloadStarted ( Download downloadItem ) [inline],[virtual]`

Called when the system tries to start a new download. Return true to allow for the download to begin or false to cancel it.

**8.13.2.4** `virtual void Coherent.UI.EventListener.OnError ( SystemError arg0 ) [inline],[virtual]`

Called when an error in the system occurs.

**8.13.2.5** `virtual void Coherent.UI.EventListener.SystemReady ( ) [inline],[virtual]`

Called when the system is fully initialized. Creating Views is now permitted.

## 8.14 Coherent.UI.EventModifiersState Class Reference

The state of the key modifiers when an event happens.

Inherits IDisposable.

### 8.14.1 Detailed Description

The state of the key modifiers when an event happens.

## 8.15 Coherent.UI.EventMouseModifiersState Class Reference

The state of the mouse modifiers when an event happens.

Inherits IDisposable.

### 8.15.1 Detailed Description

The state of the mouse modifiers when an event happens.

## 8.16 Coherent.UI.FileHandler Class Reference

Abstract interface that allows clients to provide their own file-handling functionality.

Inherits IDisposable.

### Public Member Functions

- virtual void [ReadFile](#) (string url, [ResourceResponse](#) response)  
*Requests to read a resource.*
- virtual void [WriteFile](#) (string url, [ResourceData](#) resource)  
*Request to write to a resource.*

### 8.16.1 Detailed Description

Abstract interface that allows clients to provide their own file-handling functionality.

### 8.16.2 Member Function Documentation

**8.16.2.1** virtual void Coherent.UI.FileHandler.ReadFile ( string url, ResourceResponse response ) [inline],  
[virtual]

Requests to read a resource.

Parameters

<i>url</i>	the coui url of the resource.
<i>response</i>	the response to be filled with the resource data or notified for failure

**8.16.2.2** virtual void Coherent.UI.FileHandler.WriteFile ( string url, ResourceData resource ) [inline],[virtual]

Request to write to a resource.

Parameters

<i>url</i>	the coui url of the resource.
<i>resource</i>	the resource data that has to be written

## 8.17 Coherent.UI.HTTPHeader Class Reference

Encapsulates a HTTP header field with it's content.

Inherits IDisposable.

### Properties

- string [Field](#) [get, set]  
*The name (key) of the HTTP header field.*
- string [Content](#) [get, set]  
*The value of the HTTP header field.*

### 8.17.1 Detailed Description

Encapsulates a HTTP header field with it's content.

### 8.17.2 Property Documentation

8.17.2.1 `string Coherent.UI.HTTPHeader.Content` `[get]`, `[set]`

The value of the HTTP header field.

8.17.2.2 `string Coherent.UI.HTTPHeader.Field` `[get]`, `[set]`

The name (key) of the HTTP header field.

## 8.18 Coherent.UI.ILogHandler Class Reference

Interface to allow custom logging.

Inherits `IDisposable`.

### Public Member Functions

- virtual void [WriteLog](#) (Severity severity, string message, uint length)  
*Called when a log message has to be written.*
- virtual void [Assert](#) (string message)  
*Called when an assert is triggered.*

### 8.18.1 Detailed Description

Interface to allow custom logging.

### 8.18.2 Member Function Documentation

8.18.2.1 `virtual void Coherent.UI.ILogHandler.Assert ( string message )` `[inline]`, `[virtual]`

Called when an assert is triggered.

Parameters

<i>message</i>	a message that describes the reason for the assertion
----------------	---

8.18.2.2 `virtual void Coherent.UI.ILogHandler.WriteLog ( Severity severity, string message, uint length )` `[inline]`, `[virtual]`

Called when a log message has to be written.

Parameters

<i>severity</i>	the severity of the message
<i>message</i>	the log message itself
<i>length</i>	the length of the message

## 8.19 Coherent.UI.ImageData Class Reference

This class represents a link to a

Inherits IDisposable.

### Public Member Functions

- virtual void [Update](#) (IntPtr data, bool flipY)  
*Updates the content of the*
- virtual void [Destroy](#) ()  
*Destroys the*

### 8.19.1 Detailed Description

This class represents a link to a

### 8.19.2 Member Function Documentation

**8.19.2.1** virtual void Coherent.UI.ImageData.Destroy ( ) [inline],[virtual]

Destroys the

**8.19.2.2** virtual void Coherent.UI.ImageData.Update ( IntPtr data, bool flipY ) [inline],[virtual]

Updates the content of the

Parameters

<i>data</i>	a pointer to the new data to copy in the object the size of the data must equal to the total size of the <a href="#">ImageData</a>
<i>flipY</i>	tells if to flip vertically when copying

## 8.20 Coherent.UI.Binding.InvalidValueCastException Class Reference

Thrown when casting a [Value](#) to an incompatible type

Inherits InvalidCastException.

### 8.20.1 Detailed Description

Thrown when casting a [Value](#) to an incompatible type

## 8.21 Coherent.UI.KeyEventData Class Reference

A keyboard event.

Inherits IDisposable.

## Properties

- `int KeyCode` [get, set]  
*The key code.*
- `KeyEventData.EventType Type` [get, set]  
*The type of the event.*
- `EventModifiersState Modifiers` [get, set]  
*The current key modifiers.*
- `bool IsAutoRepeat` [get, set]  
*Is it an auto-repeat event (the user is holding the key down). This value isn't crucial and you can set it to false if you don't have the information, however you might experience incorrect repeat behavior in such case.*
- `bool IsNumPad` [get, set]  
*Is it a key from the num-pad.*

### 8.21.1 Detailed Description

A keyboard event.

### 8.21.2 Property Documentation

#### 8.21.2.1 `bool Coherent.UI.KeyEventData.IsAutoRepeat` [get], [set]

Is it an auto-repeat event (the user is holding the key down). This value isn't crucial and you can set it to false if you don't have the information, however you might experience incorrect repeat behavior in such case.

#### 8.21.2.2 `bool Coherent.UI.KeyEventData.IsNumPad` [get], [set]

Is it a key from the num-pad.

#### 8.21.2.3 `int Coherent.UI.KeyEventData.KeyCode` [get], [set]

The key code.

#### 8.21.2.4 `EventModifiersState Coherent.UI.KeyEventData.Modifiers` [get], [set]

The current key modifiers.

#### 8.21.2.5 `KeyEventData.EventType Coherent.UI.KeyEventData.Type` [get], [set]

The type of the event.

## 8.22 Coherent.UI.MediaStreamDevice Class Reference

Represents a media stream device.

Inherits `IDisposable`.

## Properties

- [MediaStreamType](#) [Type](#) [get]  
*type of the device*
- string [Deviceld](#) [get]  
*Null terminated string with the unique id of the device.*
- string [Name](#) [get]  
*Null terminated string with the device name. It may not be unique.*

### 8.22.1 Detailed Description

Represents a media stream device.

### 8.22.2 Property Documentation

#### 8.22.2.1 string Coherent.UI.MediaStreamDevice.Deviceld [get]

Null terminated string with the unique id of the device.

#### 8.22.2.2 string Coherent.UI.MediaStreamDevice.Name [get]

Null terminated string with the device name. It may not be unique.

#### 8.22.2.3 [MediaStreamType](#) Coherent.UI.MediaStreamDevice.Type [get]

type of the device

## 8.23 Coherent.UI.MediaStreamRequest Class Reference

Represents a request for media stream.

Inherits [IDisposable](#).

## Public Member Functions

- void [Respond](#) (uint[] devices)  
*Respond to the media stream request.*

## Properties

- [MediaStreamDevice](#)[] [Devices](#) [get]  
*Get the list of available devices for the request. The list contains only devices of the requested types.*

### 8.23.1 Detailed Description

Represents a request for media stream.

## 8.23.2 Member Function Documentation

8.23.2.1 `void Coherent.UI.MediaStreamRequest.Respond ( uint[] devices ) [inline]`

Respond to the media stream request.



## Parameters

<i>devices</i>	array of indices of the media stream devices to use for the request
----------------	---

Only a single device per stream type must be allowed. After a call to Respond calling any method on the instance results in undefined behavior. Must be called on the thread that owns the [Coherent::UI::UISystem](#) instance.

### 8.23.3 Property Documentation

#### 8.23.3.1 `MediaStreamDevice [] Coherent.UI.MediaStreamRequest.Devices` [get]

Get the list of available devices for the request. The list contains only devices of the requested types.

## 8.24 Coherent.UI.MouseEventData Class Reference

A mouse event.

Inherits `IDisposable`.

### Properties

- `MouseEventData.EventType` [Type](#) [get, set]  
*The type of the event.*
- `MouseEventData.MouseButton` [Button](#) [get, set]  
*The mouse button that triggered the event.*
- `int X` [get, set]  
*The X position of the mouse in pixels in the space of the view.*
- `int Y` [get, set]  
*The Y position of the mouse in pixels in the space of the view.*
- `float WheelX` [get, set]  
*Scroll Delta X in wheel ticks. Positive numbers mean scrolling left.*
- `float WheelY` [get, set]  
*Scroll Delta Y in wheel ticks. Positive numbers mean scrolling up.*
- `EventModifiersState Modifiers` [get, set]  
*Key modifiers.*
- `EventMouseModifiersState MouseModifiers` [get, set]  
*Mouse modifiers.*

### 8.24.1 Detailed Description

A mouse event.

### 8.24.2 Property Documentation

#### 8.24.2.1 `MouseEventData.MouseButton Coherent.UI.MouseEventData.Button` [get], [set]

The mouse button that triggered the event.

#### 8.24.2.2 `EventModifiersState Coherent.UI.MouseEventData.Modifiers` [get], [set]

Key modifiers.

#### 8.24.2.3 `EventMouseModifiersState` `Coherent.UI.MouseEventData.MouseModifiers` `[get]`, `[set]`

Mouse modifiers.

#### 8.24.2.4 `MouseEventData.EventType` `Coherent.UI.MouseEventData.Type` `[get]`, `[set]`

The type of the event.

#### 8.24.2.5 `float` `Coherent.UI.MouseEventData.WheelX` `[get]`, `[set]`

Scroll Delta X in wheel ticks. Positive numbers mean scrolling left.

#### 8.24.2.6 `float` `Coherent.UI.MouseEventData.WheelY` `[get]`, `[set]`

Scroll Delta Y in wheel ticks. Positive numbers mean scrolling up.

#### 8.24.2.7 `int` `Coherent.UI.MouseEventData.X` `[get]`, `[set]`

The X position of the mouse in pixels in the space of the view.

#### 8.24.2.8 `int` `Coherent.UI.MouseEventData.Y` `[get]`, `[set]`

The Y position of the mouse in pixels in the space of the view.

## 8.25 `Coherent.UI.ResourceData` Class Reference

Abstract interface providing data for storing resources.

Inherits `IDisposable`.

### Public Member Functions

- virtual `IntPtr` `GetBuffer` ()  
*Get the buffer with the resource data.*
- virtual `uint` `GetSize` ()  
*Get the size of the data.*
- virtual `void` `SignalSuccess` ()  
*Report that writing to resource was successful. This should be executed in the thread that all `Coherent UI` methods are executed. After this method is called, any calls to any method of the instance are undefined behavior.*
- virtual `void` `SignalFailure` ()  
*Report that resource writing has failed. This should be executed in the thread that all `Coherent UI` methods are executed. After this method is called, any calls to any method of the instance are undefined behavior.*

### 8.25.1 Detailed Description

Abstract interface providing data for storing resources.

## 8.25.2 Member Function Documentation

### 8.25.2.1 virtual IntPtr Coherent.UI.ResourceData.GetBuffer ( ) [inline],[virtual]

Get the buffer with the resource data.

#### Returns

pointer to the beginning of the data. May return nullptr when acquiring the buffer has failed

### 8.25.2.2 virtual uint Coherent.UI.ResourceData.GetSize ( ) [inline],[virtual]

Get the size of the data.

#### Returns

the size of the data

### 8.25.2.3 virtual void Coherent.UI.ResourceData.SignalFailure ( ) [inline],[virtual]

Report that resource writing has failed. This should be executed in the thread that all [Coherent UI](#) methods are executed. After this method is called, any calls to any method of the instance are undefined behavior.

### 8.25.2.4 virtual void Coherent.UI.ResourceData.SignalSuccess ( ) [inline],[virtual]

Report that writing to resource was successful. This should be executed in the thread that all [Coherent UI](#) methods are executed. After this method is called, any calls to any method of the instance are undefined behavior.

## 8.26 Coherent.UI.ResourceResponse Class Reference

Abstract interface for responding to read resource requests.

Inherits IDisposable.

### Public Member Functions

- virtual IntPtr [GetBuffer](#) (uint size)  
*Request buffer for resource data. GetBuffer must be called only once. GetBuffer may be called in any thread.*
- virtual void [SignalSuccess](#) ()  
*Report that the resource is read successfully. This should be executed in the thread that all [Coherent UI](#) methods are executed. After this method is called, any calls to any method of the instance are undefined behavior.*
- virtual void [SignalFailure](#) ()  
*Report that resource reading has failed. This should be executed in the thread that all [Coherent UI](#) methods are executed. After this method is called, any calls to any method of the instance are undefined behavior.*

### 8.26.1 Detailed Description

Abstract interface for responding to read resource requests.

## 8.26.2 Member Function Documentation

8.26.2.1 `virtual IntPtr Coherent.UI.ResourceResponse.GetBuffer ( uint size ) [inline],[virtual]`

Request buffer for resource data. GetBuffer must be called only once. GetBuffer may be called in any thread.

## Parameters

<i>size</i>	the size of the resource
-------------	--------------------------

## Returns

pointer to the beginning of the buffer. It may return nullptr when allocating the buffer has failed

**8.26.2.2** `virtual void Coherent.UI.ResourceResponse.SignalFailure ( ) [inline],[virtual]`

Report that resource reading has failed. This should be executed in the thread that all [Coherent UI](#) methods are executed. After this method is called, any calls to any method of the instance are undefined behavior.

**8.26.2.3** `virtual void Coherent.UI.ResourceResponse.SignalSuccess ( ) [inline],[virtual]`

Report that the resource is read successfully. This should be executed in the thread that all [Coherent UI](#) methods are executed. After this method is called, any calls to any method of the instance are undefined behavior.

## 8.27 Coherent.UI.SurfaceResponse Class Reference

Interface that signals for creation of rendering surfaces.

Inherits [IDisposable](#).

### Public Member Functions

- `virtual void Signal (CoherentHandle handle)`

*Signals that the operation was completed. You should call this function in the same thread that updates the [UI](#) system.*

### 8.27.1 Detailed Description

Interface that signals for creation of rendering surfaces.

### 8.27.2 Member Function Documentation

**8.27.2.1** `virtual void Coherent.UI.SurfaceResponse.Signal ( CoherentHandle handle ) [inline],[virtual]`

Signals that the operation was completed. You should call this function in the same thread that updates the [UI](#) system.

## Parameters

<i>handle</i>	the created surface handle or a NULL handle if the operation fails
---------------	--

## 8.28 Coherent.UI.SystemError Class Reference

Encapsulates a system-related error.

Inherits [IDisposable](#).

## Properties

- [SystemErrorType ErrorCode](#) [get, set]  
*Indicates an error code.*
- string [Error](#) [get, set]  
*Error description.*

### 8.28.1 Detailed Description

Encapsulates a system-related error.

### 8.28.2 Property Documentation

#### 8.28.2.1 string Coherent.UI.SystemError.Error [get], [set]

Error description.

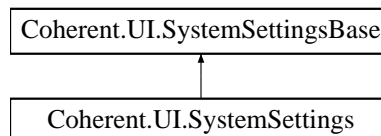
#### 8.28.2.2 SystemErrorType Coherent.UI.SystemError.ErrorCode [get], [set]

Indicates an error code.

## 8.29 Coherent.UI.SystemSettings Class Reference

Encapsulates the settings of the

Inheritance diagram for Coherent.UI.SystemSettings:



## Properties

- new string [HostDirectory](#) [get, set]  
*Points to the folder where the host application and its libraries reside (relative to the application's directory)*
- new string [CookiesResource](#) [get, set]  
*Resource name to be used for storing and loading cookies. Leave nullptr to disable persistence of cookies.*
- new string [CachePath](#) [get, set]  
*Path (relative to the host executable) where to save cached data. Leave nullptr for in-memory only caching.*
- new string [HTML5LocalStoragePath](#) [get, set]  
*Path (relative to the host executable) where to allow pages to save their HTML5 local data. Leave nullptr to forbid local storage.*
- new string [MinidumpFolder](#) [get, set]  
*The folder where eventual minidumps will be saved (Windows only)*
- new string [CustomUserAgentString](#) [get, set]  
*The user agent string that will be used if you select "UAT\_Custom" for UserAgent.*
- System.IntPtr [FocusWindow](#) [set]

*The focus window that the client has passed when creating her device. This, along with D3DInterfaceType and D3D-Object, is needed for fullscreen apps that use DirectX9 or DirectX9Ex so as to support Alt+Tab functionality. Otherwise might be omitted. (Windows only)*

- System.IntPtr [D3DObject](#) [set]

*The D3DObject used by client to create her device. This, along with D3DInterfaceType and FocusWindow, is needed for fullscreen apps that use DirectX9 or DirectX9Ex so as to support Alt+Tab functionality. Otherwise might be omitted. (Windows only)*

### 8.29.1 Detailed Description

Encapsulates the settings of the

### 8.29.2 Property Documentation

#### 8.29.2.1 new string Coherent.UI.SystemSettings.CachePath [get], [set]

Path (relative to the host executable) where to save cached data. Leave nullptr for in-memory only caching.

#### 8.29.2.2 new string Coherent.UI.SystemSettings.CookiesResource [get], [set]

Resource name to be used for storing and loading cookies. Leave nullptr to disable persistence of cookies.

#### 8.29.2.3 new string Coherent.UI.SystemSettings.CustomUserAgentString [get], [set]

The user agent string that will be used if you select "UAT\_Custom" for UserAgent.

#### 8.29.2.4 System.IntPtr Coherent.UI.SystemSettings.D3DObject [set]

The D3DObject used by client to create her device. This, along with D3DInterfaceType and FocusWindow, is needed for fullscreen apps that use DirectX9 or DirectX9Ex so as to support Alt+Tab functionality. Otherwise might be omitted. (Windows only)

#### 8.29.2.5 System.IntPtr Coherent.UI.SystemSettings.FocusWindow [set]

The focus window that the client has passed when creating her device. This, along with D3DInterfaceType and D3DObject, is needed for fullscreen apps that use DirectX9 or DirectX9Ex so as to support Alt+Tab functionality. Otherwise might be omitted. (Windows only)

#### 8.29.2.6 new string Coherent.UI.SystemSettings.HostDirectory [get], [set]

Points to the folder where the host application and it's libraries reside (relative to the application's directory)

#### 8.29.2.7 new string Coherent.UI.SystemSettings.HTML5LocalStoragePath [get], [set]

Path (relative to the host executable) where to allow pages to save their HTML5 local data. Leave nullptr to forbid local storage.

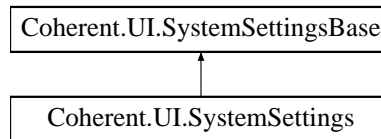
#### 8.29.2.8 new string Coherent.UI.SystemSettings.MinidumpFolder [get], [set]

The folder where eventual minidumps will be saved (Windows only)

## 8.30 Coherent.UI.SystemSettingsBase Class Reference

Encapsulates the settings of the

Inheritance diagram for Coherent.UI.SystemSettingsBase:



### Properties

- bool [EnableProxy](#) [get, set]  
*Enables/disables proxy support. Proxy settings are auto-detected when proxy support is enabled. Enable proxy support only when the user really has a proxy, because otherwise the auto-detection is very slow.*
- bool [AllowCookies](#) [get, set]  
*Enables/disables cookies when accessing the web.*
- bool [ForceDisablePluginFullscreen](#) [get, set]  
*Disables 'Fullscreen mode' on all Plugins(Flash etc.)*
- bool [DisableWebSecurity](#) [get, set]  
*Disable Same-Origin Policy. Use with caution.*
- int [DebuggerPort](#) [get, set]  
*Port to use for remote debugging. Use -1 to disable debugging.*
- bool [WriteMinidumps](#) [get, set]  
*If a*
- bool [EnableSupportForProprietaryCodecs](#) [get, set]  
*Enable support for proprietary codecs. Playing proprietary codecs requires ffmpegsumo library that is capable of playing them as well. Note that distributing such ffmpegsumo library requires license from the respective rights holder.*
- [Direct3DInterfaceType D3DInterfaceType](#) [get, set]  
*The type of D3DObject supplied. This, along with D3DObject and FocusWindow, is needed for fullscreen apps that use DirectX9 or DirectX9Ex so as to support Alt+Tab functionality. Otherwise might be omitted. (Windows only)*
- [UserAgentType UserAgent](#) [get, set]  
*The type of the user agent. If you select "UAT\_Custom", you MUST also specify the user agent string - defaults to Chrome.*
- string [CustomUserAgentString](#) [get, set]  
*The user agent string that will be used if you select "UAT\_Custom" for UserAgent.*

### 8.30.1 Detailed Description

Encapsulates the settings of the

### 8.30.2 Property Documentation

#### 8.30.2.1 bool Coherent.UI.SystemSettingsBase.AllowCookies [get], [set]

Enables/disables cookies when accessing the web.



**8.30.2.2** `string Coherent.UI.SystemSettingsBase.CustomUserAgentString` `[get], [set]`

The user agent string that will be used if you select "UAT\_Custom" for UserAgent.

**8.30.2.3** `Direct3DInterfaceType Coherent.UI.SystemSettingsBase.D3DInterfaceType` `[get], [set]`

The type of D3DObject supplied. This, along with D3DObject and FocusWindow, is needed for fullscreen apps that use DirectX9 or DirectX9Ex so as to support Alt+Tab functionality. Otherwise might be omitted. (Windows only)

**8.30.2.4** `int Coherent.UI.SystemSettingsBase.DebuggerPort` `[get], [set]`

Port to use for remote debugging. Use -1 to disable debugging.

**8.30.2.5** `bool Coherent.UI.SystemSettingsBase.DisableWebSecurity` `[get], [set]`

Disable Same-Origin Policy. Use with caution.

**8.30.2.6** `bool Coherent.UI.SystemSettingsBase.EnableProxy` `[get], [set]`

Enables/disables proxy support. Proxy settings are auto-detected when proxy support is enabled. Enable proxy support only when the user really has a proxy, because otherwise the auto-detection is very slow.

**8.30.2.7** `bool Coherent.UI.SystemSettingsBase.EnableSupportForProprietaryCodecs` `[get], [set]`

Enable support for proprietary codecs. Playing proprietary codecs requires ffmpegsumo library that is capable of playing them as well. Note that distributing such ffmpegsumo library requires license from the respective rights holder.

**8.30.2.8** `bool Coherent.UI.SystemSettingsBase.ForceDisablePluginFullscreen` `[get], [set]`

Disables 'Fullscreen mode' on all Plugins(Flash etc.)

**8.30.2.9** `UserAgentType Coherent.UI.SystemSettingsBase.UserAgent` `[get], [set]`

The type of the user agent. If you select "UAT\_Custom", you MUST also specify the user agent string - defaults to Chrome.

**8.30.2.10** `bool Coherent.UI.SystemSettingsBase.WriteMinidumps` `[get], [set]`

If a

## 8.31 Coherent.UI.TouchEventData Class Reference

A touch event.

Inherits IDisposable.

## Properties

- `int X` `[get, set]`  
*The X position where the touch happened in pixels in the space of the view.*
- `int Y` `[get, set]`  
*The Y position where the touch happened in pixels in the space of the view.*
- `uint Id` `[get, set]`  
*Identifies a particular touch input. This value must stay consistent in a touch contact sequence.*
- `TouchEventData.EventType Type` `[get, set]`  
*The type of the touch event.*
- `bool HasRadiusInformation` `[get, set]`  
*Flags if there is contact radius information for this event.*
- `uint RadiusX` `[get, set]`  
*The X radius of the contact (must be set only if HasRadiusInformation == true)*
- `uint RadiusY` `[get, set]`  
*The Y radius of the contact (must be set only if HasRadiusInformation == true)*

### 8.31.1 Detailed Description

A touch event.

### 8.31.2 Property Documentation

**8.31.2.1** `bool Coherent.UI.TouchEventdata.HasRadiusInformation` `[get], [set]`

Flags if there is contact radius information for this event.

**8.31.2.2** `uint Coherent.UI.TouchEventdata.Id` `[get], [set]`

Identifies a particular touch input. This value must stay consistent in a touch contact sequence.

**8.31.2.3** `uint Coherent.UI.TouchEventdata.RadiusX` `[get], [set]`

The X radius of the contact (must be set only if HasRadiusInformation == true)

**8.31.2.4** `uint Coherent.UI.TouchEventdata.RadiusY` `[get], [set]`

The Y radius of the contact (must be set only if HasRadiusInformation == true)

**8.31.2.5** `TouchEventData.EventType Coherent.UI.TouchEventdata.Type` `[get], [set]`

The type of the touch event.

**8.31.2.6** `int Coherent.UI.TouchEventdata.X` `[get], [set]`

The X position where the touch happened in pixels in the space of the view.

**8.31.2.7** `int Coherent.UI.TouchEventdata.Y` `[get], [set]`

The Y position where the touch happened in pixels in the space of the view.

## 8.32 Coherent.UI.UISystem Class Reference

Encapsulates basic [UI](#) system functionality.

Inherits [IDisposable](#).

### Public Member Functions

- virtual bool [Uninitialize](#) ()  
*Frees all resources used by the event system. the [UISystem](#) object is unusable after this call*
- virtual void [Update](#) ()  
*Communicates with the [UI](#) system process and performs all updates. Must be called in the thread that owns the system. All registered callbacks will be called in here*
- virtual void [FetchSurfaces](#) ()  
*Check if new surfaces have been drawn and are available. This function can be called from a thread different than the main [UI](#) system thread in order to support client applications with multi-threaded rendering. Fetches only the surfaces of buffered views; on-demand views must be fetched manually per-view*
- virtual void [CreateView](#) ([ViewInfo](#) info, string path, [ViewListenerBase](#) listener)  
*Creates a new [UI](#) Must be called after the system has signaled that it is ready via [Coherent::UI::EventListener::SystemReady](#)*
- virtual void [CreateView](#) ([ViewInfo](#) info, string path, [ViewListenerBase](#) listener, string scriptToExecute)  
*Creates a new [UI](#) Must be called after the system has signaled that it is ready via [Coherent::UI::EventListener::SystemReady](#)*

### 8.32.1 Detailed Description

Encapsulates basic [UI](#) system functionality.

### 8.32.2 Member Function Documentation

**8.32.2.1** virtual void [Coherent.UI.UISystem.CreateView](#) ( [ViewInfo](#) info, string path, [ViewListenerBase](#) listener )  
[inline],[virtual]

Creates a new [UI](#) Must be called after the system has signaled that it is ready via [Coherent::UI::EventListener::SystemReady](#)

#### Parameters

<i>info</i>	an instance of the <a href="#">ViewInfo</a> struct that describes all the parameters of the view
<i>path</i>	the page to load when the view is created
<i>listener</i>	an instance of the <a href="#">ViewListenerBase</a> interface to allow listening to all the events raised for this view

**8.32.2.2** virtual void [Coherent.UI.UISystem.CreateView](#) ( [ViewInfo](#) info, string path, [ViewListenerBase](#) listener, string scriptToExecute ) [inline],[virtual]

Creates a new [UI](#) Must be called after the system has signaled that it is ready via [Coherent::UI::EventListener::SystemReady](#)

#### Parameters

<i>info</i>	an instance of the <a href="#">ViewInfo</a> struct that describes all the parameters of the view
<i>path</i>	the page to load when the view is created
<i>listener</i>	an instance of the <a href="#">ViewListenerBase</a> interface to allow listening to all the events raised for this view

#### 8.32.2.3 virtual void Coherent.UI.UISystem.FetchSurfaces ( ) [inline],[virtual]

Check if new surfaces have been drawn and are available. This function can be called from a thread different than the main [UI](#) system thread in order to support client applications with multi-threaded rendering. Fetches only the surfaces of buffered views; on-demand views must be fetched manually per-view

#### 8.32.2.4 virtual bool Coherent.UI.UISystem.Uninitialize ( ) [inline],[virtual]

Frees all resources used by the event system. the [UISystem](#) object is unusable after this call

##### Returns

true if the system has been properly uninitialized

#### 8.32.2.5 virtual void Coherent.UI.UISystem.Update ( ) [inline],[virtual]

Communicates with the [UI](#) system process and performs all updates. Must be called in the thread that owns the system. All registered callbacks will be called in here

### 8.33 Coherent.UI.Binding.UnsupportedPrimitiveTypeException Class Reference

Thrown when trying to bind a value of unsupported primitive type such as long

Inherits Exception.

#### 8.33.1 Detailed Description

Thrown when trying to bind a value of unsupported primitive type such as long

### 8.34 Coherent.UI.URLRequest Class Reference

represents a single URL request Allows monitoring and modifications of URL requests

Inherits IDisposable.

#### Public Member Functions

- [HTTPHeader\[\]](#) [GetExtraHeaders](#) ()  
*Get extra request headers*
- void [SetExtraHeaders](#) ([HTTPHeader\[\]](#) headers)  
*Set the extra headers.*
- virtual string [GetURL](#) ()  
*Get the URL of the request.*
- virtual void [SetURL](#) (string url)  
*Replace the URL of the request.*
- virtual string [GetMethod](#) ()  
*Get the method of the request.*
- virtual void [SetMethod](#) (string method)  
*Set the method of the request.*
- virtual string [GetReferrer](#) ()

*Get the referrer of the request.*

- virtual void [SetReferrer](#) (string referrer)

*Set the referrer of the request.*

- virtual void [SetExtraHeader](#) (string field, string content)

*Set extra header for the request. If there is already a header with that field name its content will be replaced.*

- virtual void [AppendExtraHeaderValue](#) (string field, string content)

*Append extra header for the request. If there is already a header with that field name the content will be appended to the header current content with a comma as a separator between the values. If there is not such header it will be added.*

- virtual void [RemoveExtraHeader](#) (string field)

*Remove extra header from the request.*

- virtual string [GetExtraHeader](#) (string field)

*Get extra header value.*

- virtual void [Deny](#) ()

*Deny the request.*

### 8.34.1 Detailed Description

represents a single URL request Allows monitoring and modifications of URL requests

### 8.34.2 Member Function Documentation

**8.34.2.1** virtual void Coherent.UI.URLRequest.AppendExtraHeaderValue ( string *field*, string *content* ) [inline],  
[virtual]

Append extra header for the request. If there is already a header with that field name the content will be appended to the header current content with a comma as a separator between the values. If there is not such header it will be added.

Parameters

<i>field</i>	name of the header
<i>content</i>	content of the header

**8.34.2.2** virtual void Coherent.UI.URLRequest.Deny ( ) [inline],[virtual]

Deny the request.

**8.34.2.3** virtual string Coherent.UI.URLRequest.GetExtraHeader ( string *field* ) [inline],[virtual]

Get extra header value.

Parameters

<i>field</i>	field of the header
--------------	---------------------

Returns

the value of the header or nullptr if there is not such header in the request

#### 8.34.2.4 HTTPHeader [] Coherent.UI.URLRequest.GetExtraHeaders ( ) [inline]

Get extra request headers

##### Returns

array of the extra request headers

#### 8.34.2.5 virtual string Coherent.UI.URLRequest.GetMethod ( ) [inline],[virtual]

Get the method of the request.

##### Returns

the method as a string

#### 8.34.2.6 virtual string Coherent.UI.URLRequest.GetReferrer ( ) [inline],[virtual]

Get the referrer of the request.

##### Returns

the referrer of the request

#### 8.34.2.7 virtual string Coherent.UI.URLRequest.GetURL ( ) [inline],[virtual]

Get the URL of the request.

##### Returns

the URL of the request

#### 8.34.2.8 virtual void Coherent.UI.URLRequest.RemoveExtraHeader ( string field ) [inline],[virtual]

Remove extra header from the request.

##### Parameters

<i>field</i>	field of the header to be removed
--------------	-----------------------------------

#### 8.34.2.9 virtual void Coherent.UI.URLRequest.SetExtraHeader ( string field, string content ) [inline],[virtual]

Set extra header for the request. If there is already a header with that field name its content will be replaced.

##### Parameters

<i>field</i>	name of the header
<i>content</i>	new content of the header

#### 8.34.2.10 void Coherent.UI.URLRequest.SetExtraHeaders ( HTTPHeader[] headers ) [inline]

Set the extra headers.

## Parameters

<i>headers</i>	the array of new headers
----------------	--------------------------

8.34.2.11 `virtual void Coherent.UI.URLRequest.SetMethod ( string method )` `[inline],[virtual]`

Set the method of the request.

## Parameters

<i>method</i>	the new method as string. method must be a valid HTTP method name
---------------	---

8.34.2.12 `virtual void Coherent.UI.URLRequest.SetReferrer ( string referrer )` `[inline],[virtual]`

Set the referrer of the request.

## Parameters

<i>referrer</i>	the new referrer URL for the request
-----------------	--------------------------------------

8.34.2.13 `virtual void Coherent.UI.URLRequest.SetURL ( string url )` `[inline],[virtual]`

Replace the URL of the request.

## 8.35 Coherent.UI.Binding.Value Class Reference

Type for representing generic JavaScript values

### Public Member Functions

- [Value](#) ()  
*Create a null value*
- [Value](#) (bool value)  
*Create a boolean value*
- [Value](#) (int value)  
*Create an integer value*
- [Value](#) (uint value)  
*Create an unsigned integer value*
- [Value](#) (sbyte value)  
*Create a signed byte value*
- [Value](#) (byte value)  
*Create a byte value*
- [Value](#) (float value)  
*Create a float value*
- [Value](#) (double value)  
*Create a double value*
- [Value](#) (string value)  
*Create a string value*
- [Value](#) ([Value](#)[] value)  
*Create an array value*

- [Value](#) (Dictionary< string, [Value](#) > value)  
*Create an object value*
- [Value](#) ([ValueObject](#) value)  
*Create an object value*
- override bool [Equals](#) (object obj)  
*Compare to another object*
- override int [GetHashCode](#) ()  
*Hash code for [Value](#)*

### Static Public Member Functions

- static implicit [operator bool](#) ([Value](#) value)  
*cast a value to a boolean*
- static implicit [operator int](#) ([Value](#) value)  
*cast a value to an int*
- static implicit [operator uint](#) ([Value](#) value)  
*cast a value to an uint*
- static implicit [operator sbyte](#) ([Value](#) value)  
*cast a value to a signed byte*
- static implicit [operator byte](#) ([Value](#) value)  
*cast a value to a byte*
- static implicit [operator float](#) ([Value](#) value)  
*cast a value to a float*
- static implicit [operator double](#) ([Value](#) value)  
*cast a value to a double*
- static implicit [operator string](#) ([Value](#) value)  
*cast a value to a string*
- static implicit [operator Value\[\]](#) ([Value](#) value)  
*cast a value to an array*
- static implicit [operator Dictionary< string, Value >](#) ([Value](#) value)  
*cast a value to a dictionary of properties*
- static bool [operator==](#) ([Value](#) lhs, [Value](#) rhs)  
*Compares two [Value](#) objects*
- static bool [operator!=](#) ([Value](#) lhs, [Value](#) rhs)  
*Compares two [Value](#) objects*

### Public Attributes

- [ValueType](#) Type  
*The type of the data stored in this [Value](#) instance*

#### 8.35.1 Detailed Description

Type for representing generic JavaScript values

#### 8.35.2 Constructor & Destructor Documentation

##### 8.35.2.1 Coherent.UI.Binding.Value.Value ( ) [inline]

Create a null value



### 8.35.2.2 Coherent.UI.Binding.Value.Value ( bool *value* ) [inline]

Create a boolean value

## Parameters

<i>value</i>	the integer value of the <a href="#">Value</a> object
--------------	---

**8.35.2.3 Coherent.UI.Binding.Value.Value ( int *value* ) [inline]**

Create an integer value

## Parameters

<i>value</i>	the integer value of the <a href="#">Value</a> object
--------------	---

**8.35.2.4 Coherent.UI.Binding.Value.Value ( uint *value* ) [inline]**

Create an unsigned integer value

## Parameters

<i>value</i>	the unsigned integer value of the <a href="#">Value</a> object
--------------	--

**8.35.2.5 Coherent.UI.Binding.Value.Value ( sbyte *value* ) [inline]**

Create a signed byte value

## Parameters

<i>value</i>	the signed byte value of the <a href="#">Value</a> object
--------------	---

**8.35.2.6 Coherent.UI.Binding.Value.Value ( byte *value* ) [inline]**

Create a byte value

## Parameters

<i>value</i>	the byte value of the <a href="#">Value</a> object
--------------	--

**8.35.2.7 Coherent.UI.Binding.Value.Value ( float *value* ) [inline]**

Create a float value

## Parameters

<i>value</i>	the float value of the <a href="#">Value</a> object
--------------	---

**8.35.2.8 Coherent.UI.Binding.Value.Value ( double *value* ) [inline]**

Create a double value

## Parameters

<i>value</i>	the double value of the <a href="#">Value</a> object
--------------	--

#### 8.35.2.9 Coherent.UI.Binding.Value.Value ( string *value* ) [inline]

Create a string value

## Parameters

<i>value</i>	the string value of the <a href="#">Value</a> object
--------------	--

**8.35.2.10** `Coherent.UI.Binding.Value.Value ( Value[] value ) [inline]`

Create an array value

## Parameters

<i>value</i>	the array value of the <a href="#">Value</a> object
--------------	---

**8.35.2.11** `Coherent.UI.Binding.Value.Value ( Dictionary< string, Value > value ) [inline]`

Create an object value

## Parameters

<i>value</i>	dictionary of properties of the <a href="#">Value</a> object
--------------	--

**8.35.2.12** `Coherent.UI.Binding.Value.Value ( ValueObject value ) [inline]`

Create an object value

## Parameters

<i>value</i>	dictionary of properties of the <a href="#">Value</a> object
--------------	--

**8.35.3 Member Function Documentation****8.35.3.1** `override bool Coherent.UI.Binding.Value.Equals ( object obj ) [inline]`

Compare to another object

## Parameters

<i>obj</i>	
------------	--

## Returns

true if obj is a [Value](#) instance equal to this one

**8.35.3.2** `override int Coherent.UI.Binding.Value.GetHashCode ( ) [inline]`

Hash code for [Value](#)

## Returns

hash code for the current instance

**8.35.3.3** `static implicit Coherent.UI.Binding.Value.operator bool ( Value value ) [inline],[static]`

cast a value to a boolean

## Parameters

<i>value</i>	value to be casted
--------------	--------------------

## Returns

the boolean stored in the value

**8.35.3.4 static implicit Coherent.UI.Binding.Value.operator byte ( Value value ) [inline],[static]**

cast a value to a byte

## Parameters

<i>value</i>	value to be cast
--------------	------------------

## Returns

the byte stored in the value

**8.35.3.5 static implicit Coherent.UI.Binding.Value.operator Dictionary< string, Value > ( Value value ) [inline],[static]**

cast a value to a dictionary of properties

## Parameters

<i>value</i>	value to be casted
--------------	--------------------

## Returns

the dictionary of properties of the value

**8.35.3.6 static implicit Coherent.UI.Binding.Value.operator double ( Value value ) [inline],[static]**

cast a value to a double

## Parameters

<i>value</i>	value to be casted
--------------	--------------------

## Returns

the double stored in the value

**8.35.3.7 static implicit Coherent.UI.Binding.Value.operator float ( Value value ) [inline],[static]**

cast a value to a float

## Parameters

<i>value</i>	value to be casted
--------------	--------------------

## Returns

the float stored in the value

8.35.3.8    `static implicit Coherent.UI.Binding.Value.operator int ( Value value )    [inline],[static]`

cast a value to an int

## Parameters

<i>value</i>	value to be casted
--------------	--------------------

## Returns

the int stored in the value

**8.35.3.9 static implicit Coherent.UI.Binding.Value.operator sbyte ( Value value ) [inline],[static]**

cast a value to a signed byte

## Parameters

<i>value</i>	value to be cast
--------------	------------------

## Returns

the signed byte stored in the value

**8.35.3.10 static implicit Coherent.UI.Binding.Value.operator string ( Value value ) [inline],[static]**

cast a value to a string

## Parameters

<i>value</i>	value to be casted
--------------	--------------------

## Returns

the string stored in the value

**8.35.3.11 static implicit Coherent.UI.Binding.Value.operator uint ( Value value ) [inline],[static]**

cast a value to an uint

## Parameters

<i>value</i>	value to be casted
--------------	--------------------

## Returns

the unsigned int stored in the value

**8.35.3.12 static implicit Coherent.UI.Binding.Value.operator Value[] ( Value value ) [inline],[static]**

cast a value to an array

## Parameters

<i>value</i>	value to be casted
--------------	--------------------

## Returns

the array of [Value](#) objects stored in the value

8.35.3.13 `static bool Coherent.UI.Binding.Value.operator!= ( Value lhs, Value rhs )` `[inline],[static]`

Compares two [Value](#) objects



## Parameters

<i>lhs</i>	object to be compared
<i>rhs</i>	object to be compared

## Returns

false if the objects are from the same type and have the same values

8.35.3.14 `static bool Coherent.UI.Binding.Value.operator==( Value lhs, Value rhs ) [inline],[static]`

Compares two [Value](#) objects

## Parameters

<i>lhs</i>	object to be compared
<i>rhs</i>	object to be compared

## Returns

true if the objects are from the same type and have the same values

## 8.35.4 Member Data Documentation

8.35.4.1 **ValueType** `Coherent.UI.Binding.Value.Type`

The type of the data stored in this [Value](#) instance

## 8.36 Coherent.UI.Binding.ValueObject Class Reference

Class for compound JavaScript objects, behaves like a `Dictionary<string, Value>`

Inherits `IDictionary< string, Value >`, and `IDictionary`.

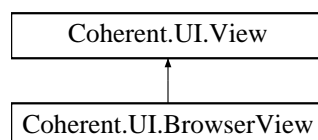
## 8.36.1 Detailed Description

Class for compound JavaScript objects, behaves like a `Dictionary<string, Value>`

## 8.37 Coherent.UI.View Class Reference

Class that encapsulates a [UI](#)

Inheritance diagram for `Coherent.UI.View`:



## Public Member Functions

- [BoundEventHandle BindCall](#) (string name, System.Delegate handler)  
*Expose C++ handler to be called from [UI](#).*
- [BoundEventHandle RegisterForEvent](#) (string name, System.Delegate handler)  
*Expose C++ handler to be called when a specific event occurs.*
- void [UnregisterFromEvent](#) ([BoundEventHandle](#) handle)  
*Remove a registered C++ handler.*
- void [UnbindCall](#) ([BoundEventHandle](#) handle)  
*Remove a bound C++ handler.*
- void [UnbindObject](#) (object target)  
*Remove all handlers bound to a specific object.*
- virtual [ViewType GetViewType](#) ()  
*Get the type of the view.*
- virtual void [Destroy](#) ()  
*Destroys this view. After a call to Destroy calling any other method except the destructor results in undefined behavior*
- virtual void [SetFocus](#) ()  
*Sets this view on focus.*
- virtual void [KillFocus](#) ()  
*Removes the focus from this view.*
- virtual void [MouseEvent](#) ([MouseEventData](#) arg0)  
*Sends a mouse event to the [UI](#) renderer.*
- virtual void [KeyEvent](#) ([KeyEventData](#) arg0)  
*Sends a key event to the [UI](#) renderer.*
- virtual void [TouchEvent](#) ([TouchEventData](#) events, uint count)  
*Sends a touch event to the [UI](#) renderer.*
- virtual void [MouseActivate](#) ()  
*Sends a mouse event to the [UI](#) renderer.*
- virtual int [GetWidth](#) ()  
*Gets the width of the*
- virtual int [GetHeight](#) ()  
*Gets the height of the*
- virtual void [Resize](#) (uint width, uint height)  
*Send a resize event to the [UI](#) renderer. Resize will result in multiple calls to CreateBuffers and DestroyBuffers in the listener*
- virtual void [IssueMouseOnUIQuery](#) (float normX, float normY)  
*Issues a query on the [Calling IsMouseOnView](#) while a query is in-flight is an error and will be logged - the result returned by IsMouseOnView in that case is undefined.*
- virtual [ViewErrorType HasMouseQueryFinished](#) ()  
*Returns the status of the mouse query.*
- virtual void [FetchMouseOnUIQuery](#) ()  
*Ends a mouse-on-UI query. Waits for the result. Should be called as far apart from IssueMouseOnUIQuery in order to maximize the chances that the query has finished at the time of this call and no waiting will be performed*
- virtual bool [IsMouseOnView](#) ()  
*Checks if the mouse is currently on the logical view (a part of the view that has elements on it) or not (click-through). Uses the coordinates last set in [IssueMouseOnUIQuery](#). Calling the method while a query is in-flight (between a call to [IssueMouseOnUIQuery](#) and [FetchMouseOnUIQuery](#)) results in a warning and the return result is undefined*
- virtual void [SetClickThroughAlphaThreshold](#) (float threshold)  
*The alpha value of the pixels is used to determine if the mouse is on an element or on the background (click-through). All pixels below or equal to the alpha threshold (default = 0) are marked as not-belonging to the*
- virtual float [GetClickThroughAlphaThreshold](#) ()  
*Get the currently set alpha threshold for click-through queries.*

- virtual void [SetTargetFramerate](#) (int target)  
*Sets the target frame-rate for the*
- virtual int [GetTargetFramerate](#) ()  
*Get the currently set target frame-rate for the*
- virtual bool [IsOnDemand](#) ()  
*Tells if this is an on-demand view. Frames must be explicitly requested for on-demand views (available only in Full versions of 'Coherent::UI::View::RequestFrame')*
- virtual [ViewErrorType RequestFrame](#) ()  
*Requests a new frame for rendering. All triggered events before this call will be accounted for in the rendered view. Only one frame per-view could be requested simultaneously (available only in Full versions of*
- virtual [ViewErrorType RequestFrame](#) (double timeSinceArbitraryMoment)  
*Requests a new frame for rendering. All triggered events before this call will be accounted for in the rendered view. Only one frame per-view could be requested simultaneously (available only in Full versions of*
- virtual [ViewErrorType PeekFrameReady](#) ()  
*Tells if the last requested frame is ready or not. Does not block. This call must always be made between calls to RequestFrame and FetchSurface. (available only in Full versions of 'Coherent::UI::View::RequestFrame' 'Coherent::UI::View::FetchSurface')*
- virtual [ViewErrorType FetchSurface](#) ()  
*Returns the requested surface. Works only for on-demand views. All buffered view should use the system's FetchSurfaces method. This call will block until the surface is ready. (available only in Full versions of*
- virtual [ViewErrorType DrawSurface](#) ()  
*[EXPERIMENTAL - DO NOT USE]*
- virtual bool [UsesSharedMemory](#) ()  
*Tells if this view uses shared memory for image transport.*
- virtual bool [IsTransparent](#) ()  
*Tells if this view is transparent or not.*
- virtual bool [SupportsClickThrough](#) ()  
*Tells if this view supports click-through queries or not.*
- virtual void [Redraw](#) ()  
*Request a redraw of the current view.*
- virtual void [DownloadUrl](#) (string path)  
*Initiates a download request on the specified path.*
- virtual void [Load](#) (string path)  
*Loads a new path in the*
- virtual void [Reload](#) (bool ignoreCache)  
*Reloads the last requested or navigated-to path.*
- virtual string [GetCurentViewPath](#) ()  
*Get the last requested or navigated-to path.*
- virtual string [GetLastRequestedPath](#) ()  
*Get the last requested path to load for the*
- virtual string [GetLastLoadedPath](#) ()  
*Get the last successfully loaded path for the*
- virtual string [GetLastFailedPath](#) ()  
*Get the last failed path for the*
- virtual void [ExecuteScript](#) (string script)  
*Execute arbitrary JavaScript code.*
- virtual void [TriggerEvent](#) (string eventName)  
*Triggers a UI event.*
- virtual void [ReplyToJavaScriptMessage](#) (bool success, string userInput)  
*Sends a reply to a javascript message triggered for the view.*
- virtual void [AuthCredentialsResponse](#) (string username, string password, bool success)  
*Sends credentials for a pending authentication request for the view.*

- virtual void [SetScriptError](#) ([ScriptCallErrorType](#) error, string message)  
*Sets an error for the currently executing handler.*
- virtual void [InterceptURLRequests](#) (bool intercept)  
*Controls intercepting URL requests. For each request*
- virtual void [IMEActivate](#) (bool active)  
*Sets if the*
- virtual bool [IMEIsActive](#) ()  
*Tells if IME events are currently active for this view.*
- virtual bool [IMESetComposition](#) (string composition, uint cursorPos, uint targetStart, uint targetEnd)  
*Updates the internal state of the IME composition in the*
- virtual bool [IMEConfirmComposition](#) (string composition)  
*Confirms the current IME composition.*
- virtual bool [IMECancelComposition](#) ()  
*Cancels the current IME composition.*
- virtual [ImageData](#) [CreateImageData](#) (string name, int width, int height, IntPtr data, bool flipY)  
*Create an*

### 8.37.1 Detailed Description

Class that encapsulates a [UI](#)

### 8.37.2 Member Function Documentation

8.37.2.1 virtual void Coherent.UI.View.AuthCredentialsResponse ( string *username*, string *password*, bool *success* )  
[inline],[virtual]

Sends credentials for a pending authentication request for the view.

Parameters

<i>username</i>	the username
<i>password</i>	the password
<i>success</i>	defines whether the response is valid, i.e. if the request should be answered or canceled

8.37.2.2 [BoundEventHandle](#) Coherent.UI.View.BindCall ( string *name*, System.Delegate *handler* ) [inline]

Expose C++ handler to be called from [UI](#).

Parameters

<i>name</i>	name for the handler in the <a href="#">UI</a>
<i>handler</i>	handler to be executed

Returns

bound handle so the handler can be unbound later

8.37.2.3 virtual [ImageData](#) Coherent.UI.View.CreateImageData ( string *name*, int *width*, int *height*, IntPtr *data*, bool *flipY* )  
[inline],[virtual]

Create an

## Parameters

<i>name</i>	the name of the image data object
<i>width</i>	the width of the image
<i>height</i>	the height of the image
<i>data</i>	initial data to fill the image data with
<i>flipY</i>	flips vertically the content of the source image

## Returns

the image data object created in the DOM of this [View](#)

#### 8.37.2.4 virtual void Coherent.UI.View.Destroy ( ) [inline],[virtual]

Destroys this view. After a call to Destroy calling any other method except the destructor results in undefined behavior

#### 8.37.2.5 virtual void Coherent.UI.View.DownloadUrl ( string path ) [inline],[virtual]

Initiates a download request on the specified path.

#### 8.37.2.6 virtual ViewErrorType Coherent.UI.View.DrawSurface ( ) [inline],[virtual]

[EXPERIMENTAL - DO NOT USE]

#### 8.37.2.7 virtual void Coherent.UI.View.ExecuteScript ( string script ) [inline],[virtual]

Execute arbitrary JavaScript code.

## Parameters

<i>script</i>	code to be evaluated in the context of the main frame of the view
---------------	---

#### 8.37.2.8 virtual void Coherent.UI.View.FetchMouseOnUIQuery ( ) [inline],[virtual]

Ends a mouse-on-UI query. Waits for the result. Should be called as far apart from IssueMouseOnUIQuery in order to maximize the chances that the query has finished at the time of this call and no waiting will be performed

#### 8.37.2.9 virtual ViewErrorType Coherent.UI.View.FetchSurface ( ) [inline],[virtual]

Returns the requested surface. Works only for on-demand views. All buffered view should use the system's Fetch-Surfaces method. This call will block until the surface is ready. (available only in Full versions of

## Returns

the outcome of the operation

#### 8.37.2.10 virtual float Coherent.UI.View.GetClickThroughAlphaThreshold ( ) [inline],[virtual]

Get the currently set alpha threshold for click-through queries.

## Returns

the currently set alpha threshold

**8.37.2.11** `virtual string Coherent.UI.View.GetCurentViewPath ( ) [inline],[virtual]`

Get the last requested or navigated-to path.

Returns

the path as null-terminated string

**8.37.2.12** `virtual int Coherent.UI.View.GetHeight ( ) [inline],[virtual]`

Gets the height of the

Returns

the height of the [View](#)

**8.37.2.13** `virtual string Coherent.UI.View.GetLastFailedPath ( ) [inline],[virtual]`

Get the last failed path for the

Returns

the path as null-terminated string

**8.37.2.14** `virtual string Coherent.UI.View.GetLastLoadedPath ( ) [inline],[virtual]`

Get the last successfully loaded path for the

Returns

the path as null-terminated string

**8.37.2.15** `virtual string Coherent.UI.View.GetLastRequestedPath ( ) [inline],[virtual]`

Get the last requested path to load for the

Returns

the path as null-terminated string

**8.37.2.16** `virtual int Coherent.UI.View.GetTargetFramerate ( ) [inline],[virtual]`

Get the currently set target frame-rate for the

Returns

the current target frame-rate (available only in Full versions of [Coherent UI](#))

**8.37.2.17** `virtual ViewType Coherent.UI.View.GetViewType ( ) [inline],[virtual]`

Get the type of the view.

Returns

the type of the view

Reimplemented in [Coherent.UI.BrowserView](#).

8.37.2.18 `virtual int Coherent.UI.View.GetWidth ( ) [inline],[virtual]`

Gets the width of the

Returns

the width of the [View](#)

8.37.2.19 `virtual ViewErrorType Coherent.UI.View.HasMouseQueryFinished ( ) [inline],[virtual]`

Returns the status of the mouse query.

Returns

the result will be Success if the query has finished, QueryNotReady otherwise. If no query has been issued InvalidCall will be returned.

8.37.2.20 `virtual void Coherent.UI.View.IMEActivate ( bool active ) [inline],[virtual]`

Sets if the

Parameters

<i>active</i>	whether to activate IME handling for this view
---------------	--

8.37.2.21 `virtual bool Coherent.UI.View.IMECancelComposition ( ) [inline],[virtual]`

Cancels the current IME composition.

Returns

the outcome of the operation (will fail if IME is inactive for this view)

8.37.2.22 `virtual bool Coherent.UI.View.IMEConfirmComposition ( string composition ) [inline],[virtual]`

Confirms the current IME composition.

Parameters

<i>composition</i>	the string to confirm
--------------------	-----------------------

Returns

the outcome of the operation (will fail if IME is inactive for this view)

8.37.2.23 `virtual bool Coherent.UI.View.IMEIsActive ( ) [inline],[virtual]`

Tells if IME events are currently active for this view.

Returns

if the view has IME active

8.37.2.24 `virtual bool Coherent.UI.View.IMESetComposition ( string composition, uint cursorPos, uint targetStart, uint targetEnd ) [inline],[virtual]`

Updates the internal state of the IME composition in the

## Parameters

<i>composition</i>	the composition string so far
<i>the</i>	cursor position in the composition
<i>targetStart</i>	the index at the start of the selection
<i>targetEnd</i>	the index at the end of the selection

## Returns

the outcome of the operation (will fail if IME is inactive for this view)

**8.37.2.25** `virtual void Coherent.UI.View.InterceptURLRequests ( bool intercept ) [inline],[virtual]`

Controls intercepting URL requests. For each request

## Parameters

<i>intercept</i>	whether to intercept all URL requests
------------------	---------------------------------------

**8.37.2.26** `virtual bool Coherent.UI.View.IsMouseOnView ( ) [inline],[virtual]`

Checks if the mouse is currently on the logical view (a part of the view that has elements on it) or not (click-through). Uses the coordinates last set in `IssueMouseOnUIQuery`. Calling the method while a query is in-flight (between a call to `IssueMouseOnUIQuery` and `FetchMouseOnUIQuery`) results in a warning and the return result is undefined

## Returns

true if the mouse is on the view

**8.37.2.27** `virtual bool Coherent.UI.View.IsOnDemand ( ) [inline],[virtual]`

Tells if this is an on-demand view. Frames must be explicitly requested for on-demand views (available only in Full versions of '[Coherent::UI::View::RequestFrame](#)')

## Returns

true if this is an on-demand view

**8.37.2.28** `virtual void Coherent.UI.View.IssueMouseOnUIQuery ( float normX, float normY ) [inline],[virtual]`

Issues a query on the `IsMouseOnView` while a query is in-flight is an error and will be logged - the result returned by `IsMouseOnView` in that case is undefined.

Put as much processing as possible between `IssueMouseOnUIQuery` and the corresponding `FetchMouseOnUIQuery`, this allows leveraging the cost of the query and to reduce it to  $\sim 0$ . The best way to do this is call `IssueMouseOnUIQuery` early in the frame, perform all game rendering and the call `FetchMouseOnUIQuery` followed by `IsMouseOnView` to update the needed data.

## Parameters

<i>normX</i>	the x coordinate of the mouse in the normalized space of the view [0..1]
--------------	--



<i>normY</i>	the y coordinate of the mouse in the normalized space of the view [0..1]
--------------	--

**8.37.2.29** `virtual bool Coherent.UI.View.IsTransparent ( ) [inline],[virtual]`

Tells if this view is transparent or not.

Returns

true if the view is transparent

**8.37.2.30** `virtual void Coherent.UI.View.KeyEvent ( KeyEventData arg0 ) [inline],[virtual]`

Sends a key event to the [UI](#) renderer.

Parameters

<i>event</i>	the key event
--------------	---------------

**8.37.2.31** `virtual void Coherent.UI.View.KillFocus ( ) [inline],[virtual]`

Removes the focus from this view.

**8.37.2.32** `virtual void Coherent.UI.View.Load ( string path ) [inline],[virtual]`

Loads a new path in the

Parameters

<i>path</i>	the path to load
-------------	------------------

**8.37.2.33** `virtual void Coherent.UI.View.MouseActivate ( ) [inline],[virtual]`

Sends a mouse event to the [UI](#) renderer.

**8.37.2.34** `virtual void Coherent.UI.View.MouseEvent ( MouseEventData arg0 ) [inline],[virtual]`

Sends a mouse event to the [UI](#) renderer.

Parameters

<i>event</i>	the mouse event
--------------	-----------------

**8.37.2.35** `virtual ViewErrorType Coherent.UI.View.PeekFrameReady ( ) [inline],[virtual]`

Tells if the last requested frame is ready or not. Does not block. This call must always be made between calls to RequestFrame and FetchSurface. (available only in Full versions of '[Coherent::UI::View::RequestFrame](#)' '[Coherent::UI::View::FetchSurface](#)'

Returns

VE\_Success; VE\_FrameNotReady; VE\_FrameNotRequested; VE\_InvalidCall

**8.37.2.36** `virtual void Coherent.UI.View.Redraw ( ) [inline],[virtual]`

Request a redraw of the current view.

**8.37.2.37** `BoundEventHandle Coherent.UI.View.RegisterForEvent ( string name, System.Delegate handler ) [inline]`

Expose C++ handler to be called when a specific event occurs.

Parameters

<i>name</i>	name of the event
<i>handler</i>	handler to be executed

Returns

bound handle so the handler can be unregistered

**8.37.2.38** `virtual void Coherent.UI.View.Reload ( bool ignoreCache ) [inline],[virtual]`

Reloads the last requested or navigated-to path.

Parameters

<i>ignoreCache</i>	whether to ignore the cache for the reload
--------------------	--

**8.37.2.39** `virtual void Coherent.UI.View.ReplyToJavaScriptMessage ( bool success, string userInput ) [inline],[virtual]`

Sends a reply to a javascript message triggered for the view.

Parameters

<i>success</i>	defines whether the user replies positively to the message (i.e. clicks the OK button)
<i>userInput</i>	the user reply message (only used in prompt messages)

**8.37.2.40** `virtual ViewErrorType Coherent.UI.View.RequestFrame ( ) [inline],[virtual]`

Requests a new frame for rendering. All triggered events before this call will be accounted for in the rendered view. Only one frame per-view could be requested simultaneously (available only in Full versions of

Returns

the outcome of the operation

**8.37.2.41** `virtual ViewErrorType Coherent.UI.View.RequestFrame ( double timeSinceArbitraryMoment ) [inline],[virtual]`

Requests a new frame for rendering. All triggered events before this call will be accounted for in the rendered view. Only one frame per-view could be requested simultaneously (available only in Full versions of

Returns

the outcome of the operation

8.37.2.42 `virtual void Coherent.UI.View.Resize ( uint width, uint height ) [inline],[virtual]`

Send a resize event to the [UI](#) renderer. Resize will result in multiple calls to `CreateBuffers` and `DestroyBuffers` in the listener

## Parameters

<i>width</i>	the new width of the <a href="#">View</a>
<i>height</i>	the new height of the <a href="#">View</a>

**8.37.2.43** `virtual void Coherent.UI.View.SetClickThroughAlphaThreshold ( float threshold )` `[inline],[virtual]`

The alpha value of the pixels is used to determine if the mouse is on an element or on the background (click-through). All pixels below or equal to the alpha threshold (default = 0) are marked as not-belonging to the

## Parameters

<i>threshold</i>	sets the new alpha threshold
------------------	------------------------------

**8.37.2.44** `virtual void Coherent.UI.View.SetFocus ( )` `[inline],[virtual]`

Sets this view on focus.

**8.37.2.45** `virtual void Coherent.UI.View.SetScriptError ( ScriptCallErrorType error, string message )` `[inline],[virtual]`

Sets an error for the currently executing handler.

## Parameters

<i>error</i>	the type of the error
<i>message</i>	the error message

**8.37.2.46** `virtual void Coherent.UI.View.SetTargetFramerate ( int target )` `[inline],[virtual]`

Sets the target frame-rate for the

## Parameters

<i>target</i>	the new target frame-rate
---------------	---------------------------

**8.37.2.47** `virtual bool Coherent.UI.View.SupportsClickThrough ( )` `[inline],[virtual]`

Tells if this view supports click-through queries or not.

## Returns

true if click-through is supported

**8.37.2.48** `virtual void Coherent.UI.View.TouchEvent ( TouchEventData events, uint count )` `[inline],[virtual]`

Sends a touch event to the [UI](#) renderer.

## Parameters

---

<i>events</i>	array of touch events
<i>count</i>	the count of the passed touch events in the array

8.37.2.49 `virtual void Coherent.UI.View.TriggerEvent ( string eventName ) [inline],[virtual]`

Triggers a [UI](#) event.

8.37.2.50 `void Coherent.UI.View.UnbindCall ( BoundEventHandle handle ) [inline]`

Remove a bound C++ handler.

Parameters

<i>handle</i>	the handle of the handler to be unbound
---------------	---

8.37.2.51 `void Coherent.UI.View.UnbindObject ( object target ) [inline]`

Remove all handlers bound to a specific object.

Parameters

<i>object</i>	the object whose handlers will be removed
---------------	---

8.37.2.52 `void Coherent.UI.View.UnregisterFromEvent ( BoundEventHandle handle ) [inline]`

Remove a registered C++ handler.

Parameters

<i>handle</i>	the handle of the event handler to be unregistered
---------------	--

8.37.2.53 `virtual bool Coherent.UI.View.UsesSharedMemory ( ) [inline],[virtual]`

Tells if this view uses shared memory for image transport.

Returns

true if the view uses shared memory, false if shared textures

## 8.38 Coherent.UI.ViewError Class Reference

Encapsulates a view-related error.

Inherits [IDisposable](#).

### Properties

- [ViewErrorType ErrorCode](#) [get, set]  
*Indicates an error code.*
- string [Error](#) [get, set]  
*Error description.*

### 8.38.1 Detailed Description

Encapsulates a view-related error.

### 8.38.2 Property Documentation

#### 8.38.2.1 `string Coherent.UI.ViewError.Error` [get], [set]

Error description.

#### 8.38.2.2 `ViewErrorType Coherent.UI.ViewError.ErrorCode` [get], [set]

Indicates an error code.

## 8.39 Coherent.UI.ViewInfo Class Reference

Encapsulates the options of a

Inherits `IDisposable`.

### Properties

- `int Width` [get, set]  
*The Width of a*
- `int Height` [get, set]  
*The Height of a*
- `bool UsesSharedMemory` [get, set]  
*Flags if a*
- `bool SupportClickThrough` [get, set]  
*Defines if the view should support click-through queries. Not supporting click-through increases performance slightly.*
- `float ClickThroughAlphaThreshold` [get, set]  
*The alpha threshold for click-through queries.*
- `bool IsTransparent` [get, set]  
*Defines if a view is transparent or not. A transparent view can be have transparent parts that will be correctly blended with the client's surfaces. Non-transparent views are always implicitly composited as if on a white background. Don't use transparent views unless you have transparent parts in the page because non-transparent view are slightly faster. Do not blend non-transparent views.*
- `int TargetFrameRate` [get, set]  
*The frame-rate of the view will never exceed the target set here. It will also never exceed the rate at which the System Update is called. (available only in Full versions of*
- `bool IsOnDemand` [get, set]  
*Defines if a view is an on-demand view. On-demand views require explicit frame requests. (available only in Full versions of*
- `bool ControlTimeOnDemand` [get, set]  
*Defines if the client controls the time of on-demand views himself. If not*
- `bool ForceSoftwareRendering` [get, set]  
*Forces the view to use software rendering. If the application is GPU-bound pure software Views might be a good choice. CSS 3D transforms, WebGL and accelerated Canvas don't work with software Views. Software Views are incompatible with the OnDemand option and work only with shared memory surfaces.*

### 8.39.1 Detailed Description

Encapsulates the options of a

### 8.39.2 Property Documentation

8.39.2.1 float `Coherent.UI.ViewInfo.ClickThroughAlphaThreshold` [get], [set]

The alpha threshold for click-through queries.

8.39.2.2 bool `Coherent.UI.ViewInfo.ControlTimeOnDemand` [get], [set]

Defines if the client controls the time of on-demand views himself. If not

8.39.2.3 bool `Coherent.UI.ViewInfo.ForceSoftwareRendering` [get], [set]

Forces the view to use software rendering. If the application is GPU-bound pure software Views might be a good choice. CSS 3D transforms, WebGL and accelerated Canvas don't work with software Views. Software Views are incompatible with the OnDemand option and work only with shared memory surfaces.

8.39.2.4 int `Coherent.UI.ViewInfo.Height` [get], [set]

The Height of a

8.39.2.5 bool `Coherent.UI.ViewInfo.IsOnDemand` [get], [set]

Defines if a view is an on-demand view. On-demand views require explicit frame requests. (available only in Full versions of

8.39.2.6 bool `Coherent.UI.ViewInfo.IsTransparent` [get], [set]

Defines if a view is transparent or not. A transparent view can be have transparent parts that will be correctly blended with the client's surfaces. Non-transparent views are always implicitly composited as if on a white background. Don't use transparent views unless you have transparent parts in the page because non-transparent view are slightly faster. Do not blend non-transparent views.

8.39.2.7 bool `Coherent.UI.ViewInfo.SupportClickThrough` [get], [set]

Defines if the view should support click-through queries. Not supporting click-through increases performance slightly.

8.39.2.8 int `Coherent.UI.ViewInfo.TargetFrameRate` [get], [set]

The frame-rate of the view will never exceed the target set here. It will also never exceed the rate at which the System Update is called. (available only in Full versions of

8.39.2.9 bool `Coherent.UI.ViewInfo.UsesSharedMemory` [get], [set]

Flags if a

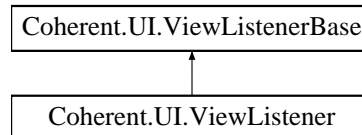
8.39.2.10 `int Coherent.UI.ViewInfo.Width` `[get]`, `[set]`

The Width of a

## 8.40 Coherent.UI.ViewListener Class Reference

Interface that allows clients to listen to

Inheritance diagram for Coherent.UI.ViewListener:



### Public Member Functions

- override void [Release](#) ()  
*Called when the listener is no longer needed by the [UI](#) system.*
- override void [OnViewCreated](#) ([View](#) view)  
*Called when the requested*
- override void [OnDraw](#) ([CoherentHandle](#) handle, bool usesSharedMemory, int width, int height)  
*Called when a new surface has been drawn and is ready to use by the client.*
- override void [OnFinishLoad](#) (int frameId, string validatedPath, bool isMainFrame, int statusCode, [HTTPHeader\[\]](#) headers)  
*Called when a frame has been successfully loaded.*
- override void [OnFailLoad](#) (int frameId, string validatedPath, bool isMainFrame, string error)  
*Called when a frame has been failed loading.*
- override void [OnURLRequest](#) (string url, [URLResponse](#) response)  
*Called before an URL request is made. The default implementation allows all requests.*
- override void [OnReadyForBindings](#) (int frameId, string path, bool isMainFrame)  
*Called when a frame is ready for bindings.*
- override void [OnBindingsReleased](#) (int frameId, string path, bool isMainFrame)  
*Called when the bindings for frame are released.*
- override void [OnStartLoading](#) ()  
*Called when a new path has started loading.*
- override void [OnStopLoading](#) ()  
*Called when all load operations have completed.*
- override void [OnNavigateTo](#) (string path)  
*Called when the view starts navigation to a new path.*
- override void [OnError](#) ([ViewError](#) error)  
*Called when an error occurs for this specific*
- override void [OnScriptMessage](#) ([ViewListenerBase.MessageLevel](#) level, string message, string sourceId, int line)  
*Called when a message is sent from a script running in this specific*
- override void [OnCursorChanged](#) ([CursorTypes](#) cursor)  
*Called when the cursor has changed internally in the*
- override void [OnCallback](#) (string eventName, [Binding.CallbackArguments](#) arguments)  
*Called by the [UI](#) when there is no registered handler for this event.*
- override void [OnJavaScriptMessage](#) (string message, string defaultPrompt, string frameUrl, int messageType)



- Called when the view triggered a javascript message box, i.e. an alert, confirmation dialog or a prompt dialog.*

  - override void [OnGetAuthCredentials](#) (bool isProxy, string host, uint port, string realm, string scheme)

*Called when a view requires authentication credentials.*

  - override void [CreateSurface](#) (bool sharedMemory, uint width, uint height, [SurfaceResponse](#) response)

*Called when the The format for DirectX9 must be D3DFMT\_A8R8G8B8 The format for DirectX10 and DirectX11 must be B8G8R8A8\_UNORM*

  - override void [DestroySurface](#) (CoherentHandle surface, bool usesSharedMemory)

*Called when a surface is unneeded anymore and should be destroyed This function can be called from a thread different than the main UI system thread in order to support client applications with multi-threaded rendering.*

  - override void [OnCertificateError](#) (string url, [CertificateStatus](#) status, [Certificate](#) certificate, [CertificateError-Response](#) response)

*Called when there is an error with the certificate of a particular URL. the certificate and response pointers are valid only for the duration of this call*

  - override void [OnRequestMediaStream](#) ([MediaStreamRequest](#) request)

*Called when the view requests access to a media stream. Media streams are the audio capture (microphone) and video capture (camera) devices on the system.*

  - override void [OnTextInputTypeChanged](#) ([TextInputControlType](#) type, bool canComposeInline)

*Called when the current text input control changes (i.e. the user click an edit-box). Use this methof to decide when to allow for IME input. the method will be called ONLY if IME is activated on this [View](#). [View::IMEActivate](#)*

  - override void [OnCaretRectChanged](#) (uint x, uint y, uint width, uint height)

*Called when the caret changes during IME composition. You can use this method to correctly position a custom IME control & candidate list.*

  - override void [OnIMEShouldCancelComposition](#) ()

*Called when the user must cancel the IME composition due to an event in the*

  - override void [OnFileSelectRequest](#) ([FileSelectRequest](#) request)

*Called when the view requests file selection. It could be either single file, directory or multiple files.*

## Events

- CoherentUI\_OnViewCreated [ViewCreated](#)

*Fired when the requested*
- CoherentUI\_OnDraw [Draw](#)

*Fired when a new surface has been drawn and is ready to use by the client.*
- CoherentUI\_OnFinishLoad [FinishLoad](#)

*Fired when a frame has been successfully loaded.*
- CoherentUI\_OnFailLoad [FailLoad](#)

*Fired when a frame has been failed loading.*
- CoherentUI\_OnURLRequest [URLRequest](#)

*Fired before an URL request is made. The default implementation allows all requests.*
- CoherentUI\_OnReadyForBindings [ReadyForBindings](#)

*Fired when a frame is ready for bindings.*
- CoherentUI\_OnBindingsReleased [BindingsReleased](#)

*Fired when the bindings for frame are released.*
- CoherentUI\_OnStartLoading [StartLoading](#)

*Fired when a new path has started loading.*
- CoherentUI\_OnStopLoading [StopLoading](#)

*Fired when all load operations have completed.*
- CoherentUI\_OnNavigateTo [NavigateTo](#)

*Fired when the view starts navigation to a new path.*
- CoherentUI\_OnError [Error](#)

*Fired when an error occurs for this specific*
- CoherentUI\_OnScriptMessage [ScriptMessage](#)

- Fired when a message is sent from a script running in this specific*
- CoherentUI\_OnCursorChanged [CursorChanged](#)
  - Fired when the cursor has changed internally in the*
- CoherentUI\_OnCallback [Callback](#)
  - Fired by the [UI](#) when there is no registered handler for this event.*
- CoherentUI\_OnJavaScriptMessage [JavaScriptMessage](#)
  - Fired when the view triggered a javascript message box, i.e. an alert, confirmation dialog or a prompt dialog.*
- CoherentUI\_OnGetAuthCredentials [GetAuthCredentials](#)
  - Fired when a view requires authentication credentials.*
- CoherentUI\_OnCertificateError [CertificateError](#)
  - Fired when there is an error with the certificate of a particular URL. the certificate and response pointers are valid only for the duration of this call*
- CoherentUI\_OnRequestMediaStream [RequestMediaStream](#)
  - Fired when the view requests access to a media stream. Media streams are the audio capture (microphone) and video capture (camera) devices on the system.*
- CoherentUI\_OnTextInputTypeChanged [TextInputTypeChanged](#)
  - Fired when the current text input control changes (i.e. the user click an edit-box). Use this method to decide when to allow for IME input. the method will be called ONLY if IME is activated on this [View](#). [View::IMEActivate](#)*
- CoherentUI\_OnCaretRectChanged [CaretRectChanged](#)
  - Fired when the caret changes during IME composition. You can use this method to correctly position a custom IME control & candidate list.*
- CoherentUI\_OnIMEShouldCancelComposition [IMEShouldCancelComposition](#)
  - Fired when the user must cancel the IME composition due to an event in the*
- CoherentUI\_OnFileSelectRequest [FileSelectRequest](#)
  - Fired when the view requests file selection. It could be either single file, directory or multiple files.*

## Additional Inherited Members

### 8.40.1 Detailed Description

Interface that allows clients to listen to

### 8.40.2 Member Function Documentation

8.40.2.1 `override void Coherent.UI.ViewListener.CreateSurface ( bool sharedMemory, uint width, uint height, SurfaceResponse response ) [inline], [virtual]`

Called when the The format for DirectX9 must be D3DFMT\_A8R8G8B8 The format for DirectX10 and DirectX11 must be B8G8R8A8\_UNORM

#### Parameters

<i>sharedMemory</i>	true if the surface should be created in shared memory (4 * width * height bytes); false if a shared texture must be created.
<i>width</i>	the width of the surface in pixels
<i>height</i>	the height of the surface in pixels
<i>response</i>	<ul style="list-style-type: none"> <li>• object to hold the response when the surface is created or fails it's creation - must be signaled</li> </ul>

#### Returns

a handle to the created surface

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.40.2.2** `override void Coherent.UI.ViewListener.DestroySurface ( CoherentHandle surface, bool usesSharedMemory )`  
`[inline], [virtual]`

Called when a surface is unneeded anymore and should be destroyed This function can be called from a thread different than the main [UI](#) system thread in order to support client applications with multi-threaded rendering.

#### Parameters

<i>surface</i>	handle to the surface
<i>usesSharedMemory</i>	determines whether the surface parameter is a handle to shared memory or shared texture

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.40.2.3** `override void Coherent.UI.ViewListener.OnBindingsReleased ( int frameId, string path, bool isMainFrame )`  
`[inline], [virtual]`

Called when the bindings for frame are released.

#### Parameters

<i>frameId</i>	the id of the frame
<i>path</i>	the path in the frame
<i>isMainFrame</i>	true if this is the main frame of the view

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.40.2.4** `override void Coherent.UI.ViewListener.OnCallback ( string eventName, Binding.CallbackArguments arguments )`  
`[inline], [virtual]`

Called by the [UI](#) when there is no registered handler for this event.

#### Parameters

<i>eventName</i>	name of the event
<i>arguments</i>	arguments of the event invocation

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.40.2.5** `override void Coherent.UI.ViewListener.OnCaretRectChanged ( uint x, uint y, uint width, uint height )` `[inline], [virtual]`

Called when the caret changes during IME composition. You can use this method to correctly position a custom IME control & candidate list.

#### Parameters

<i>the</i>	x position of the selection caret
<i>the</i>	y position of the selection caret
<i>the</i>	width of the selection caret
<i>the</i>	height of the selection caret

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.40.2.6** `override void Coherent.UI.ViewListener.OnCertificateError ( string url, CertificateStatus status, Certificate certificate, CertificateErrorResponse response )` `[inline], [virtual]`

Called when there is an error with the certificate of a particular URL. the certificate and response pointers are valid only for the duration of this call

## Parameters

<i>url</i>	the url of the request
<i>status</i>	the error status of the certificate
<i>certificate</i>	the certificate details. This pointer will be valid only for this call
<i>response</i>	object to signal whether to continue loading the URL. This pointer will be valid only for this call

Reimplemented from [Coherent.UI.ViewListenerBase](#).

8.40.2.7 `override void Coherent.UI.ViewListener.OnCursorChanged ( CursorTypes cursor ) [inline],[virtual]`

Called when the cursor has changed internally in the

## Parameters

<i>cursor</i>	the new cursor
---------------	----------------

Reimplemented from [Coherent.UI.ViewListenerBase](#).

8.40.2.8 `override void Coherent.UI.ViewListener.OnDraw ( CoherentHandle handle, bool usesSharedMemory, int width, int height ) [inline],[virtual]`

Called when a new surface has been drawn and is ready to use by the client.

## Parameters

<i>handle</i>	a handle to one of the buffers created by CreateSurface. May be a shared memory buffer or a shared texture depending on the way the <a href="#">View</a> was created. The handle is valid only during this call
<i>usesSharedMemory</i>	determines whether the handle parameter is a handle to shared memory or shared texture
<i>width</i>	the width of the surface
<i>height</i>	the height of the surface

Reimplemented from [Coherent.UI.ViewListenerBase](#).

8.40.2.9 `override void Coherent.UI.ViewListener.OnError ( ViewError error ) [inline],[virtual]`

Called when an error occurs for this specific

## Parameters

<i>error</i>	error description
--------------	-------------------

Reimplemented from [Coherent.UI.ViewListenerBase](#).

8.40.2.10 `override void Coherent.UI.ViewListener.OnFailLoad ( int frameId, string validatedPath, bool isMainFrame, string error ) [inline],[virtual]`

Called when a frame has been failed loading.

## Parameters

<i>frameId</i>	the id of the frame
<i>validatedPath</i>	the path in the frame

<i>isMainFrame</i>	true if this is the main frame of the <a href="#">View</a>
<i>error</i>	error message for the failure

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.40.2.11** `override void Coherent.UI.ViewListener.OnFileSelectRequest ( FileSelectRequest request ) [inline], [virtual]`

Called when the view requests file selection. It could be either single file, directory or multiple files.

Parameters

<i>request</i>	contains the file selection params for the request
----------------	--

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.40.2.12** `override void Coherent.UI.ViewListener.OnFinishLoad ( int frameId, string validatedPath, bool isMainFrame, int statusCode, HTTPHeader[] headers ) [inline], [virtual]`

Called when a frame has been successfully loaded.

Parameters

<i>frameId</i>	the id of the loaded frame
<i>validatedPath</i>	the path loaded in the frame
<i>isMainFrame</i>	true if this is the main frame of the <a href="#">View</a>
<i>statusCode</i>	the status code of the response
<i>headers</i>	an array of header fields
<i>headersCount</i>	the count of items in the headers array

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.40.2.13** `override void Coherent.UI.ViewListener.OnGetAuthCredentials ( bool isProxy, string host, uint port, string realm, string scheme ) [inline], [virtual]`

Called when a view requires authentication credentials.

Parameters

<i>isProxy</i>	whether the request came from a server or a proxy
<i>host</i>	the host which triggered the request
<i>port</i>	the port at which the request was triggered
<i>realm</i>	realm of the authentication challenge. Encoded in UTF-8
<i>scheme</i>	the authentication scheme used, e.g. "basic" or "digest". Encoded in ASCII

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.40.2.14** `override void Coherent.UI.ViewListener.OnIMEShouldCancelComposition ( ) [inline], [virtual]`

Called when the user must cancel the IME composition due to an event in the

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.40.2.15** `override void Coherent.UI.ViewListener.OnJavaScriptMessage ( string message, string defaultPrompt, string frameUrl, int messageType ) [inline], [virtual]`

Called when the view triggered a javascript message box, i.e. an alert, confirmation dialog or a prompt dialog.

## Parameters

<i>message</i>	the JavaScript message
<i>defaultPrompt</i>	the default value of the prompt text box, in case the message type is prompt
<i>frameUrl</i>	the URL which created the message
<i>messageType</i>	the type of the message (alert/confirm/prompt)

Reimplemented from [Coherent.UI.ViewListenerBase](#).

#### 8.40.2.16 `override void Coherent.UI.ViewListener.OnNavigateTo ( string path ) [inline],[virtual]`

Called when the view starts navigation to a new path.

## Parameters

<i>path</i>	URL that the view is navigating to
-------------	------------------------------------

Reimplemented from [Coherent.UI.ViewListenerBase](#).

#### 8.40.2.17 `override void Coherent.UI.ViewListener.OnReadyForBindings ( int frameId, string path, bool isMainFrame ) [inline],[virtual]`

Called when a frame is ready for bindings.

## Parameters

<i>frameId</i>	the id of the frame
<i>path</i>	the path in the frame
<i>isMainFrame</i>	true if this is the main frame of the view

Reimplemented from [Coherent.UI.ViewListenerBase](#).

#### 8.40.2.18 `override void Coherent.UI.ViewListener.OnRequestMediaStream ( MediaStreamRequest request ) [inline],[virtual]`

Called when the view requests access to a media stream. Media streams are the audio capture (microphone) and video capture (camera) devices on the system.

## Parameters

<i>request</i>	contains the available media streams for the request
----------------	--

Reimplemented from [Coherent.UI.ViewListenerBase](#).

#### 8.40.2.19 `override void Coherent.UI.ViewListener.OnScriptMessage ( ViewListenerBase.MessageLevel level, string message, string sourceId, int line ) [inline],[virtual]`

Called when a message is sent from a script running in this specific

## Parameters

<i>level</i>	message level
<i>message</i>	the text of the message
<i>sourceId</i>	id of the script (usually file name)
<i>line</i>	the number of the line in which the message was sent

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.40.2.20** `override void Coherent.UI.ViewListener.OnStartLoading ( ) [inline],[virtual]`

Called when a new path has started loading.

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.40.2.21** `override void Coherent.UI.ViewListener.OnStopLoading ( ) [inline],[virtual]`

Called when all load operations have completed.

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.40.2.22** `override void Coherent.UI.ViewListener.OnTextInputTypeChanged ( TextInputType type, bool canComposeInline ) [inline],[virtual]`

Called when the current text input control changes (i.e. the user click an edit-box). Use this method to decide when to allow for IME input. the method will be called ONLY if IME is activated on this [View](#). [View::IMEActivate](#)

Parameters

<i>type</i>	the type of the currently focused text input control by the user
<i>canComposeInline</i>	if the IME composition could be performed in-line in the control

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.40.2.23** `override void Coherent.UI.ViewListener.OnURLRequest ( string url, URLResponse response ) [inline],[virtual]`

Called before an URL request is made. The default implementation allows all requests.

Parameters

<i>url</i>	the request URL
<i>response</i>	response whether to allow the request and may be redirect it. This pointer will be valid only for this call

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.40.2.24** `override void Coherent.UI.ViewListener.OnViewCreated ( View view ) [inline],[virtual]`

Called when the requested

Parameters

<i>view</i>	the instance of the view containing all manipulation methods
-------------	--

Reimplemented from [Coherent.UI.ViewListenerBase](#).

**8.40.2.25** `override void Coherent.UI.ViewListener.Release ( ) [inline],[virtual]`

Called when the listener is no longer needed by the [UI](#) system.

Reimplemented from [Coherent.UI.ViewListenerBase](#).

## 8.40.3 Event Documentation

#### 8.40.3.1 CoherentUI\_OnBindingsReleased Coherent.UI.ViewListener.BindingsReleased

Fired when the bindings for frame are released.

#### 8.40.3.2 CoherentUI\_OnCallback Coherent.UI.ViewListener.Callback

Fired by the [UI](#) when there is no registered handler for this event.

#### 8.40.3.3 CoherentUI\_OnCaretRectChanged Coherent.UI.ViewListener.CaretRectChanged

Fired when the caret changes during IME composition. You can use this method to correctly position a custom IME control & candidate list.

#### 8.40.3.4 CoherentUI\_OnCertificateError Coherent.UI.ViewListener.CertificateError

Fired when there is an error with the certificate of a particular URL. the certificate and response pointers are valid only for the duration of this call

#### 8.40.3.5 CoherentUI\_OnCursorChanged Coherent.UI.ViewListener.CursorChanged

Fired when the cursor has changed internally in the

#### 8.40.3.6 CoherentUI\_OnDraw Coherent.UI.ViewListener.Draw

Fired when a new surface has been drawn and is ready to use by the client.

#### 8.40.3.7 CoherentUI\_OnError Coherent.UI.ViewListener.Error

Fired when an error occurs for this specific

#### 8.40.3.8 CoherentUI\_OnFailLoad Coherent.UI.ViewListener.FailLoad

Fired when a frame has been failed loading.

#### 8.40.3.9 CoherentUI\_OnFileSelectRequest Coherent.UI.ViewListener.FileSelectRequest

Fired when the view requests file selection. It could be either single file, directory or multiple files.

#### 8.40.3.10 CoherentUI\_OnFinishLoad Coherent.UI.ViewListener.FinishLoad

Fired when a frame has been successfully loaded.

#### 8.40.3.11 CoherentUI\_OnGetAuthCredentials Coherent.UI.ViewListener.GetAuthCredentials

Fired when a view requires authentication credentials.

#### 8.40.3.12 CoherentUI\_OnIMEShouldCancelComposition Coherent.UI.ViewListener.IMEShouldCancelComposition

Fired when the user must cancel the IME composition due to an event in the



**8.40.3.13 CoherentUI\_OnJavaScriptMessage Coherent.UI.ViewListener.JavaScriptMessage**

Fired when the view triggered a javascript message box, i.e. an alert, confirmation dialog or a prompt dialog.

**8.40.3.14 CoherentUI\_OnNavigateTo Coherent.UI.ViewListener.NavigateTo**

Fired when the view starts navigation to a new path.

**8.40.3.15 CoherentUI\_OnReadyForBindings Coherent.UI.ViewListener.ReadyForBindings**

Fired when a frame is ready for bindings.

**8.40.3.16 CoherentUI\_OnRequestMediaStream Coherent.UI.ViewListener.RequestMediaStream**

Fired when the view requests access to a media stream. Media streams are the audio capture (microphone) and video capture (camera) devices on the system.

**8.40.3.17 CoherentUI\_OnScriptMessage Coherent.UI.ViewListener.ScriptMessage**

Fired when a message is sent from a script running in this specific

**8.40.3.18 CoherentUI\_OnStartLoading Coherent.UI.ViewListener.StartLoading**

Fired when a new path has started loading.

**8.40.3.19 CoherentUI\_OnStopLoading Coherent.UI.ViewListener.StopLoading**

Fired when all load operations have completed.

**8.40.3.20 CoherentUI\_OnTextInputTypeChanged Coherent.UI.ViewListener.TextInputTypeChanged**

Fired when the current text input control changes (i.e. the user click an edit-box). Use this method to decide when to allow for IME input. the method will be called ONLY if IME is activated on this [View](#). [View::IMEActivate](#)

**8.40.3.21 CoherentUI\_OnURLRequest Coherent.UI.ViewListener.URLRequest**

Fired before an URL request is made. The default implementation allows all requests.

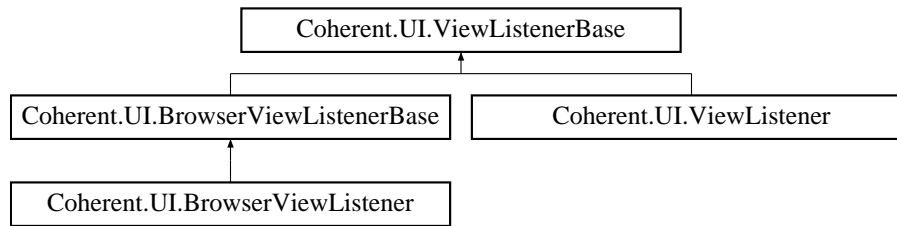
**8.40.3.22 CoherentUI\_OnViewCreated Coherent.UI.ViewListener.ViewCreated**

Fired when the requested

## 8.41 Coherent.UI.ViewListenerBase Class Reference

Interface all view listeners inherit. For an easier to use interface inherit instead - [Coherent::UI::ViewListener](#)

Inheritance diagram for Coherent.UI.ViewListenerBase:



## Public Types

- enum [MessageLevel](#)  
*Levels of script messages.*

### 8.41.1 Detailed Description

Interface all view listeners inherit. For an easier to use interface inherit instead - [Coherent::UI::ViewListener](#)

### 8.41.2 Member Enumeration Documentation

#### 8.41.2.1 enum `Coherent.UI.ViewListenerBase.MessageLevel`

Levels of script messages.

# Index

- All
  - Coherent::UI::Binding, [44](#)
- AllowCookies
  - Coherent::UI::SystemSettingsBase, [79](#)
- AppendExtraHeaderValue
  - Coherent::UI::URLRequest, [84](#)
- Arguments
  - Coherent::UI::Binding::CallbackArguments, [60](#)
- Array
  - Coherent::UI::Binding, [44](#)
- Assert
  - Coherent::UI::ILogHandler, [69](#)
- AuthCredentialsResponse
  - Coherent::UI::View, [98](#)
- BindCall
  - Coherent::UI::View, [98](#)
- BindMethods
  - Coherent::UI::Binding::BoundObject, [45](#)
- BindingsReleased
  - Coherent::UI::BrowserViewListener, [57](#)
  - Coherent::UI::ViewListener, [117](#)
- Boolean
  - Coherent::UI::Binding, [44](#)
- Button
  - Coherent::UI::MouseEventData, [73](#)
- Byte
  - Coherent::UI::Binding, [44](#)
- CachePath
  - Coherent::UI::SystemSettings, [78](#)
- Callback
  - Coherent::UI::BrowserViewListener, [57](#)
  - Coherent::UI::ViewListener, [117](#)
- Cancel
  - Coherent::UI::Download, [64](#)
- CaretRectChanged
  - Coherent::UI::BrowserViewListener, [57](#)
  - Coherent::UI::ViewListener, [117](#)
- CertificateError
  - Coherent::UI::BrowserViewListener, [57](#)
  - Coherent::UI::ViewListener, [118](#)
- CertificateStatus
  - Coherent::UI, [41](#)
- ClickThroughAlphaThreshold
  - Coherent::UI::ViewInfo, [109](#)
- Coherent, [39](#)
- Coherent.UI, [39](#)
- Coherent.UI.Binding, [43](#)
- Coherent.UI.Binding.BoundObject, [45](#)
- Coherent.UI.Binding.CallbackArguments, [59](#)
- Coherent.UI.Binding.CoherentProperty, [63](#)
- Coherent.UI.Binding.CoherentType, [63](#)
- Coherent.UI.Binding.InvalidValueCastException, [70](#)
- Coherent.UI.Binding.UnsupportedPrimitiveTypeException, [83](#)
- Coherent.UI.Binding.Value, [86](#)
- Coherent.UI.Binding.ValueObject, [95](#)
- Coherent.UI.BoundEventHandle, [45](#)
- Coherent.UI.BrowserView, [46](#)
- Coherent.UI.BrowserViewListener, [47](#)
- Coherent.UI.BrowserViewListenerBase, [59](#)
- Coherent.UI.Certificate, [60](#)
- Coherent.UI.CertificatePrincipal, [61](#)
- Coherent.UI.ChildViewInfo, [62](#)
- Coherent.UI.Download, [64](#)
- Coherent.UI.EventListener, [66](#)
- Coherent.UI.EventModifiersState, [67](#)
- Coherent.UI.EventMouseModifiersState, [67](#)
- Coherent.UI.FileHandler, [67](#)
- Coherent.UI.HTTPHeader, [68](#)
- Coherent.UI.ILogHandler, [69](#)
- Coherent.UI.ImageData, [69](#)
- Coherent.UI.KeyEventData, [70](#)
- Coherent.UI.MediaStreamDevice, [71](#)
- Coherent.UI.MediaStreamRequest, [72](#)
- Coherent.UI.MouseEventData, [73](#)
- Coherent.UI.ResourceData, [74](#)
- Coherent.UI.ResourceResponse, [75](#)
- Coherent.UI.SurfaceResponse, [76](#)
- Coherent.UI.SystemError, [76](#)
- Coherent.UI.SystemSettings, [77](#)
- Coherent.UI.SystemSettingsBase, [78](#)
- Coherent.UI.TouchEventData, [80](#)
- Coherent.UI.UISystem, [81](#)
- Coherent.UI.URLRequest, [83](#)
- Coherent.UI.View, [95](#)
- Coherent.UI.ViewError, [107](#)
- Coherent.UI.ViewInfo, [108](#)
- Coherent.UI.ViewListener, [110](#)
- Coherent.UI.ViewListenerBase, [119](#)
- Coherent::UI
  - DET\_DownloadOperationFailed, [42](#)
  - DET\_Success, [42](#)
  - SCE\_ArgumentType, [42](#)
  - SCE\_NoResult, [42](#)
  - SCE\_NoSuchMethod, [42](#)
  - SCE\_Success, [42](#)
  - SE\_InternalError, [42](#)

- SE\_InvalidKey, [42](#)
- SE\_MissingComponent, [42](#)
- SE\_Success, [42](#)
- VE\_AbnormalTermination, [43](#)
- VE\_ChildCompositionFailed, [43](#)
- VE\_ClickThroughFailed, [43](#)
- VE\_FailedCreation, [43](#)
- VE\_FailedRenderingCreation, [43](#)
- VE\_FrameAlreadyInFlight, [43](#)
- VE\_FrameNotReady, [43](#)
- VE\_FrameNotRequested, [43](#)
- VE\_InvalidCall, [43](#)
- VE\_InvalidPath, [43](#)
- VE\_QueryNotReady, [43](#)
- VE\_Success, [43](#)
- Coherent::UI::Binding
  - All, [44](#)
  - Array, [44](#)
  - Boolean, [44](#)
  - Byte, [44](#)
  - Explicit, [44](#)
  - Instance, [44](#)
  - Integer, [44](#)
  - Null, [44](#)
  - Number, [44](#)
  - Object, [44](#)
  - SByte, [44](#)
  - Static, [44](#)
  - String, [44](#)
  - UInteger, [44](#)
- Coherent::UI
  - CertificateStatus, [41](#)
  - CursorTypes, [41](#)
  - Direct3DInterfaceType, [41](#)
  - DownloadErrorType, [41](#)
  - FileSelectionMode, [42](#)
  - JavaScriptMessageType, [42](#)
  - MediaStreamType, [42](#)
  - RenderingDeviceType, [42](#)
  - ScriptCallErrorType, [42](#)
  - SystemErrorType, [42](#)
  - TextInputControlType, [42](#)
  - UserAgentType, [42](#)
  - ViewErrorType, [42](#)
  - ViewType, [43](#)
- Coherent::UI::Binding
  - PropertyBindingFlags, [44](#)
  - ValueType, [44](#)
- Coherent::UI::Binding::BoundObject
  - BindMethods, [45](#)
- Coherent::UI::Binding::CallbackArguments
  - Arguments, [60](#)
- Coherent::UI::Binding::CoherentProperty
  - CoherentProperty, [63](#)
- Coherent::UI::Binding::Value
  - Equals, [90](#)
  - GetHashCode, [90](#)
  - operator bool, [90](#)
  - operator byte, [90](#)
  - operator Dictionary< string, Value >, [91](#)
  - operator double, [91](#)
  - operator float, [91](#)
  - operator int, [91](#)
  - operator sbyte, [91](#)
  - operator string, [93](#)
  - operator uint, [93](#)
  - operator==, [93](#)
  - Type, [95](#)
  - Value, [87](#), [88](#), [90](#)
- Coherent::UI::BrowserView
  - GetHistory, [46](#)
  - GetType, [46](#)
  - GoBack, [46](#)
  - GoForward, [46](#)
- Coherent::UI::BrowserViewListener
  - BindingsReleased, [57](#)
  - Callback, [57](#)
  - CaretRectChanged, [57](#)
  - CertificateError, [57](#)
  - CreateSurface, [49](#)
  - CursorChanged, [57](#)
  - DestroySurface, [49](#)
  - Draw, [58](#)
  - Error, [58](#)
  - FailLoad, [58](#)
  - FileSelectRequest, [58](#)
  - FinishLoad, [58](#)
  - GetAuthCredentials, [58](#)
  - IMEShouldCancelComposition, [58](#)
  - JavaScriptMessage, [58](#)
  - NavigateTo, [58](#)
  - OnBindingsReleased, [51](#)
  - OnCallback, [51](#)
  - OnCaretRectChanged, [51](#)
  - OnCertificateError, [51](#)
  - OnCursorChanged, [52](#)
  - OnDraw, [52](#)
  - OnError, [52](#)
  - OnFailLoad, [52](#)
  - OnFileSelectRequest, [52](#)
  - OnFinishLoad, [53](#)
  - OnGetAuthCredentials, [53](#)
  - OnIMEShouldCancelComposition, [53](#)
  - OnJavaScriptMessage, [53](#)
  - OnNavigateTo, [53](#)
  - OnReadyForBindings, [55](#)
  - OnRequestMediaStream, [55](#)
  - OnScriptMessage, [55](#)
  - OnStartLoading, [55](#)
  - OnStopLoading, [55](#)
  - OnTextInputTypeChanged, [55](#)
  - OnURLRequest, [57](#)
  - OnViewCreated, [57](#)
  - ReadyForBindings, [58](#)
  - Release, [57](#)
  - RequestMediaStream, [58](#)

- ScriptMessage, [58](#)
- StartLoading, [59](#)
- StopLoading, [59](#)
- TextInputTypeChanged, [59](#)
- URLRequest, [59](#)
- ViewCreated, [59](#)
- Coherent::UI::Certificate
  - Fingerprint, [60](#)
  - FingerprintCA, [60](#)
  - Issuer, [61](#)
  - SerialNumber, [61](#)
  - Subject, [61](#)
  - ValidEnd, [61](#)
  - ValidStart, [61](#)
- Coherent::UI::CertificatePrincipal
  - CommonName, [62](#)
  - CountryName, [62](#)
  - DomainComponents, [62](#)
  - LocalityName, [62](#)
  - OrganizationNames, [62](#)
  - OrganizationUnitNames, [62](#)
  - StateOrProvinceName, [62](#)
  - StreetAddresses, [62](#)
- Coherent::UI::ChildViewInfo
  - Listener, [63](#)
  - NewViewInfo, [63](#)
- Coherent::UI::Download
  - Cancel, [64](#)
  - Destroy, [64](#)
  - GetDownloadedBytesSoFar, [65](#)
  - GetFilename, [65](#)
  - GetId, [65](#)
  - GetReferrer, [65](#)
  - GetTotalBytes, [65](#)
  - GetURL, [65](#)
  - HasFailed, [65](#)
  - IsCancelled, [66](#)
  - IsComplete, [66](#)
- Coherent::UI::EventListener
  - OnDownloadComplete, [66](#)
  - OnDownloadFailed, [66](#)
  - OnDownloadStarted, [67](#)
  - OnError, [67](#)
  - SystemReady, [67](#)
- Coherent::UI::FileHandler
  - ReadFile, [68](#)
  - WriteFile, [68](#)
- Coherent::UI::HTTPHeader
  - Content, [68](#)
  - Field, [68](#)
- Coherent::UI::ILogHandler
  - Assert, [69](#)
  - WriteLog, [69](#)
- Coherent::UI::ImageData
  - Destroy, [70](#)
  - Update, [70](#)
- Coherent::UI::KeyEventData
  - IsAutoRepeat, [71](#)
  - IsNumPad, [71](#)
  - KeyCode, [71](#)
  - Modifiers, [71](#)
  - Type, [71](#)
- Coherent::UI::MediaStreamDevice
  - DeviceId, [72](#)
  - Name, [72](#)
  - Type, [72](#)
- Coherent::UI::MediaStreamRequest
  - Devices, [72](#)
  - Respond, [72](#)
- Coherent::UI::MouseEventData
  - Button, [73](#)
  - Modifiers, [73](#)
  - MouseModifiers, [73](#)
  - Type, [73](#)
  - WheelX, [73](#)
  - WheelY, [73](#)
  - X, [74](#)
  - Y, [74](#)
- Coherent::UI::ResourceData
  - GetBuffer, [74](#)
  - GetSize, [74](#)
  - SignalFailure, [74](#)
  - SignalSuccess, [75](#)
- Coherent::UI::ResourceResponse
  - GetBuffer, [75](#)
  - SignalFailure, [75](#)
  - SignalSuccess, [75](#)
- Coherent::UI::SurfaceResponse
  - Signal, [76](#)
- Coherent::UI::SystemError
  - Error, [77](#)
  - ErrorCode, [77](#)
- Coherent::UI::SystemSettings
  - CachePath, [78](#)
  - CookiesResource, [78](#)
  - CustomUserAgentString, [78](#)
  - D3DObject, [78](#)
  - FocusWindow, [78](#)
  - HTML5LocalStoragePath, [78](#)
  - HostDirectory, [78](#)
  - MinidumpFolder, [78](#)
- Coherent::UI::SystemSettingsBase
  - AllowCookies, [79](#)
  - CustomUserAgentString, [79](#)
  - D3DInterfaceType, [79](#)
  - DebuggerPort, [79](#)
  - DisableWebSecurity, [79](#)
  - EnableProxy, [80](#)
  - EnableSupportForProprietaryCodecs, [80](#)
  - ForceDisablePluginFullscreen, [80](#)
  - UserAgent, [80](#)
  - WriteMinidumps, [80](#)
- Coherent::UI::TouchEventData
  - HasRadiusInformation, [81](#)
  - Id, [81](#)
  - RadiusX, [81](#)

- RadiusY, [81](#)
- Type, [81](#)
- X, [81](#)
- Y, [81](#)
- Coherent::UI::UISystem
  - CreateView, [82](#)
  - FetchSurfaces, [82](#)
  - Uninitialize, [82](#)
  - Update, [83](#)
- Coherent::UI::URLRequest
  - AppendExtraHeaderValue, [84](#)
  - Deny, [84](#)
  - GetExtraHeader, [84](#)
  - GetExtraHeaders, [84](#)
  - GetMethod, [84](#)
  - GetReferrer, [84](#)
  - GetURL, [85](#)
  - RemoveExtraHeader, [85](#)
  - SetExtraHeader, [85](#)
  - SetExtraHeaders, [85](#)
  - SetMethod, [85](#)
  - SetReferrer, [85](#)
  - SetURL, [86](#)
- Coherent::UI::View
  - AuthCredentialsResponse, [98](#)
  - BindCall, [98](#)
  - CreateImageData, [98](#)
  - Destroy, [98](#)
  - DownloadUrl, [98](#)
  - DrawSurface, [98](#)
  - ExecuteScript, [99](#)
  - FetchMouseOnUIQuery, [99](#)
  - FetchSurface, [99](#)
  - GetClickThroughAlphaThreshold, [99](#)
  - GetCurrentViewPath, [99](#)
  - GetHeight, [99](#)
  - GetLastFailedPath, [99](#)
  - GetLastLoadedPath, [100](#)
  - GetLastRequestedPath, [100](#)
  - GetTargetFramerate, [100](#)
  - GetViewType, [100](#)
  - GetWidth, [100](#)
  - HasMouseQueryFinished, [100](#)
  - IMEActivate, [101](#)
  - IMECancelComposition, [101](#)
  - IMEConfirmComposition, [101](#)
  - IMEIsActive, [101](#)
  - IMESetComposition, [101](#)
  - InterceptURLRequests, [102](#)
  - IsMouseOnView, [102](#)
  - IsOnDemand, [102](#)
  - IsTransparent, [102](#)
  - IssueMouseOnUIQuery, [102](#)
  - KeyEvent, [103](#)
  - KillFocus, [103](#)
  - Load, [103](#)
  - MouseActivate, [103](#)
  - MouseEvent, [103](#)
  - PeekFrameReady, [103](#)
  - Redraw, [103](#)
  - RegisterForEvent, [103](#)
  - Reload, [105](#)
  - ReplyToJavaScriptMessage, [105](#)
  - RequestFrame, [105](#)
  - Resize, [105](#)
  - SetClickThroughAlphaThreshold, [106](#)
  - SetFocus, [106](#)
  - SetScriptError, [106](#)
  - SetTargetFramerate, [106](#)
  - SupportsClickThrough, [106](#)
  - TouchEvent, [106](#)
  - TriggerEvent, [107](#)
  - UnbindCall, [107](#)
  - UnbindObject, [107](#)
  - UnregisterFromEvent, [107](#)
  - UsesSharedMemory, [107](#)
- Coherent::UI::ViewError
  - Error, [108](#)
  - ErrorCode, [108](#)
- Coherent::UI::ViewInfo
  - ClickThroughAlphaThreshold, [109](#)
  - ControlTimeOnDemand, [109](#)
  - ForceSoftwareRendering, [109](#)
  - Height, [109](#)
  - IsOnDemand, [109](#)
  - IsTransparent, [109](#)
  - SupportClickThrough, [109](#)
  - TargetFrameRate, [109](#)
  - UsesSharedMemory, [109](#)
  - Width, [109](#)
- Coherent::UI::ViewListener
  - BindingsReleased, [117](#)
  - Callback, [117](#)
  - CaretRectChanged, [117](#)
  - CertificateError, [118](#)
  - CreateSurface, [112](#)
  - CursorChanged, [118](#)
  - DestroySurface, [112](#)
  - Draw, [118](#)
  - Error, [118](#)
  - FailLoad, [118](#)
  - FileSelectRequest, [118](#)
  - FinishLoad, [118](#)
  - GetAuthCredentials, [118](#)
  - IMEShouldCancelComposition, [118](#)
  - JavaScriptMessage, [118](#)
  - NavigateTo, [118](#)
  - OnBindingsReleased, [113](#)
  - OnCallback, [113](#)
  - OnCaretRectChanged, [113](#)
  - OnCertificateError, [113](#)
  - OnCursorChanged, [114](#)
  - OnDraw, [114](#)
  - OnError, [114](#)
  - OnFailLoad, [114](#)
  - OnFileSelectRequest, [114](#)

- OnFinishLoad, 115
- OnGetAuthCredentials, 115
- OnIMEShouldCancelComposition, 115
- OnJavaScriptMessage, 115
- OnNavigateTo, 116
- OnReadyForBindings, 116
- OnRequestMediaStream, 116
- OnScriptMessage, 116
- OnStartLoading, 116
- OnStopLoading, 116
- OnTextInputTypeChanged, 117
- OnURLRequest, 117
- OnViewCreated, 117
- ReadyForBindings, 118
- Release, 117
- RequestMediaStream, 119
- ScriptMessage, 119
- StartLoading, 119
- StopLoading, 119
- TextInputTypeChanged, 119
- URLRequest, 119
- ViewCreated, 119
- Coherent::UI::ViewListenerBase
  - MessageLevel, 120
- CoherentProperty
  - Coherent::UI::Binding::CoherentProperty, 63
- CommonName
  - Coherent::UI::CertificatePrincipal, 62
- Content
  - Coherent::UI::HTTPHeader, 68
- ControlTimeOnDemand
  - Coherent::UI::ViewInfo, 109
- CookiesResource
  - Coherent::UI::SystemSettings, 78
- CountryName
  - Coherent::UI::CertificatePrincipal, 62
- CreateImageData
  - Coherent::UI::View, 98
- CreateSurface
  - Coherent::UI::BrowserViewListener, 49
  - Coherent::UI::ViewListener, 112
- CreateView
  - Coherent::UI::UISystem, 82
- CursorChanged
  - Coherent::UI::BrowserViewListener, 57
  - Coherent::UI::ViewListener, 118
- CursorTypes
  - Coherent::UI, 41
- CustomUserAgentString
  - Coherent::UI::SystemSettings, 78
  - Coherent::UI::SystemSettingsBase, 79
- D3DInterfaceType
  - Coherent::UI::SystemSettingsBase, 79
- D3DObject
  - Coherent::UI::SystemSettings, 78
- DET\_DownloadOperationFailed
  - Coherent::UI, 42
- DET\_Success
  - Coherent::UI, 42
- DebuggerPort
  - Coherent::UI::SystemSettingsBase, 79
- Deny
  - Coherent::UI::URLRequest, 84
- Destroy
  - Coherent::UI::Download, 64
  - Coherent::UI::ImageData, 70
  - Coherent::UI::View, 98
- DestroySurface
  - Coherent::UI::BrowserViewListener, 49
  - Coherent::UI::ViewListener, 112
- DeviceId
  - Coherent::UI::MediaStreamDevice, 72
- Devices
  - Coherent::UI::MediaStreamRequest, 72
- Direct3DInterfaceType
  - Coherent::UI, 41
- DisableWebSecurity
  - Coherent::UI::SystemSettingsBase, 79
- DomainComponents
  - Coherent::UI::CertificatePrincipal, 62
- DownloadErrorType
  - Coherent::UI, 41
- DownloadUrl
  - Coherent::UI::View, 98
- Draw
  - Coherent::UI::BrowserViewListener, 58
  - Coherent::UI::ViewListener, 118
- DrawSurface
  - Coherent::UI::View, 98
- EnableProxy
  - Coherent::UI::SystemSettingsBase, 80
- EnableSupportForProprietaryCodecs
  - Coherent::UI::SystemSettingsBase, 80
- Equals
  - Coherent::UI::Binding::Value, 90
- Error
  - Coherent::UI::BrowserViewListener, 58
  - Coherent::UI::SystemError, 77
  - Coherent::UI::ViewError, 108
  - Coherent::UI::ViewListener, 118
- ErrorCode
  - Coherent::UI::SystemError, 77
  - Coherent::UI::ViewError, 108
- ExecuteScript
  - Coherent::UI::View, 99
- Explicit
  - Coherent::UI::Binding, 44
- FailLoad
  - Coherent::UI::BrowserViewListener, 58
  - Coherent::UI::ViewListener, 118
- FetchMouseOnUIQuery
  - Coherent::UI::View, 99
- FetchSurface
  - Coherent::UI::View, 99
- FetchSurfaces

- Coherent::UI::UISystem, 82
- Field
  - Coherent::UI::HTTPHeader, 68
- FileSelectRequest
  - Coherent::UI::BrowserViewListener, 58
  - Coherent::UI::ViewListener, 118
- FileSelectionMode
  - Coherent::UI, 42
- Fingerprint
  - Coherent::UI::Certificate, 60
- FingerprintCA
  - Coherent::UI::Certificate, 60
- FinishLoad
  - Coherent::UI::BrowserViewListener, 58
  - Coherent::UI::ViewListener, 118
- FocusWindow
  - Coherent::UI::SystemSettings, 78
- ForceDisablePluginFullscreen
  - Coherent::UI::SystemSettingsBase, 80
- ForceSoftwareRendering
  - Coherent::UI::ViewInfo, 109
- GetAuthCredentials
  - Coherent::UI::BrowserViewListener, 58
  - Coherent::UI::ViewListener, 118
- GetBuffer
  - Coherent::UI::ResourceData, 74
  - Coherent::UI::ResourceResponse, 75
- GetClickThroughAlphaThreshold
  - Coherent::UI::View, 99
- GetCurrentViewPath
  - Coherent::UI::View, 99
- GetDownloadedBytesSoFar
  - Coherent::UI::Download, 65
- GetExtraHeader
  - Coherent::UI::URLRequest, 84
- GetExtraHeaders
  - Coherent::UI::URLRequest, 84
- GetFilename
  - Coherent::UI::Download, 65
- GetHashCode
  - Coherent::UI::Binding::Value, 90
- GetHeight
  - Coherent::UI::View, 99
- GetHistory
  - Coherent::UI::BrowserView, 46
- GetId
  - Coherent::UI::Download, 65
- GetLastFailedPath
  - Coherent::UI::View, 99
- GetLastLoadedPath
  - Coherent::UI::View, 100
- GetLastRequestedPath
  - Coherent::UI::View, 100
- GetMethod
  - Coherent::UI::URLRequest, 84
- GetReferrer
  - Coherent::UI::Download, 65
  - Coherent::UI::URLRequest, 84
- GetSize
  - Coherent::UI::ResourceData, 74
- GetTargetFramerate
  - Coherent::UI::View, 100
- GetTotalBytes
  - Coherent::UI::Download, 65
- GetURL
  - Coherent::UI::Download, 65
  - Coherent::UI::URLRequest, 85
- GetViewType
  - Coherent::UI::BrowserView, 46
  - Coherent::UI::View, 100
- GetWidth
  - Coherent::UI::View, 100
- GoBack
  - Coherent::UI::BrowserView, 46
- GoForward
  - Coherent::UI::BrowserView, 46
- HTML5LocalStoragePath
  - Coherent::UI::SystemSettings, 78
- HasFailed
  - Coherent::UI::Download, 65
- HasMouseQueryFinished
  - Coherent::UI::View, 100
- HasRadiusInformation
  - Coherent::UI::TouchEventData, 81
- Height
  - Coherent::UI::ViewInfo, 109
- HostDirectory
  - Coherent::UI::SystemSettings, 78
- IMEActivate
  - Coherent::UI::View, 101
- IMECancelComposition
  - Coherent::UI::View, 101
- IMEConfirmComposition
  - Coherent::UI::View, 101
- IMEIsActive
  - Coherent::UI::View, 101
- IMESetComposition
  - Coherent::UI::View, 101
- IMEShouldCancelComposition
  - Coherent::UI::BrowserViewListener, 58
  - Coherent::UI::ViewListener, 118
- Id
  - Coherent::UI::TouchEventData, 81
- Instance
  - Coherent::UI::Binding, 44
- Integer
  - Coherent::UI::Binding, 44
- InterceptURLRequests
  - Coherent::UI::View, 102
- IsAutoRepeat
  - Coherent::UI::KeyEventData, 71
- IsCancelled
  - Coherent::UI::Download, 66
- IsComplete
  - Coherent::UI::Download, 66



- IsMouseOnView
  - Coherent::UI::View, [102](#)
- IsNumPad
  - Coherent::UI::KeyEventData, [71](#)
- IsOnDemand
  - Coherent::UI::View, [102](#)
  - Coherent::UI::ViewInfo, [109](#)
- IsTransparent
  - Coherent::UI::View, [102](#)
  - Coherent::UI::ViewInfo, [109](#)
- IssueMouseOnUIQuery
  - Coherent::UI::View, [102](#)
- Issuer
  - Coherent::UI::Certificate, [61](#)
- JavaScriptMessage
  - Coherent::UI::BrowserViewListener, [58](#)
  - Coherent::UI::ViewListener, [118](#)
- JavaScriptMessageType
  - Coherent::UI, [42](#)
- KeyCode
  - Coherent::UI::KeyEventData, [71](#)
- KeyEvent
  - Coherent::UI::View, [103](#)
- KillFocus
  - Coherent::UI::View, [103](#)
- Listener
  - Coherent::UI::ChildViewInfo, [63](#)
- Load
  - Coherent::UI::View, [103](#)
- LocalityName
  - Coherent::UI::CertificatePrincipal, [62](#)
- MediaStreamType
  - Coherent::UI, [42](#)
- MessageLevel
  - Coherent::UI::ViewListenerBase, [120](#)
- MinidumpFolder
  - Coherent::UI::SystemSettings, [78](#)
- Modifiers
  - Coherent::UI::KeyEventData, [71](#)
  - Coherent::UI::MouseEventData, [73](#)
- MouseActivate
  - Coherent::UI::View, [103](#)
- MouseEvent
  - Coherent::UI::View, [103](#)
- MouseModifiers
  - Coherent::UI::MouseEventData, [73](#)
- Name
  - Coherent::UI::MediaStreamDevice, [72](#)
- NavigateTo
  - Coherent::UI::BrowserViewListener, [58](#)
  - Coherent::UI::ViewListener, [118](#)
- NewViewInfo
  - Coherent::UI::ChildViewInfo, [63](#)
- Null
  - Coherent::UI::Binding, [44](#)
- Number
  - Coherent::UI::Binding, [44](#)
- Object
  - Coherent::UI::Binding, [44](#)
- OnBindingsReleased
  - Coherent::UI::BrowserViewListener, [51](#)
  - Coherent::UI::ViewListener, [113](#)
- OnCallback
  - Coherent::UI::BrowserViewListener, [51](#)
  - Coherent::UI::ViewListener, [113](#)
- OnCaretRectChanged
  - Coherent::UI::BrowserViewListener, [51](#)
  - Coherent::UI::ViewListener, [113](#)
- OnCertificateError
  - Coherent::UI::BrowserViewListener, [51](#)
  - Coherent::UI::ViewListener, [113](#)
- OnCursorChanged
  - Coherent::UI::BrowserViewListener, [52](#)
  - Coherent::UI::ViewListener, [114](#)
- OnDownloadComplete
  - Coherent::UI::EventListener, [66](#)
- OnDownloadFailed
  - Coherent::UI::EventListener, [66](#)
- OnDownloadStarted
  - Coherent::UI::EventListener, [67](#)
- OnDraw
  - Coherent::UI::BrowserViewListener, [52](#)
  - Coherent::UI::ViewListener, [114](#)
- OnError
  - Coherent::UI::BrowserViewListener, [52](#)
  - Coherent::UI::EventListener, [67](#)
  - Coherent::UI::ViewListener, [114](#)
- OnFailLoad
  - Coherent::UI::BrowserViewListener, [52](#)
  - Coherent::UI::ViewListener, [114](#)
- OnFileSelectRequest
  - Coherent::UI::BrowserViewListener, [52](#)
  - Coherent::UI::ViewListener, [114](#)
- OnFinishLoad
  - Coherent::UI::BrowserViewListener, [53](#)
  - Coherent::UI::ViewListener, [115](#)
- OnGetAuthCredentials
  - Coherent::UI::BrowserViewListener, [53](#)
  - Coherent::UI::ViewListener, [115](#)
- OnIMEShouldCancelComposition
  - Coherent::UI::BrowserViewListener, [53](#)
  - Coherent::UI::ViewListener, [115](#)
- OnJavaScriptMessage
  - Coherent::UI::BrowserViewListener, [53](#)
  - Coherent::UI::ViewListener, [115](#)
- OnNavigateTo
  - Coherent::UI::BrowserViewListener, [53](#)
  - Coherent::UI::ViewListener, [116](#)
- OnReadyForBindings
  - Coherent::UI::BrowserViewListener, [55](#)
  - Coherent::UI::ViewListener, [116](#)
- OnRequestMediaStream

- Coherent::UI::BrowserViewListener, 55
- Coherent::UI::ViewListener, 116
- OnScriptMessage
  - Coherent::UI::BrowserViewListener, 55
  - Coherent::UI::ViewListener, 116
- OnStartLoading
  - Coherent::UI::BrowserViewListener, 55
  - Coherent::UI::ViewListener, 116
- OnStopLoading
  - Coherent::UI::BrowserViewListener, 55
  - Coherent::UI::ViewListener, 116
- OnTextInputTypeChanged
  - Coherent::UI::BrowserViewListener, 55
  - Coherent::UI::ViewListener, 117
- OnURLRequest
  - Coherent::UI::BrowserViewListener, 57
  - Coherent::UI::ViewListener, 117
- OnViewCreated
  - Coherent::UI::BrowserViewListener, 57
  - Coherent::UI::ViewListener, 117
- operator bool
  - Coherent::UI::Binding::Value, 90
- operator byte
  - Coherent::UI::Binding::Value, 90
- operator Dictionary< string, Value >
  - Coherent::UI::Binding::Value, 91
- operator double
  - Coherent::UI::Binding::Value, 91
- operator float
  - Coherent::UI::Binding::Value, 91
- operator int
  - Coherent::UI::Binding::Value, 91
- operator sbyte
  - Coherent::UI::Binding::Value, 91
- operator string
  - Coherent::UI::Binding::Value, 93
- operator uint
  - Coherent::UI::Binding::Value, 93
- operator==
  - Coherent::UI::Binding::Value, 93
- OrganizationNames
  - Coherent::UI::CertificatePrincipal, 62
- OrganizationUnitNames
  - Coherent::UI::CertificatePrincipal, 62
- PeekFrameReady
  - Coherent::UI::View, 103
- PropertyBindingFlags
  - Coherent::UI::Binding, 44
- RadiusX
  - Coherent::UI::TouchEventData, 81
- RadiusY
  - Coherent::UI::TouchEventData, 81
- ReadFile
  - Coherent::UI::FileHandler, 68
- ReadyForBindings
  - Coherent::UI::BrowserViewListener, 58
  - Coherent::UI::ViewListener, 118
- Redraw
  - Coherent::UI::View, 103
- RegisterForEvent
  - Coherent::UI::View, 103
- Release
  - Coherent::UI::BrowserViewListener, 57
  - Coherent::UI::ViewListener, 117
- Reload
  - Coherent::UI::View, 105
- RemoveExtraHeader
  - Coherent::UI::URLRequest, 85
- RenderingDeviceType
  - Coherent::UI, 42
- ReplyToJavaScriptMessage
  - Coherent::UI::View, 105
- RequestFrame
  - Coherent::UI::View, 105
- RequestMediaStream
  - Coherent::UI::BrowserViewListener, 58
  - Coherent::UI::ViewListener, 119
- Resize
  - Coherent::UI::View, 105
- Respond
  - Coherent::UI::MediaStreamRequest, 72
- SByte
  - Coherent::UI::Binding, 44
- SCE\_ArgumentType
  - Coherent::UI, 42
- SCE\_NoResult
  - Coherent::UI, 42
- SCE\_NoSuchMethod
  - Coherent::UI, 42
- SCE\_Success
  - Coherent::UI, 42
- SE\_InternalError
  - Coherent::UI, 42
- SE\_InvalidKey
  - Coherent::UI, 42
- SE\_MissingComponent
  - Coherent::UI, 42
- SE\_Success
  - Coherent::UI, 42
- ScriptCallErrorType
  - Coherent::UI, 42
- ScriptMessage
  - Coherent::UI::BrowserViewListener, 58
  - Coherent::UI::ViewListener, 119
- SerialNumber
  - Coherent::UI::Certificate, 61
- SetClickThroughAlphaThreshold
  - Coherent::UI::View, 106
- SetExtraHeader
  - Coherent::UI::URLRequest, 85
- SetExtraHeaders
  - Coherent::UI::URLRequest, 85
- SetFocus
  - Coherent::UI::View, 106
- SetMethod

- Coherent::UI::URLRequest, 85
- SetReferrer
  - Coherent::UI::URLRequest, 85
- SetScriptError
  - Coherent::UI::View, 106
- SetTargetFramerate
  - Coherent::UI::View, 106
- SetURL
  - Coherent::UI::URLRequest, 86
- Signal
  - Coherent::UI::SurfaceResponse, 76
- SignalFailure
  - Coherent::UI::ResourceData, 74
  - Coherent::UI::ResourceResponse, 75
- SignalSuccess
  - Coherent::UI::ResourceData, 75
  - Coherent::UI::ResourceResponse, 75
- StartLoading
  - Coherent::UI::BrowserViewListener, 59
  - Coherent::UI::ViewListener, 119
- StateOrProvinceName
  - Coherent::UI::CertificatePrincipal, 62
- Static
  - Coherent::UI::Binding, 44
- StopLoading
  - Coherent::UI::BrowserViewListener, 59
  - Coherent::UI::ViewListener, 119
- StreetAddresses
  - Coherent::UI::CertificatePrincipal, 62
- String
  - Coherent::UI::Binding, 44
- Subject
  - Coherent::UI::Certificate, 61
- SupportClickThrough
  - Coherent::UI::ViewInfo, 109
- SupportsClickThrough
  - Coherent::UI::View, 106
- SystemErrorType
  - Coherent::UI, 42
- SystemReady
  - Coherent::UI::EventListener, 67
- TargetFrameRate
  - Coherent::UI::ViewInfo, 109
- TextInputControlType
  - Coherent::UI, 42
- TextInputTypeChanged
  - Coherent::UI::BrowserViewListener, 59
  - Coherent::UI::ViewListener, 119
- TouchEvent
  - Coherent::UI::View, 106
- TriggerEvent
  - Coherent::UI::View, 107
- Type
  - Coherent::UI::Binding::Value, 95
  - Coherent::UI::KeyEventData, 71
  - Coherent::UI::MediaStreamDevice, 72
  - Coherent::UI::MouseEventData, 73
  - Coherent::UI::TouchEventData, 81
- UInteger
  - Coherent::UI::Binding, 44
- URLRequest
  - Coherent::UI::BrowserViewListener, 59
  - Coherent::UI::ViewListener, 119
- UnbindCall
  - Coherent::UI::View, 107
- UnbindObject
  - Coherent::UI::View, 107
- Uninitialize
  - Coherent::UI::UISystem, 82
- UnregisterFromEvent
  - Coherent::UI::View, 107
- Update
  - Coherent::UI::ImageData, 70
  - Coherent::UI::UISystem, 83
- UserAgent
  - Coherent::UI::SystemSettingsBase, 80
- UserAgentType
  - Coherent::UI, 42
- UsesSharedMemory
  - Coherent::UI::View, 107
  - Coherent::UI::ViewInfo, 109
- VE\_AbnormalTermination
  - Coherent::UI, 43
- VE\_ChildCompositionFailed
  - Coherent::UI, 43
- VE\_ClickThroughFailed
  - Coherent::UI, 43
- VE\_FailedCreation
  - Coherent::UI, 43
- VE\_FailedRenderingCreation
  - Coherent::UI, 43
- VE\_FrameAlreadyInFlight
  - Coherent::UI, 43
- VE\_FrameNotReady
  - Coherent::UI, 43
- VE\_FrameNotRequested
  - Coherent::UI, 43
- VE\_InvalidCall
  - Coherent::UI, 43
- VE\_InvalidPath
  - Coherent::UI, 43
- VE\_QueryNotReady
  - Coherent::UI, 43
- VE\_Success
  - Coherent::UI, 43
- ValidEnd
  - Coherent::UI::Certificate, 61
- ValidStart
  - Coherent::UI::Certificate, 61
- Value
  - Coherent::UI::Binding::Value, 87, 88, 90
- ValueType
  - Coherent::UI::Binding, 44
- ViewCreated
  - Coherent::UI::BrowserViewListener, 59
  - Coherent::UI::ViewListener, 119

ViewErrorType  
    Coherent::UI, [42](#)  
ViewType  
    Coherent::UI, [43](#)  
  
WheelX  
    Coherent::UI::MouseEventData, [73](#)  
WheelY  
    Coherent::UI::MouseEventData, [73](#)  
Width  
    Coherent::UI::ViewInfo, [109](#)  
WriteFile  
    Coherent::UI::FileHandler, [68](#)  
WriteLog  
    Coherent::UI::ILogHandler, [69](#)  
WriteMinidumps  
    Coherent::UI::SystemSettingsBase, [80](#)  
  
X  
    Coherent::UI::MouseEventData, [74](#)  
    Coherent::UI::TouchEventData, [81](#)  
  
Y  
    Coherent::UI::MouseEventData, [74](#)  
    Coherent::UI::TouchEventData, [81](#)