# Coherent UI for Unity3D

2.5.9

# Contents


**1 Copyright** **1**

**2 Requirements** **3**

2.1 Software Requirements . . . 3

2.2 Hardware Requirements . . . 4

**3 Introduction** **5**

3.1 Brief overview of Coherent UI . . . 5

3.2 Differences between Desktop and Mobile version . . . 5

**4 Installation** **7**

4.1 Prerequisites . . . 7

4.2 Package contents and structure . . . 8

4.3 Usage of the package . . . 8

4.4 Switching between build targets . . . 10

4.5 Trial version registration . . . 10

**5 Samples** **11**

5.1 Hello, Coherent UI . . . 11

5.2 Facebook integration . . . 13

5.3 Menu And HUD . . . 14

5.4 Binding Demo . . . 16

5.5 Archive resource . . . 20

5.6 Mobile Input . . . 21

5.7 Oculus . . . 23

5.8 Mobile Surface . . . 24

5.9 Live Game Views . . . 24

5.10 IME Sample . . . 24

**6 Programmer's API** **27**

6.1 CoherentUISystem properties . . . 27

6.2 CoherentUIView properties . . . 27

6.3 CoherentUILiveGameView properties . . . 29

**7 Important points** 31
7.1 Binding . . . 31
7.2 CoherentMethod attribute for .NET scripts . . . 31
7.3 Namespaces . . . 33
7.4 Subclassing CoherentUIView and UnityViewListener . . . 33
7.5 Coherent UI system lifetime . . . 34
7.6 Customizing initialization of the Coherent UI System . . . 34
7.7 Update cycle . . . 35
7.8 Input forwarding - Desktop . . . 35
7.9 Input forwarding - Mobile . . . 35
7.10 Mobile Preview . . . 36
7.11 Custom file handler . . . 36
7.12 UI Resources . . . 37
7.13 Click-to-focus Views - Desktop only . . . 37
7.14 View Filters - Desktop only . . . 39
7.15 Hit testing - Desktop only . . . 39
7.16 Mobile Surface Views . . . 40
7.17 Logging . . . 40
7.18 Live Game Views . . . 40
7.19 Font Rendering . . . 41

**8 Unity Script Binding** 43
8.1 Sample Unity Script Binding . . . 43

**9 Coherent UI Menu** 45
9.1 Binding actions to events . . . 45
9.2 Setting up the HTML . . . 47
9.3 JavaScript API . . . 48

**10 Namespace Index** 49
10.1 Namespace List . . . 49

**11 Hierarchical Index** 51
11.1 Class Hierarchy . . . 51

**12 Class Index** 53
12.1 Class List . . . 53

**13 Namespace Documentation** 57
13.1 Package Coherent . . . 57
13.2 Package Coherent.UI . . . 57
13.2.1 Enumeration Type Documentation . . . 60

13.2.1.1 CertificateStatus . . . 60
13.2.1.2 ContextErrorType . . . 60
13.2.1.3 CursorTypes . . . 60
13.2.1.4 Direct3DInterfaceType . . . 60
13.2.1.5 DownloadErrorType . . . 60
13.2.1.6 FileSelectionMode . . . 61
13.2.1.7 JavaScriptMessageType . . . 61
13.2.1.8 MediaStreamType . . . 61
13.2.1.9 RenderingDeviceType . . . 61
13.2.1.10 ScriptCallErrorType . . . 61
13.2.1.11 SubscriptionErrorType . . . 61
13.2.1.12 TextInputControlType . . . 61
13.2.1.13 UserAgentType . . . 61
13.2.1.14 ViewErrorType . . . 61
13.2.1.15 ViewType . . . 62
13.3 Package Coherent.UI.Binding . . . 62
13.3.1 Enumeration Type Documentation . . . 63
13.3.1.1 PropertyBindingFlags . . . 63
13.3.1.2 ValueType . . . 63
13.4 engine Namespace Reference . . . 63
13.4.1 Detailed Description . . . 64
13.4.2 Function Documentation . . . 64
13.4.2.1 call . . . 64
13.4.2.2 createDeferred . . . 64
13.4.2.3 off . . . 64
13.4.2.4 on . . . 65
13.4.2.5 trigger . . . 65

**14 Class Documentation** **67**

14.1 Coherent.UI.Binding.BoundObject Struct Reference . . . 67
14.1.1 Detailed Description . . . 67
14.1.2 Member Function Documentation . . . 67
14.1.2.1 BindMethods . . . 67
14.2 Coherent.UI.BrowserView Class Reference . . . 67
14.2.1 Detailed Description . . . 68
14.2.2 Member Function Documentation . . . 68
14.2.2.1 GetHistory . . . 68
14.2.2.2 GetViewType . . . 68
14.2.2.3 GoBack . . . 68
14.2.2.4 GoForward . . . 68

14.3 Coherent.UI.BrowserViewListener Class Reference . . . 68
14.3.1 Detailed Description . . . 71
14.3.2 Member Function Documentation . . . 71
14.3.2.1 CreateSurface . . . 71
14.3.2.2 DestroySurface . . . 72
14.3.2.3 OnAudioStreamClose . . . 73
14.3.2.4 OnAudioStreamCreated . . . 73
14.3.2.5 OnAudioStreamPause . . . 73
14.3.2.6 OnAudioStreamPlay . . . 73
14.3.2.7 OnBindingsReleased . . . 73
14.3.2.8 OnCallback . . . 74
14.3.2.9 OnCanCreateChildWindow . . . 74
14.3.2.10 OnCaretRectChanged . . . 74
14.3.2.11 OnCertificateError . . . 74
14.3.2.12 OnCursorChanged . . . 75
14.3.2.13 OnDraw . . . 75
14.3.2.14 OnError . . . 75
14.3.2.15 OnFailLoad . . . 75
14.3.2.16 OnFileSelectRequest . . . 76
14.3.2.17 OnFinishLoad . . . 76
14.3.2.18 OnGetAuthCredentials . . . 76
14.3.2.19 OnIMEShouldCancelComposition . . . 76
14.3.2.20 OnJavaScriptMessage . . . 76
14.3.2.21 OnNavigateTo . . . 77
14.3.2.22 OnReadyForBindings . . . 77
14.3.2.23 OnRequestMediaStream . . . 77
14.3.2.24 OnScriptMessage . . . 77
14.3.2.25 OnStartLoading . . . 78
14.3.2.26 OnStopLoading . . . 78
14.3.2.27 OnTextInputTypeChanged . . . 78
14.3.2.28 OnURLRequest . . . 78
14.3.2.29 OnViewCreated . . . 78
14.3.2.30 Release . . . 78
14.3.3 Event Documentation . . . 78
14.3.3.1 AudioStreamClose . . . 78
14.3.3.2 AudioStreamCreated . . . 79
14.3.3.3 AudioStreamPause . . . 79
14.3.3.4 AudioStreamPlay . . . 79
14.3.3.5 BindingsReleased . . . 79
14.3.3.6 Callback . . . 79

14.3.3.7 CaretRectChanged . . . 79
14.3.3.8 CertificateError . . . 79
14.3.3.9 CursorChanged . . . 79
14.3.3.10 Draw . . . 79
14.3.3.11 Error . . . 79
14.3.3.12 FailLoad . . . 79
14.3.3.13 FileSelectRequest . . . 79
14.3.3.14 FinishLoad . . . 80
14.3.3.15 GetAuthCredentials . . . 80
14.3.3.16 IMEShouldCancelComposition . . . 80
14.3.3.17 JavaScriptMessage . . . 80
14.3.3.18 NavigateTo . . . 80
14.3.3.19 ReadyForBindings . . . 80
14.3.3.20 RequestMediaStream . . . 80
14.3.3.21 ScriptMessage . . . 80
14.3.3.22 StartLoading . . . 80
14.3.3.23 StopLoading . . . 80
14.3.3.24 TextInputTypeChanged . . . 80
14.3.3.25 URLRequest . . . 80
14.3.3.26 ViewCreated . . . 81
14.4 Coherent.UI.BrowserViewListenerBase Class Reference . . . 81
14.4.1 Detailed Description . . . 81
14.5 Coherent.UI.Button Class Reference . . . 81
14.5.1 Detailed Description . . . 81
14.5.2 Member Function Documentation . . . 82
14.5.2.1 Initialize . . . 82
14.5.3 Member Data Documentation . . . 83
14.5.3.1 Click . . . 83
14.5.3.2 Disabled . . . 83
14.5.3.3 Label . . . 83
14.6 Coherent.UI.ButtonHandler Class Reference . . . 83
14.6.1 Detailed Description . . . 83
14.6.2 Member Function Documentation . . . 84
14.6.2.1 Initialize . . . 84
14.6.2.2 Invoke . . . 85
14.6.2.3 Invoker . . . 85
14.6.3 Member Data Documentation . . . 85
14.6.3.1 Handler . . . 85
14.6.3.2 IsMessage . . . 85
14.6.3.3 Method . . . 85

14.6.3.4 Target . . . 85
14.7 Coherent.UI.Binding.CallbackArguments Class Reference . . . 85
14.7.1 Detailed Description . . . 85
14.7.2 Member Data Documentation . . . 85
14.7.2.1 Arguments . . . 85
14.8 Coherent.UI.CertificatePrincipal Class Reference . . . 86
14.8.1 Detailed Description . . . 86
14.8.2 Property Documentation . . . 86
14.8.2.1 CommonName . . . 86
14.8.2.2 CountryName . . . 86
14.8.2.3 DomainComponents . . . 86
14.8.2.4 LocalityName . . . 86
14.8.2.5 OrganizationNames . . . 86
14.8.2.6 OrganizationUnitNames . . . 87
14.8.2.7 StateOrProvinceName . . . 87
14.8.2.8 StreetAddresses . . . 87
14.9 Coherent.UI.ChildViewInfo Class Reference . . . 87
14.9.1 Detailed Description . . . 87
14.9.2 Property Documentation . . . 87
14.9.2.1 Listener . . . 87
14.9.2.2 NewViewInfo . . . 87
14.10 Coherent.UI.Binding.CoherentProperty Class Reference . . . 87
14.10.1 Detailed Description . . . 88
14.10.2 Constructor & Destructor Documentation . . . 88
14.10.2.1 CoherentProperty . . . 88
14.10.2.2 CoherentProperty . . . 88
14.11 Coherent.UI.Binding.CoherentType Class Reference . . . 88
14.11.1 Detailed Description . . . 88
14.12 CoherentUILiveGameView Class Reference . . . 88
14.12.1 Detailed Description . . . 88
14.13 CoherentUISystem Class Reference . . . 88
14.13.1 Detailed Description . . . 89
14.13.2 Member Function Documentation . . . 89
14.13.2.1 IsReady . . . 89
14.13.3 Member Data Documentation . . . 90
14.13.3.1 FileHandlerFactoryFunc . . . 90
14.13.3.2 m_MainCamera . . . 90
14.13.3.3 SystemListenerFactoryFunc . . . 90
14.13.4 Property Documentation . . . 90
14.13.4.1 DiskCacheSize . . . 90

14.13.4.2 HasFocusedView . . . 90
14.13.4.3 IsUpdating . . . 90
14.13.4.4 MemoryCacheSize . . . 91
14.13.4.5 UISystem . . . 91
14.13.4.6 UseURLCache . . . 91
14.13.5 Event Documentation . . . 91
14.13.5.1 OnViewFocused . . . 91
14.14 CoherentUIView Class Reference . . . 91
14.14.1 Detailed Description . . . 93
14.14.2 Member Function Documentation . . . 93
14.14.2.1 DestroyView . . . 93
14.14.2.2 GetCamDimensions . . . 93
14.14.2.3 Reload . . . 93
14.14.2.4 Resize . . . 94
14.14.2.5 SetMousePosition . . . 94
14.14.2.6 ViewDestroyedHandler . . . 94
14.14.3 Property Documentation . . . 94
14.14.3.1 ClickThroughAlphaThreshold . . . 94
14.14.3.2 ClickToFocus . . . 94
14.14.3.3 CorrectGamma . . . 94
14.14.3.4 DrawAfterPostEffects . . . 94
14.14.3.5 EnableBindingAttribute . . . 95
14.14.3.6 EnableIME . . . 95
14.14.3.7 EnableWebGLSupport . . . 95
14.14.3.8 FlipY . . . 95
14.14.3.9 ForceSoftwareRendering . . . 95
14.14.3.10 Height . . . 95
14.14.3.11 IgnoreDisplayDensity . . . 95
14.14.3.12 InitialScript . . . 96
14.14.3.13 InputState . . . 96
14.14.3.14 InterceptAllEvents . . . 96
14.14.3.15 IsIndependentOfZBuffer . . . 96
14.14.3.16 IsSurfaceView . . . 96
14.14.3.17 IsTransparent . . . 96
14.14.3.18 Listener . . . 96
14.14.3.19 OnReadyForBindings . . . 96
14.14.3.20 OnViewCreated . . . 97
14.14.3.21 Page . . . 97
14.14.3.22 ReceivesInput . . . 97
14.14.3.23 ScaleToFit . . . 97

14.14.3.24 ShowJavaScriptDialogs . . . 97
14.14.3.25 SupportClickThrough . . . 97
14.14.3.26 UseCameraDimensions . . . 97
14.14.3.27 View . . . 97
14.14.3.28 Width . . . 98
14.14.3.29 XPos . . . 98
14.14.3.30 YPos . . . 98
14.14.4 Event Documentation . . . 98
14.14.4.1 OnViewDestroyed . . . 98
14.15 Coherent.UI.ContextError Class Reference . . . 98
14.15.1 Detailed Description . . . 98
14.15.2 Property Documentation . . . 98
14.15.2.1 Error . . . 98
14.15.2.2 ErrorCode . . . 98
14.16 Coherent.UI.ContextListener Class Reference . . . 99
14.16.1 Detailed Description . . . 99
14.16.2 Member Function Documentation . . . 99
14.16.2.1 ContextReady . . . 99
14.16.2.2 OnCookiesSaved . . . 99
14.16.2.3 OnDownloadComplete . . . 99
14.16.2.4 OnDownloadFailed . . . 99
14.16.2.5 OnDownloadStarted . . . 100
14.16.2.6 OnError . . . 100
14.16.2.7 OnSubscriptionCheckDone . . . 100
14.17 Coherent.UI.ContextSettings Class Reference . . . 100
14.17.1 Detailed Description . . . 100
14.17.2 Property Documentation . . . 100
14.17.2.1 CachePath . . . 100
14.17.2.2 CookiesResource . . . 101
14.17.2.3 CustomUserAgentString . . . 101
14.17.2.4 HTML5LocalStoragePath . . . 101
14.18 Coherent.UI.ContextSettingsBase Class Reference . . . 101
14.18.1 Detailed Description . . . 102
14.18.2 Property Documentation . . . 102
14.18.2.1 AdditionalHostOptions . . . 102
14.18.2.2 AllowCookies . . . 102
14.18.2.3 DebuggerPort . . . 102
14.18.2.4 DisableWebSecurity . . . 102
14.18.2.5 EnableClientAudioPlayback . . . 102
14.18.2.6 EnableProxy . . . 102

14.18.2.7 ForceDisablePluginFullscreen . . . 102
14.18.2.8 SubscriptionEmail . . . 102
14.18.2.9 SubscriptionKey . . . 102
14.18.2.10 UserAgent . . . 102
14.19 Coherent.UI.Download Class Reference . . . 103
14.19.1 Detailed Description . . . 103
14.19.2 Member Function Documentation . . . 103
14.19.2.1 Cancel . . . 103
14.19.2.2 Destroy . . . 103
14.19.2.3 GetDownloadedBytesSoFar . . . 104
14.19.2.4 GetFilename . . . 104
14.19.2.5 GetId . . . 104
14.19.2.6 GetReferrer . . . 104
14.19.2.7 GetTotalBytes . . . 104
14.19.2.8 GetURL . . . 104
14.19.2.9 HasFailed . . . 105
14.19.2.10 IsCancelled . . . 105
14.19.2.11 IsComplete . . . 105
14.20 Coherent.UI.EventModifiersState Class Reference . . . 105
14.20.1 Detailed Description . . . 105
14.21 Coherent.UI.EventMouseModifiersState Class Reference . . . 105
14.21.1 Detailed Description . . . 105
14.22 Coherent.UI.FactorySettings Class Reference . . . 105
14.22.1 Detailed Description . . . 106
14.22.2 Property Documentation . . . 106
14.22.2.1 AppGroupName . . . 106
14.22.2.2 HostDirectory . . . 106
14.22.2.3 MinidumpFolder . . . 106
14.23 Coherent.UI.FactorySettingsBase Class Reference . . . 106
14.23.1 Detailed Description . . . 107
14.23.2 Property Documentation . . . 107
14.23.2.1 EnableSupportForProprietaryCodecs . . . 107
14.23.2.2 WriteMinidumps . . . 107
14.24 Coherent.UI.FileHandler Class Reference . . . 107
14.24.1 Detailed Description . . . 107
14.24.2 Member Function Documentation . . . 107
14.24.2.1 ReadFile . . . 107
14.24.2.2 WriteFile . . . 108
14.25 Coherent.UI.FileSelectionParams Class Reference . . . 108
14.25.1 Detailed Description . . . 108

14.25.2 Property Documentation . . . 108
14.25.2.1 AcceptedMIMETypes . . . 108
14.25.2.2 AcceptedMIMETypesCount . . . 108
14.25.2.3 DefaultFileName . . . 109
14.25.2.4 DialogTitle . . . 109
14.25.2.5 Mode . . . 109
14.26 Coherent.UI.FileSelectRequest Class Reference . . . 109
14.26.1 Detailed Description . . . 109
14.26.2 Member Function Documentation . . . 109
14.26.2.1 Add . . . 109
14.26.2.2 Finish . . . 109
14.26.2.3 GetFileSelectionParams . . . 110
14.27 Coherent.UI.HTTPHeader Class Reference . . . 110
14.27.1 Detailed Description . . . 111
14.27.2 Member Enumeration Documentation . . . 111
14.27.2.1 FieldType . . . 111
14.27.3 Member Function Documentation . . . 111
14.27.3.1 GetHeaderField . . . 111
14.27.4 Property Documentation . . . 112
14.27.4.1 Content . . . 112
14.27.4.2 Field . . . 112
14.28 Coherent.UI.ILogHandler Class Reference . . . 112
14.28.1 Detailed Description . . . 112
14.28.2 Member Function Documentation . . . 112
14.28.2.1 Assert . . . 112
14.28.2.2 WriteLog . . . 112
14.29 Coherent.UI.ImageData Class Reference . . . 113
14.29.1 Detailed Description . . . 113
14.29.2 Member Function Documentation . . . 113
14.29.2.1 Destroy . . . 113
14.29.2.2 Update . . . 113
14.30 Coherent.UI.Binding.InvalidValueCastException Class Reference . . . 113
14.30.1 Detailed Description . . . 113
14.31 Coherent.UI.KeyEventData Class Reference . . . 114
14.31.1 Detailed Description . . . 114
14.31.2 Property Documentation . . . 114
14.31.2.1 IsAutoRepeat . . . 114
14.31.2.2 IsNumPad . . . 114
14.31.2.3 KeyCode . . . 114
14.31.2.4 Modifiers . . . 114

14.31.2.5 Type . . . 114
14.32 Coherent.UI.MediaStreamDevice Class Reference . . . 115
14.32.1 Detailed Description . . . 115
14.32.2 Property Documentation . . . 115
14.32.2.1 DeviceId . . . 115
14.32.2.2 Name . . . 115
14.32.2.3 Type . . . 115
14.33 Coherent.UI.MediaStreamRequest Class Reference . . . 115
14.33.1 Detailed Description . . . 116
14.33.2 Member Function Documentation . . . 116
14.33.2.1 Respond . . . 116
14.33.3 Property Documentation . . . 116
14.33.3.1 Devices . . . 116
14.34 MobileSurfaceView Class Reference . . . 116
14.34.1 Detailed Description . . . 116
14.34.2 Member Function Documentation . . . 116
14.34.2.1 UpdateView . . . 116
14.35 Coherent.UI.MouseEventData Class Reference . . . 117
14.35.1 Detailed Description . . . 117
14.35.2 Property Documentation . . . 117
14.35.2.1 Button . . . 117
14.35.2.2 Modifiers . . . 117
14.35.2.3 MouseModifiers . . . 117
14.35.2.4 Type . . . 117
14.35.2.5 WheelX . . . 118
14.35.2.6 WheelY . . . 118
14.35.2.7 X . . . 118
14.35.2.8 Y . . . 118
14.36 Coherent.UI.ResourceData Class Reference . . . 118
14.36.1 Detailed Description . . . 118
14.36.2 Member Function Documentation . . . 118
14.36.2.1 GetBuffer . . . 118
14.36.2.2 GetSize . . . 119
14.36.2.3 SignalFailure . . . 119
14.36.2.4 SignalSuccess . . . 119
14.37 Coherent.UI.ResourceResponse Class Reference . . . 119
14.37.1 Detailed Description . . . 119
14.37.2 Member Function Documentation . . . 119
14.37.2.1 GetBuffer . . . 119
14.37.2.2 SetResponseHeader . . . 120

14.37.2.3 SetStatus . . . 120
14.37.2.4 SignalFailure . . . 120
14.37.2.5 SignalSuccess . . . 120
14.38 Coherent.UI.SubscriptionError Class Reference . . . 120
14.38.1 Detailed Description . . . 120
14.38.2 Property Documentation . . . 121
14.38.2.1 Error . . . 121
14.38.2.2 ErrorCode . . . 121
14.39 Coherent.UI.SurfaceResponse Class Reference . . . 121
14.39.1 Detailed Description . . . 121
14.39.2 Member Function Documentation . . . 121
14.39.2.1 Signal . . . 121
14.40 Coherent.UI.TouchEventData Class Reference . . . 121
14.40.1 Detailed Description . . . 122
14.40.2 Property Documentation . . . 122
14.40.2.1 HasRadiusInformation . . . 122
14.40.2.2 Id . . . 122
14.40.2.3 RadiusX . . . 122
14.40.2.4 RadiusY . . . 122
14.40.2.5 Type . . . 122
14.40.2.6 X . . . 122
14.40.2.7 Y . . . 122
14.41 Coherent.UI.Binding.UnsupportedPrimitiveTypeException Class Reference . . . 122
14.41.1 Detailed Description . . . 123
14.42 Coherent.UI.URLRequest Class Reference . . . 123
14.42.1 Detailed Description . . . 123
14.42.2 Member Function Documentation . . . 123
14.42.2.1 AppendExtraHeaderValue . . . 123
14.42.2.2 Deny . . . 124
14.42.2.3 RemoveExtraHeader . . . 124
14.42.2.4 SetExtraHeader . . . 124
14.42.2.5 SetExtraHeaders . . . 124
14.42.2.6 SetMethod . . . 124
14.42.2.7 SetReferrer . . . 124
14.42.2.8 SetURL . . . 124
14.43 Coherent.UI.URLRequestBase Class Reference . . . 125
14.43.1 Detailed Description . . . 125
14.43.2 Member Function Documentation . . . 125
14.43.2.1 GetExtraHeader . . . 125
14.43.2.2 GetExtraHeaderIndex . . . 125

14.43.2.3 GetExtraHeaders . . . 126
14.43.2.4 GetMethod . . . 126
14.43.2.5 GetReferrer . . . 126
14.43.2.6 GetURL . . . 126
14.43.2.7 HasExtraHeader . . . 126
14.44 Coherent.UI.Binding.Value Class Reference . . . 127
14.44.1 Detailed Description . . . 128
14.44.2 Constructor & Destructor Documentation . . . 128
14.44.2.1 Value . . . 128
14.44.2.2 Value . . . 128
14.44.2.3 Value . . . 128
14.44.2.4 Value . . . 128
14.44.2.5 Value . . . 129
14.44.2.6 Value . . . 129
14.44.2.7 Value . . . 129
14.44.2.8 Value . . . 129
14.44.2.9 Value . . . 129
14.44.2.10 Value . . . 129
14.44.2.11 Value . . . 130
14.44.2.12 Value . . . 131
14.44.2.13 Value . . . 131
14.44.3 Member Function Documentation . . . 131
14.44.3.1 Equals . . . 131
14.44.3.2 GetHashCode . . . 131
14.44.3.3 operator bool . . . 131
14.44.3.4 operator byte . . . 131
14.44.3.5 operator char . . . 132
14.44.3.6 operator Dictionary< string, Value > . . . 132
14.44.3.7 operator double . . . 132
14.44.3.8 operator float . . . 132
14.44.3.9 operator int . . . 133
14.44.3.10 operator sbyte . . . 134
14.44.3.11 operator string . . . 134
14.44.3.12 operator uint . . . 134
14.44.3.13 operator Value[] . . . 134
14.44.3.14 operator!= . . . 135
14.44.3.15 operator== . . . 136
14.44.4 Member Data Documentation . . . 136
14.44.4.1 Type . . . 136
14.45 Coherent.UI.Binding.ValueObject Class Reference . . . 136

14.45.1 Detailed Description . . . 136
14.46 Coherent.UI.View Class Reference . . . 136
14.46.1 Detailed Description . . . 139
14.46.2 Member Function Documentation . . . 139
14.46.2.1 AuthCredentialsResponse . . . 139
14.46.2.2 BindCall . . . 140
14.46.2.3 CreateImageData . . . 140
14.46.2.4 Destroy . . . 140
14.46.2.5 DownloadUrl . . . 140
14.46.2.6 DrawSurface . . . 140
14.46.2.7 ExecuteScript . . . 140
14.46.2.8 FetchMouseOnUIQuery . . . 141
14.46.2.9 FetchSurface . . . 141
14.46.2.10 GetAudioData . . . 141
14.46.2.11 GetClickThroughAlphaThreshold . . . 141
14.46.2.12 GetContext . . . 141
14.46.2.13 GetCurentViewPath . . . 142
14.46.2.14 GetHeight . . . 142
14.46.2.15 GetLastFailedPath . . . 142
14.46.2.16 GetLastLoadedPath . . . 142
14.46.2.17 GetLastRequestedPath . . . 142
14.46.2.18 GetTargetFramerate . . . 142
14.46.2.19 GetViewType . . . 142
14.46.2.20 GetWidth . . . 143
14.46.2.21 HasMouseQueryFinished . . . 143
14.46.2.22 IMEActivate . . . 143
14.46.2.23 IMECancelComposition . . . 143
14.46.2.24 IMEConfirmComposition . . . 143
14.46.2.25 IMEIsActive . . . 143
14.46.2.26 IMESetComposition . . . 143
14.46.2.27 InterceptURLRequests . . . 144
14.46.2.28 IsMouseOnView . . . 144
14.46.2.29 IsOnDemand . . . 144
14.46.2.30 IssueMouseOnUIQuery . . . 144
14.46.2.31 IsTransparent . . . 145
14.46.2.32 KeyEvent . . . 145
14.46.2.33 KillFocus . . . 145
14.46.2.34 Load . . . 145
14.46.2.35 MouseActivate . . . 145
14.46.2.36 MouseEvent . . . 145

14.46.2.37 PeekFrameReady . . . 145
14.46.2.38 Redraw . . . 146
14.46.2.39 RegisterForEvent . . . 146
14.46.2.40 Reload . . . 146
14.46.2.41 ReplyToJavaScriptMessage . . . 146
14.46.2.42 RequestFrame . . . 146
14.46.2.43 RequestFrame . . . 146
14.46.2.44 Resize . . . 147
14.46.2.45 SetClickThroughAlphaThreshold . . . 147
14.46.2.46 SetFocus . . . 147
14.46.2.47 SetMasterVolume . . . 147
14.46.2.48 SetScriptError . . . 147
14.46.2.49 SetTargetFramerate . . . 147
14.46.2.50 SetZoomLevel . . . 148
14.46.2.51 SupportsClickThrough . . . 148
14.46.2.52 TouchEvent . . . 148
14.46.2.53 TriggerEvent . . . 148
14.46.2.54 UnbindCall . . . 148
14.46.2.55 UnbindObject . . . 148
14.46.2.56 UnregisterFromEvent . . . 148
14.46.2.57 UsesSharedMemory . . . 149
14.47 Coherent.UI.ViewContext Class Reference . . . 149
14.47.1 Detailed Description . . . 149
14.47.2 Member Function Documentation . . . 150
14.47.2.1 AddCrossOriginWhitelistEntry . . . 150
14.47.2.2 ClearCrossOriginWhitelist . . . 150
14.47.2.3 CreateView . . . 150
14.47.2.4 CreateView . . . 150
14.47.2.5 FetchSurfaces . . . 150
14.47.2.6 RemoveCrossOriginWhitelistEntry . . . 150
14.47.2.7 SaveCookies . . . 151
14.47.2.8 Uninitialize . . . 151
14.47.2.9 Update . . . 151
14.48 Coherent.UI.ViewError Class Reference . . . 151
14.48.1 Detailed Description . . . 151
14.48.2 Property Documentation . . . 151
14.48.2.1 Error . . . 151
14.48.2.2 ErrorCode . . . 151
14.49 Coherent.UI.ViewInfo Class Reference . . . 152
14.49.1 Detailed Description . . . 152

14.49.2 Property Documentation . . . 152
14.49.2.1 ClickThroughAlphaThreshold . . . 152
14.49.2.2 ControlTimeOnDemand . . . 152
14.49.2.3 ForceSoftwareRendering . . . 153
14.49.2.4 Height . . . 153
14.49.2.5 IsOnDemand . . . 153
14.49.2.6 IsTransparent . . . 153
14.49.2.7 SupportClickThrough . . . 153
14.49.2.8 TargetFrameRate . . . 153
14.49.2.9 UsesSharedMemory . . . 153
14.49.2.10 Width . . . 153
14.50 Coherent.UI.ViewListener Class Reference . . . 153
14.50.1 Detailed Description . . . 156
14.50.2 Member Function Documentation . . . 156
14.50.2.1 CreateSurface . . . 156
14.50.2.2 DestroySurface . . . 157
14.50.2.3 OnAudioStreamClose . . . 157
14.50.2.4 OnAudioStreamCreated . . . 157
14.50.2.5 OnAudioStreamPause . . . 157
14.50.2.6 OnAudioStreamPlay . . . 158
14.50.2.7 OnBindingsReleased . . . 158
14.50.2.8 OnCallback . . . 158
14.50.2.9 OnCanCreateChildWindow . . . 158
14.50.2.10 OnCaretRectChanged . . . 159
14.50.2.11 OnCertificateError . . . 160
14.50.2.12 OnCursorChanged . . . 160
14.50.2.13 OnDraw . . . 160
14.50.2.14 OnError . . . 160
14.50.2.15 OnFailLoad . . . 161
14.50.2.16 OnFileSelectRequest . . . 161
14.50.2.17 OnFinishLoad . . . 161
14.50.2.18 OnGetAuthCredentials . . . 161
14.50.2.19 OnIMEShouldCancelComposition . . . 162
14.50.2.20 OnJavaScriptMessage . . . 162
14.50.2.21 OnNavigateTo . . . 162
14.50.2.22 OnReadyForBindings . . . 162
14.50.2.23 OnRequestMediaStream . . . 162
14.50.2.24 OnScriptMessage . . . 163
14.50.2.25 OnStartLoading . . . 163
14.50.2.26 OnStopLoading . . . 163

14.50.2.27 OnTextInputTypeChanged . . . 163
14.50.2.28 OnURLRequest . . . 163
14.50.2.29 OnViewCreated . . . 164
14.50.2.30 Release . . . 165
14.50.3 Event Documentation . . . 165
14.50.3.1 AudioStreamClose . . . 165
14.50.3.2 AudioStreamCreated . . . 165
14.50.3.3 AudioStreamPause . . . 165
14.50.3.4 AudioStreamPlay . . . 165
14.50.3.5 BindingsReleased . . . 165
14.50.3.6 Callback . . . 165
14.50.3.7 CaretRectChanged . . . 165
14.50.3.8 CertificateError . . . 165
14.50.3.9 CursorChanged . . . 165
14.50.3.10 Draw . . . 166
14.50.3.11 Error . . . 166
14.50.3.12 FailLoad . . . 166
14.50.3.13 FileSelectRequest . . . 166
14.50.3.14 FinishLoad . . . 166
14.50.3.15 GetAuthCredentials . . . 166
14.50.3.16 IMEShouldCancelComposition . . . 166
14.50.3.17 JavaScriptMessage . . . 166
14.50.3.18 NavigateTo . . . 166
14.50.3.19 ReadyForBindings . . . 166
14.50.3.20 RequestMediaStream . . . 166
14.50.3.21 ScriptMessage . . . 166
14.50.3.22 StartLoading . . . 167
14.50.3.23 StopLoading . . . 167
14.50.3.24 TextInputTypeChanged . . . 167
14.50.3.25 URLRequest . . . 167
14.50.3.26 ViewCreated . . . 167
14.51 Coherent.UI.ViewListenerBase Class Reference . . . 167
14.51.1 Detailed Description . . . 167
14.51.2 Member Enumeration Documentation . . . 167
14.51.2.1 MessageLevel . . . 167

**Index** **168**

# Chapter 1

# Copyright

Copyright © 2012-2014 Coherent Labs Limited and/or its licensors. All rights reserved in all media.

The coded instructions, statements, computer programs, and/or related material (collectively the "Data") in these files contain confidential and unpublished information proprietary Coherent Labs and/or its licensors, which is protected by United States of America federal copyright law and by international treaties.

This software or source code is supplied under the terms of a license agreement and nondisclosure agreement with Coherent Labs Limited and may not be copied, disclosed, or exploited except in accordance with the terms of that agreement. The Data may not be disclosed or distributed to third parties, in whole or in part, without the prior written consent of Coherent Labs Limited.

COHERENT LABS MAKES NO REPRESENTATION ABOUT THE SUITABILITY OF THIS SOURCE CODE FOR ANY PURPOSE. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER, ITS AFFILIATES, PARENT COMPANIES, LICENSORS, SUPPLIERS, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR PERFORMANCE OF THIS SOFTWARE OR SOURCE CODE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# Chapter 2

# 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.

## 2.1 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.

## 2.2 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 3

# 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.

## 3.1 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.

## 3.2 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 that 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 4

# 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.
**Versions 2.4.4 and later**: CoherentUI 2.4.0.0 and later versions have support for Unity 4.x and 5 versions, but will no longer support Unity3D 3.5.7. In order to support the new Unity 5, the `Coherent UI` package has been reorganized. So if you are upgrading to version 2.4.0.0 or later, it is recommended that you take the following steps before importing the new `Coherent UI` unitypackage in your project:

1. Delete *CoherentUI_Native.dll*, *CoherentUI_Native.pdb*, *libCoherentUI_Native.so* and their meta files from the `Assets/Plugins` folder of your project.
2. Delete *CoherentUINet.dll64*, *CoherentUI64_Native.dll* and their meta files from the `Assets/CoherentUI/Binaries` folder of the project. Now you are ready to import the new CoherentUI package.

## 4.1 Prerequisites

- Unity3D 4 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.

## 4.2 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.

## 4.3 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 CoherentUIViews is enough - they will automatically create an instance of the Coherent UI System for you with reasonable default parameters. Check CoherentUISystem Lifetime for details.

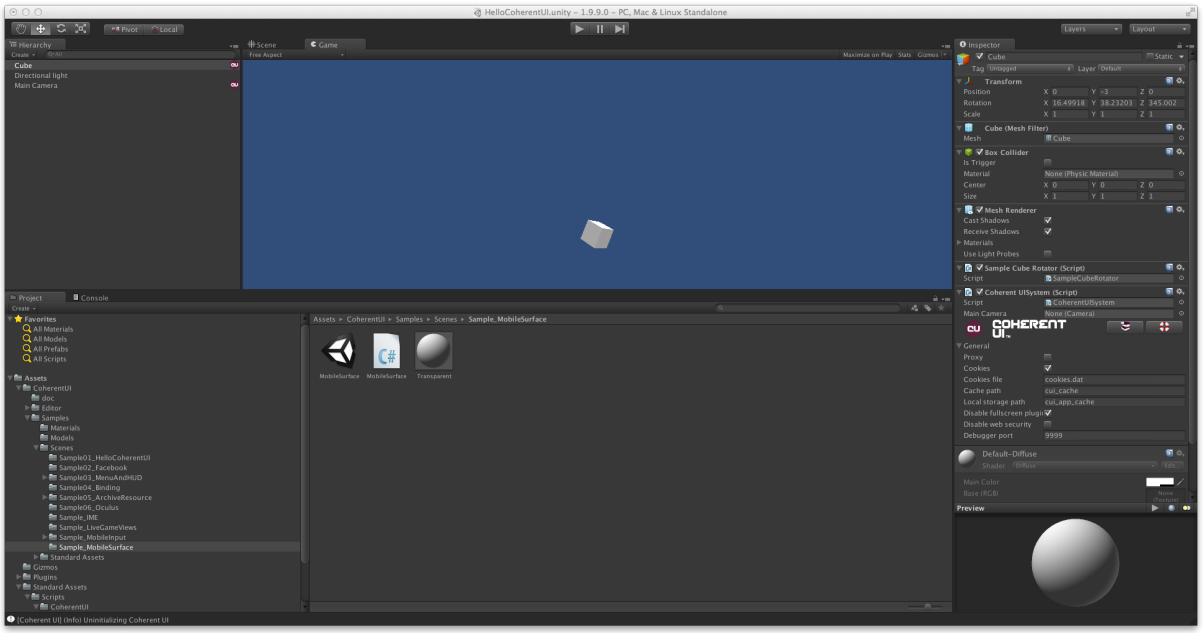


Figure 4.1: CoherentUISystem properties

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.

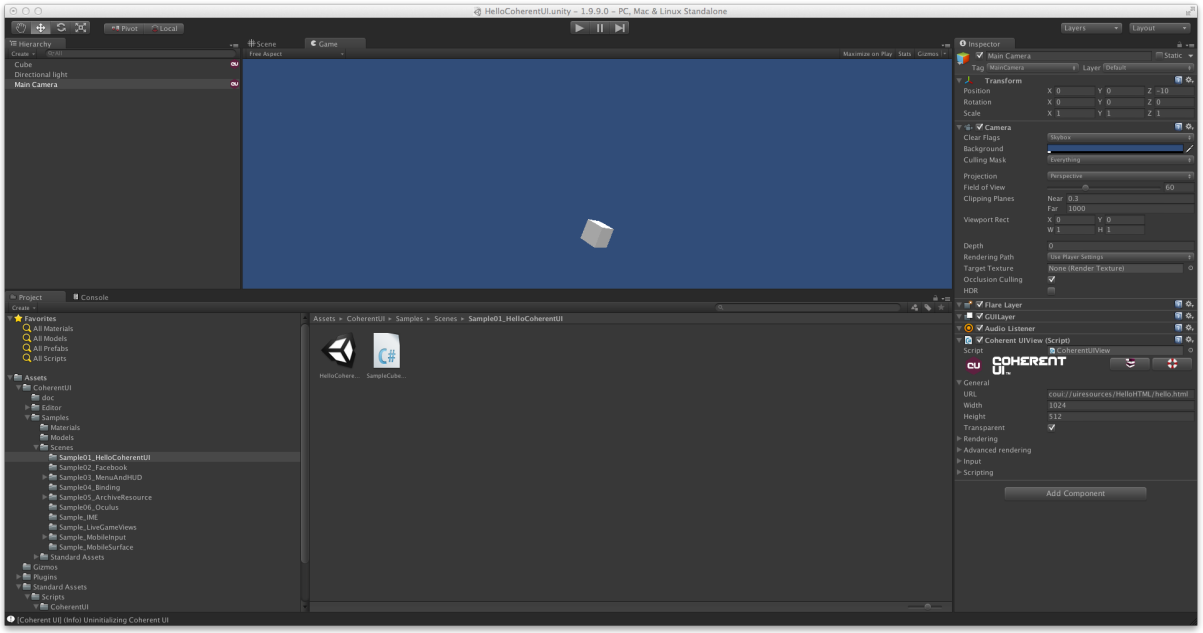


Figure 4.2: CoherentUIView properties

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).

## 4.4 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).

## 4.5 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 5

# 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.

## 5.1 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.

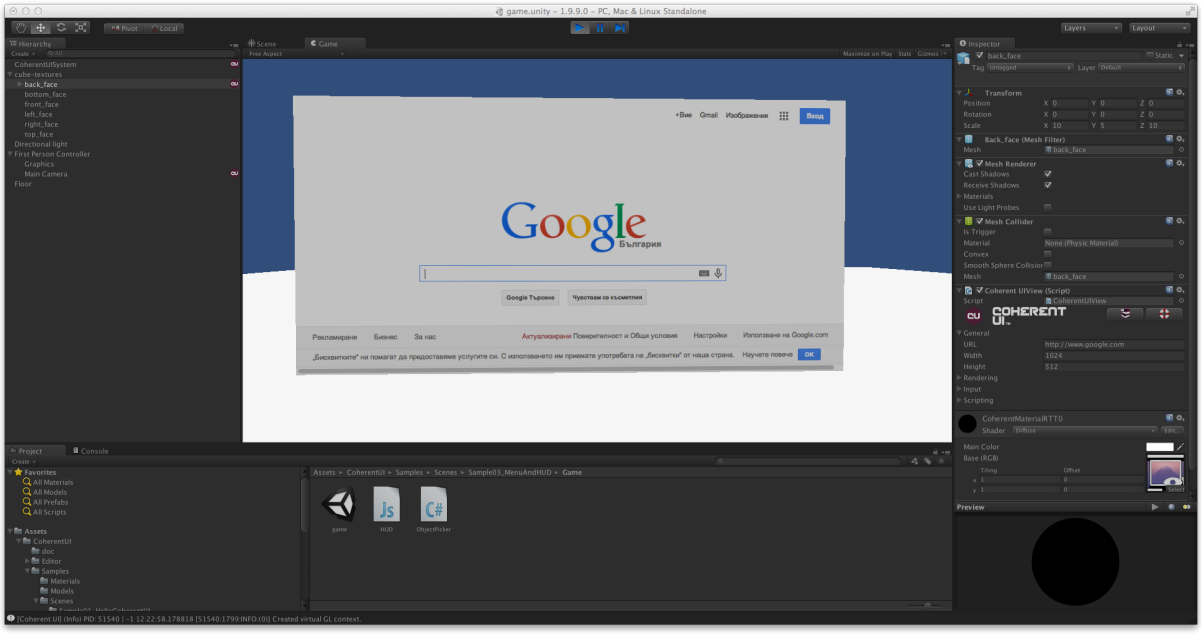



Figure 5.1: CoherentUIView on a texture

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_confirm
- 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.

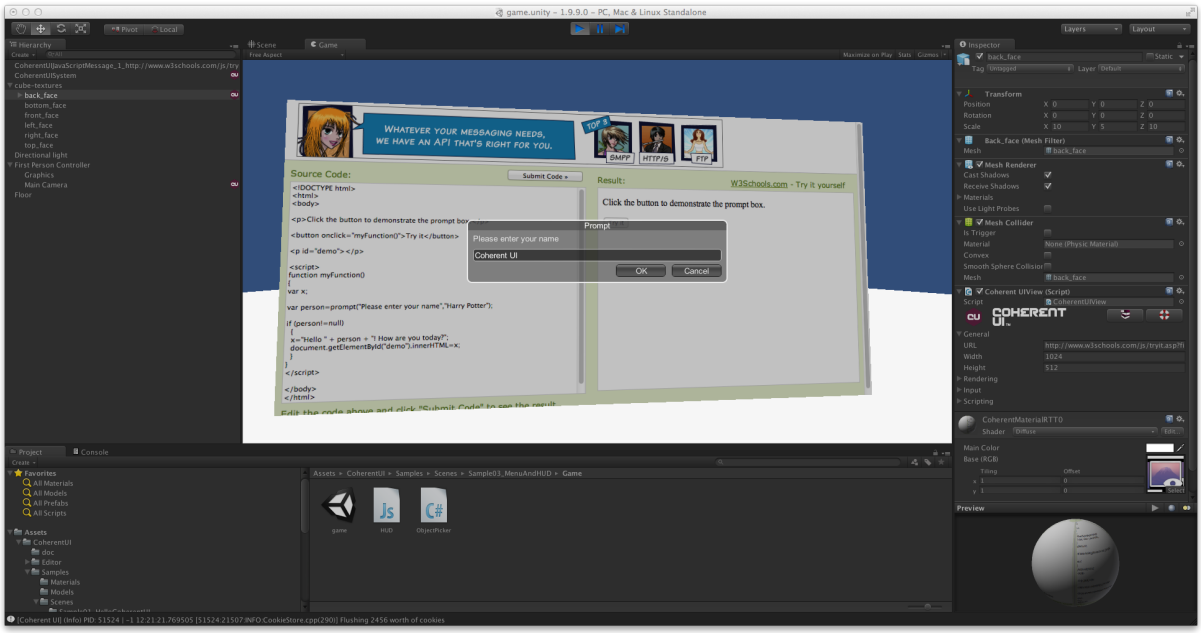



Figure 5.2: CoherentUIView showing a prompt

## 5.2 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.

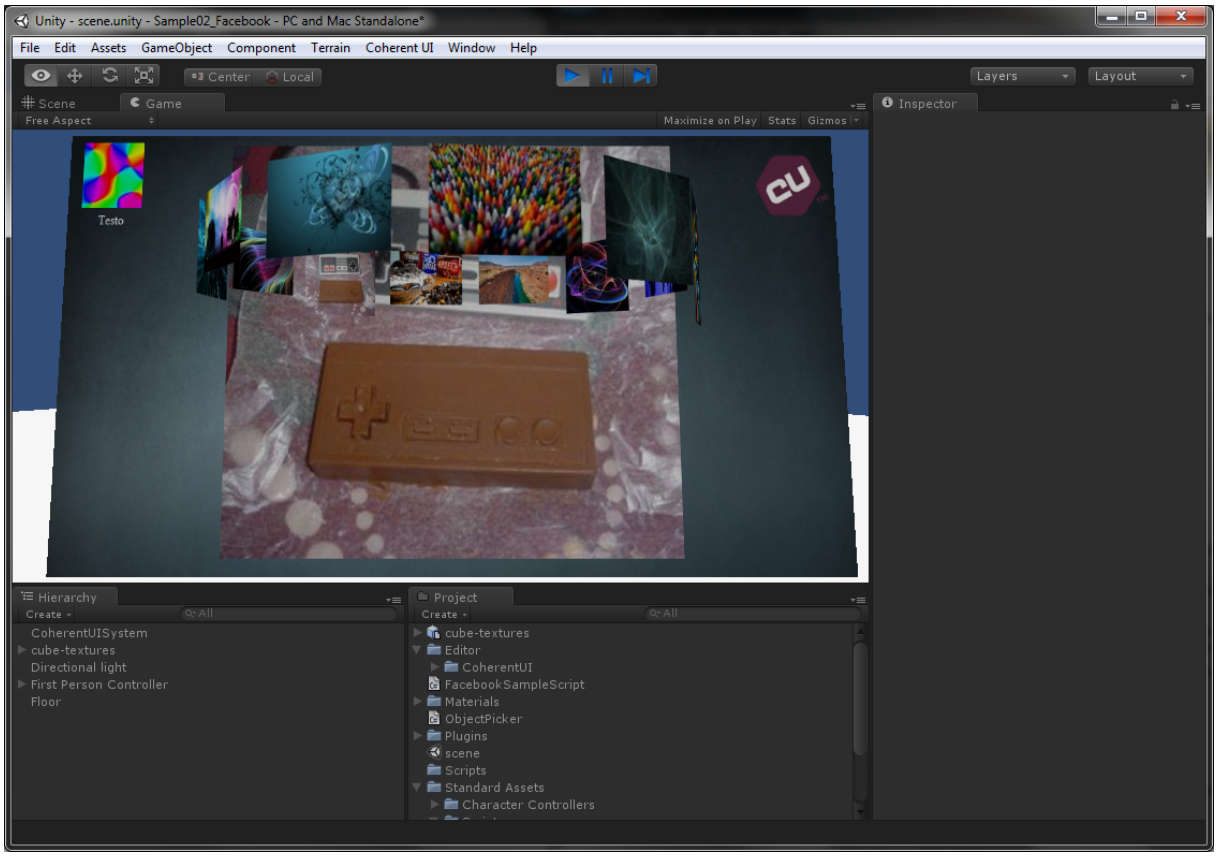



Figure 5.3: CoherentUIView Facebook integration sample

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.

## 5.3 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.

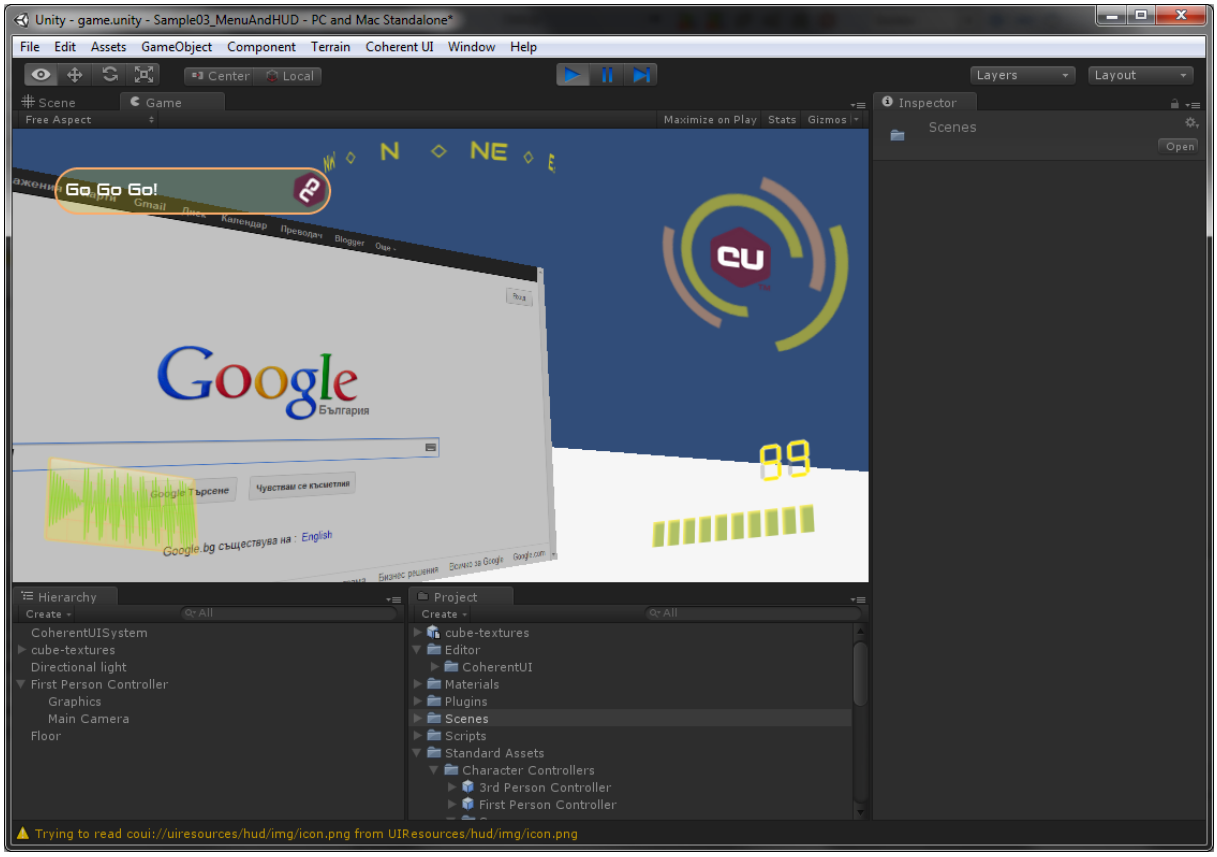



Figure 5.4: CoherentUIView HUD sample in action

## 5.4 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, for UnityScript binding, please check the UnityScript section See the Unity Script Page

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.

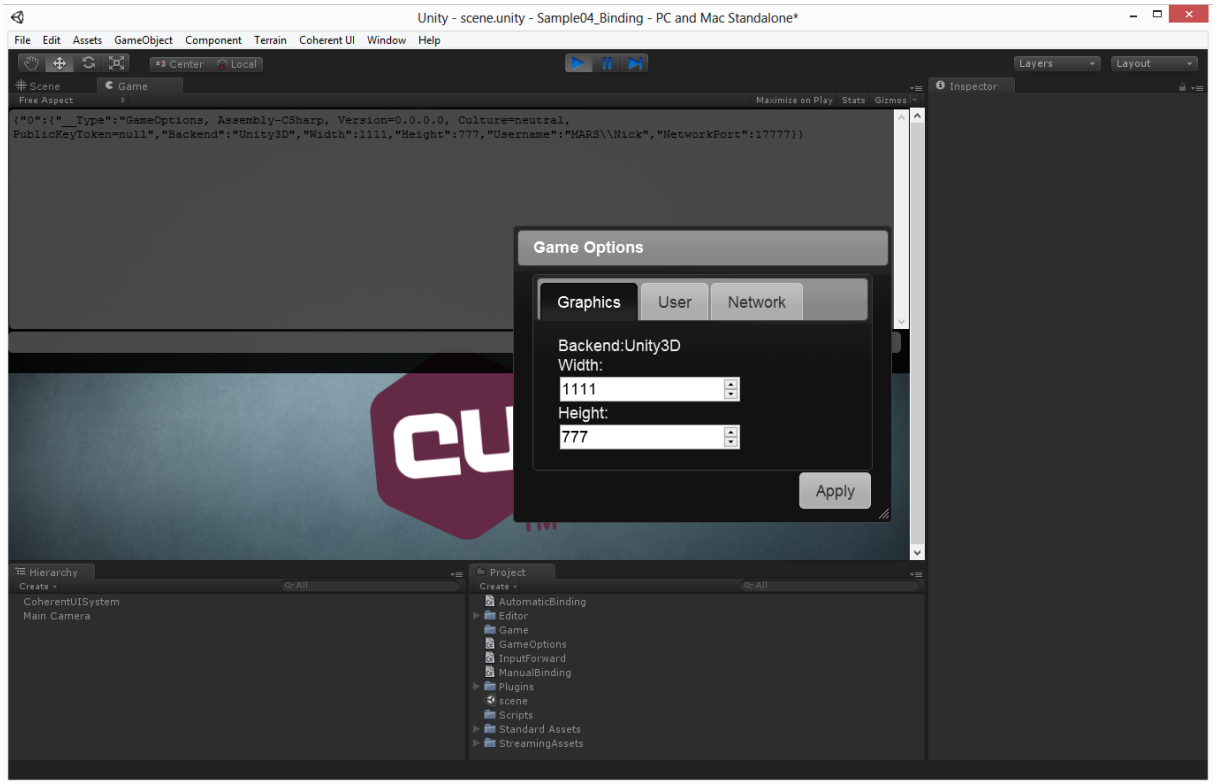



Figure 5.5: Binding sample in action

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.

## 5.5 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.URLRequestBase request,
    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.URLRequestBase request,
    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.

## 5.6 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-plug-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 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.

## 5.7 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

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 OVRCamera-Controller 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 coui://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.

**Note**

If you want to change the draw order of a Coherent UI view, the Left eye camera and a Right eye camera, you can adjust the camera depth property of the camera component for all of the cameras you want to reorder. Cameras with larger depth value will be drawn on top of cameras with lower depth value. When rendering surface views, Coherent UI creates a CoherentRenderingCamera (in the ViewListener.cs script) that has a default depth of 0.

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

## 5.8 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 achive 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*.

## 5.9 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 automatiically 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 int 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.

## 5.10 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 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. This method will help you set the position of the candidate list window.

**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 6

# Programmer's API

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

## 6.1 CoherentUISystem properties

- **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.

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

## 6.2 CoherentUIView properties

General section:

- **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.

Rendering section:

- **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.
- **Texture Filtering** - The filtering mode used for rendering the views. The two available modes are Point Filtering and Linear Filtering. (View Filters - Desktop only )
- **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.

Advanced rendering section:

- **On Demand** - 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.

Note: The synchronization is done in Unity's LateUpdate() method, so updating of objects that have UI elements should be done in regular Update(), or updates of components should be re-arranged (via Script Execution Order Settings) so the update of CoherentUIView is last.

- **Timer override** - The UI uses the in-game timer. (requires On-demand)
- **Max. frame-rate** - Sets the maximum framerate for the View. The view will never exceed this framerate.
- **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.
- **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.

Input section:

- **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 langugages that have more complex input like Chinese.

Scripting section:

- **Pre-load script** - The script will be executed before any other code in the UI View.
- **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]. (Coherent-Method attribute for .NET scripts )
- **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.

## 6.3 CoherentUILiveGameView properties

General section:

- **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 specifing 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 7

# Important points

## 7.1 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 `InterceptAllEvents` 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.

## 7.2 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 walkthrough 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. The 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);
```

## 7.3 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.

## 7.4 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.

## 7.5 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;
```

## 7.6 Customizing initialization of the Coherent UI System

When using only `CoherentUIView` 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.

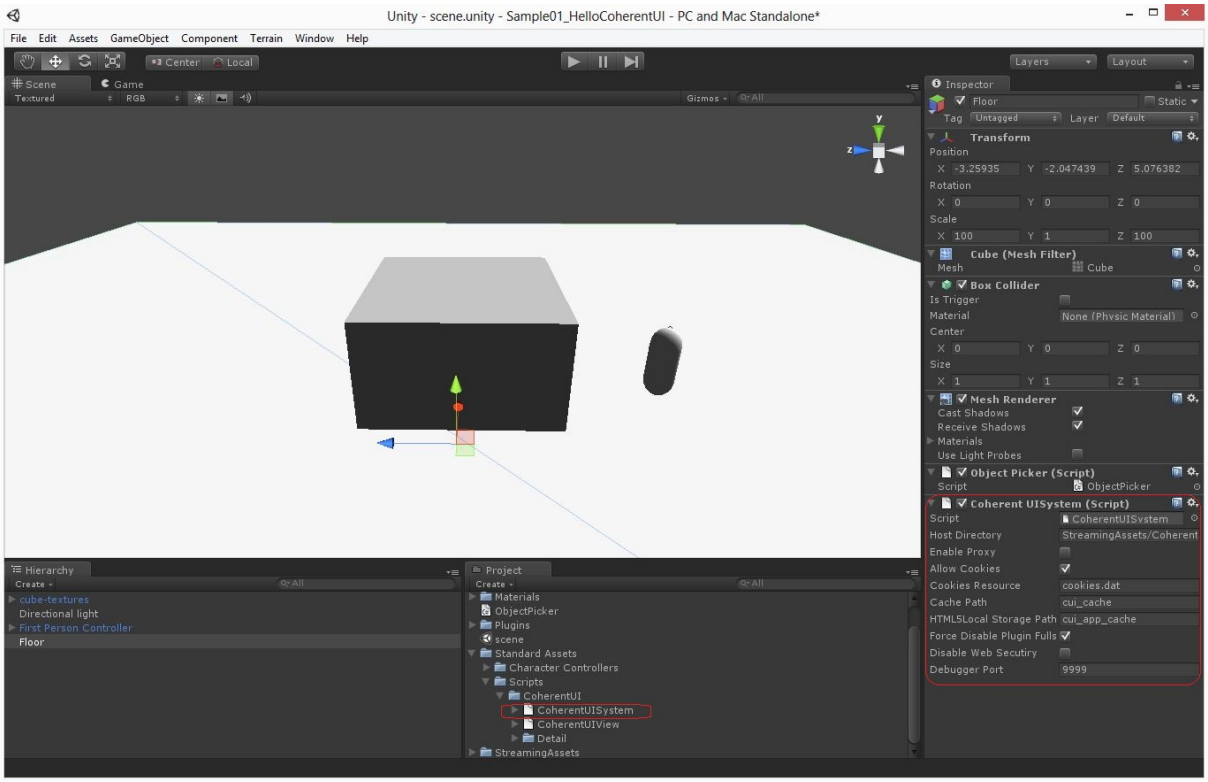



Figure 7.1: Custom CoherentUISystem component

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

## 7.7 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!

## 7.8 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.

## 7.9 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 interactable 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.

## 7.10 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.

## 7.11 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.URLRequestBase request,
            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.

## 7.12 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.

## 7.13 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 `CoherentUIView`s 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:

```
#if UNITY_STANDALONE || UNITY_STANDALONE_WIN || UNITY_STANDALONE_OSX
#define COHERENT_UNITY_STANDALONE
#endif
using UnityEngine;
using System.Collections;

#if COHERENT_UNITY_STANDALONE || UNITY_EDITOR
public class ObjectPicker : MonoBehaviour {

    private Camera m_MainCamera;
    private CoherentUISystem m_UISystem;

    // Use this for initialization
    void Start () {
        m_MainCamera = GameObject.Find("Main Camera").GetComponent<Camera>();
        m_UISystem = Component.FindObjectOfType(typeof(CoherentUISystem)) as
      CoherentUISystem;
    }

    // Update is called once per frame
    void Update () {
        if (m_UISystem.HasFocusedView)
        {
            return;
        }
        // Reset input processing for all views
        foreach (var item in m_UISystem.UIViews) {
            if (!item.ClickToFocus) {
                item.ReceivesInput = false;
            }
        }

        var cameraView = m_MainCamera.gameObject.GetComponent<CoherentUIView>();
        if (cameraView && !cameraView.ClickToFocus)
        {
            var view = cameraView.View;
            if (view != null)
            {
                var factorY = cameraView.Height / m_MainCamera.pixelHeight;

                //Normalize the view coordinates. We need this when we use view dimensions
                //are different than the camera ones
                var normX = Input.mousePosition.x / cameraView.Width;
                var normY = 1 - Input.mousePosition.y / cameraView.Height;

                //After the normalizations, the Normalized Y coordinate will be displaced because Y coords
```

```
                //grow downwards and we have to put it back into the [0-1] range.

                //E.g. if the view height is two times smaller
                //than the camera height, the normalized Y coords will start from -1 and end to +1.
                //If the view's height is 4 times smaller, the normalized coords will start from -3 and end
        to +1

                //The formula puts the Y coord back in [0-1] range.
                normY = ( (normY * factorY) + (1 - factorY) ) / factorY;

                if (normX >= 0 && normX <= 1 && normY >= 0 && normY <= 1)
                {
                    view.IssueMouseOnUIQuery(normX, normY);
                    view.FetchMouseOnUIQuery();
                    if (view.IsMouseOnView())
                    {
                        cameraView.ReceivesInput = true;
                        return;
                    }
                }
            }
        }


        // Activate input processing for the view below the mouse cursor
        RaycastHit hitInfo;
        if (Physics.Raycast(m_MainCamera.ScreenPointToRay(Input.mousePosition), out hitInfo))
        {
            //Debug.Log (hitInfo.collider.name);

            CoherentUIView viewComponent = hitInfo.collider.gameObject.GetComponent(typeof(
      CoherentUIView)) as CoherentUIView;
            if (viewComponent == null)
            {
                viewComponent = hitInfo.collider.gameObject.GetComponentInChildren(typeof(
      CoherentUIView)) as CoherentUIView;
            }

            if (viewComponent != null && !viewComponent.ClickToFocus)
            {
                viewComponent.ReceivesInput = true;
                viewComponent.SetMousePosition(
                    (int)(hitInfo.textureCoord.x * viewComponent.Width),
                    (int)(hitInfo.textureCoord.y * viewComponent.Height));
            }
        }
    }
}
#endif
```

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.

## 7.14 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.

## 7.15 Hit testing - Desktop only

**Note**

For and 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.

## 7.16 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.

## 7.17 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.

## 7.18 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.

## 7.19 Font Rendering

Coherent UI uses `ClearType` on Microsoft Windows. This allows fonts to look the same way as they do in the operating system and the rest of applications.

However, there are fonts that look fuzzy in certain sizes when they are rendered with ClearType. Here is a screenshot of the `Lato` font using ClearType:

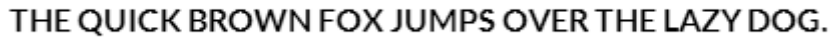



Figure 7.2: Lato font with ClearType

Coherent UI has a CSS property `-coherent-font-cleartype` that gives you control over whether ClearType should be used or standard antialiasing. `-coherent-font-cleartype` has two values:

- `on` - use ClearType if possible. This is the default value
- `off` - do not use ClearType at all

To turn off ClearType for a certain element you can use

```
<p style="-coherent-font-cleartype: off">
    Grumpy wizards make toxic brew for the evil Queen and Jack.
</p>
```

And here is the result with `-coherent-font-cleartype: off`:

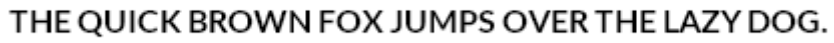



Figure 7.3: Lato font without ClearType

Note that the result of using or turning off ClearType depends a lot on the font itself, the font size and weight.

# Chapter 8

# Unity Script Binding

## 8.1 Sample Unity Script Binding

This is a sample script that binds a function in Unity engine that is to be called from JavaScript. The interface used for the binding is exactly the same as in C#.

The first example shows how a manual binding is done.

```
#pragma strict
import Coherent.UI.Binding;

var m_View : CoherentUIView ;

// Use this for initialization
function Start ()
{
    m_View = GetComponent(typeof(CoherentUIView)) as
      CoherentUIView;
    //Listen for the ReadyForBindings event
    m_View.Listener.ReadyForBindings += HandleReadyForBindings;
}

function HandleReadyForBindings ( frameId:int,
                                  path:String,
                                  isMainFrame : boolean)
{
    if (isMainFrame)
    {
        /* Bind FunctionInUnity so that JavaScript can call it */
        Debug.Log("Ready for bindings..");
        m_View.View.BindCall("CallFunctionInUnity", this.FunctionInUnity);
    }
}

function FunctionInUnity()
{
    Debug.Log("SUCCESS! This function is called from Javascript!");
}
```

Here is another sample - the same code used to the the binding in BindingSample, but in UnityScript and using the automatic binding feature of CoherentUI.

```
#pragma strict
import Coherent.UI.Binding; // to use View.TriggerEvent with extra arguments

public class GameOptions {
    var Backend : String;
    var Width : String;
    var Height : String;
    var NetPort : String;

    function GameOptions(){
        this.Backend = "Unity3D";
        this.Width = "1024";
        this.Height = "768";
        this.NetPort = "17777";
    }
```

```
    function GameOptions(backend : String,
                         width : String,
                         height : String,
                         netPort : String) {
        this.Backend = backend;
        this.Width = width;
        this.Height = height;
        this.NetPort = netPort;
    }
}
private var m_View : CoherentUIView;
private var m_GameOptions;

function Start () {
    m_View = GetComponent(typeof(CoherentUIView)) as CoherentUIView;

    m_GameOptions =  new GameOptions("Unity3D", "1024", "768", "17777");
}

@Coherent.UI.CoherentMethod ("ApplyOptions", false)
function ApplyOptions(options : GameOptions)
{
    m_View.View.TriggerEvent("gameConsole:Trace", options);
}

@Coherent.UI.CoherentMethod("GetLatency")
function GetNetworkLatency()
{
    // not actual latency :)
    return 100;
}

@Coherent.UI.CoherentMethod("GetGameTime")
function GetGameTime()
{
    // not actual time :)
    return 20;
}

@Coherent.UI.CoherentMethod("ViewReady", true)
function ViewReady()
{
    // show the options
    m_View.View.TriggerEvent("OpenOptions", m_GameOptions);
}
```

# Chapter 9

# Coherent UI Menu

Coherent UI Menu Kit allows you to start immediately using Coherent UI for the menus in your game. It also allows to create the menu and implement its logic without leaving Unity3D editor. This menu can be used for a quick prototype, as substitute for the final menus during development and even for the final game after some styling. You can see the menu in action in the *Sample_Menu* in the *CoherentUI/Samples/Scenes* folder.

To create a menu, you have to add the `CoherentUIMenu` component to any game object in the scene.

The properties of `CoherentUIMenu` are:

- `View` - the `CoherentUIView` in which the menu should appear. If it is *None* CoherentUIMenu will look for its view in the same game object.
- `MenuID` - the unique id of this menu in its view. This id is used internally by `CoherentUIMenu`, but can be used from your JavaScript code to manipulate the menu or to listen for its events. See the JavaScript API for more details.
- `Parent` - the id of the element in which to show the menu. You can control the position of the menu by changing the position of the parent element.
- `Visible` - whether the menu is visible by default or not. You can later change the visiblity of the menu using the `Show` and `Hide` methods.
- `Buttons` - the list of buttons in the menu. **Note**: currently the list of the buttons can not be changed after the menu has been created.

Then pick the `CoherentUIView` component in which the menu should be displayed, set its *ID* and parent element ID and add the buttons.

## 9.1 Binding actions to events

To add a button to the menu, simply increase the size of the *Buttons* array. Set its *Label* property to the label you want for the button. You can also mark the button as disabled by checking the *Disabled* property.

**Note**

Currently the enabled/disabled state of the buttons can not be changed after the menu has been created.

To make the button functional, edit the *Click* handler. It has three properties:

- `Target` - the `GameObject` or `MonoBehavior` that is target for the event. It its value is *None*, the GameObject that owns the `CoherentUIMenu` component will be used.
- `Method` - the name of the method or message to be called

- `IsMessage` - if set to `true`, `SendMessage` will be used to execute all methods with name specified by `Method` in `Target`. Otherwise call a single method on the `MonoBehavior` specified by `Target`.

Here is how the final menu looks in the Unity3D Inspector:

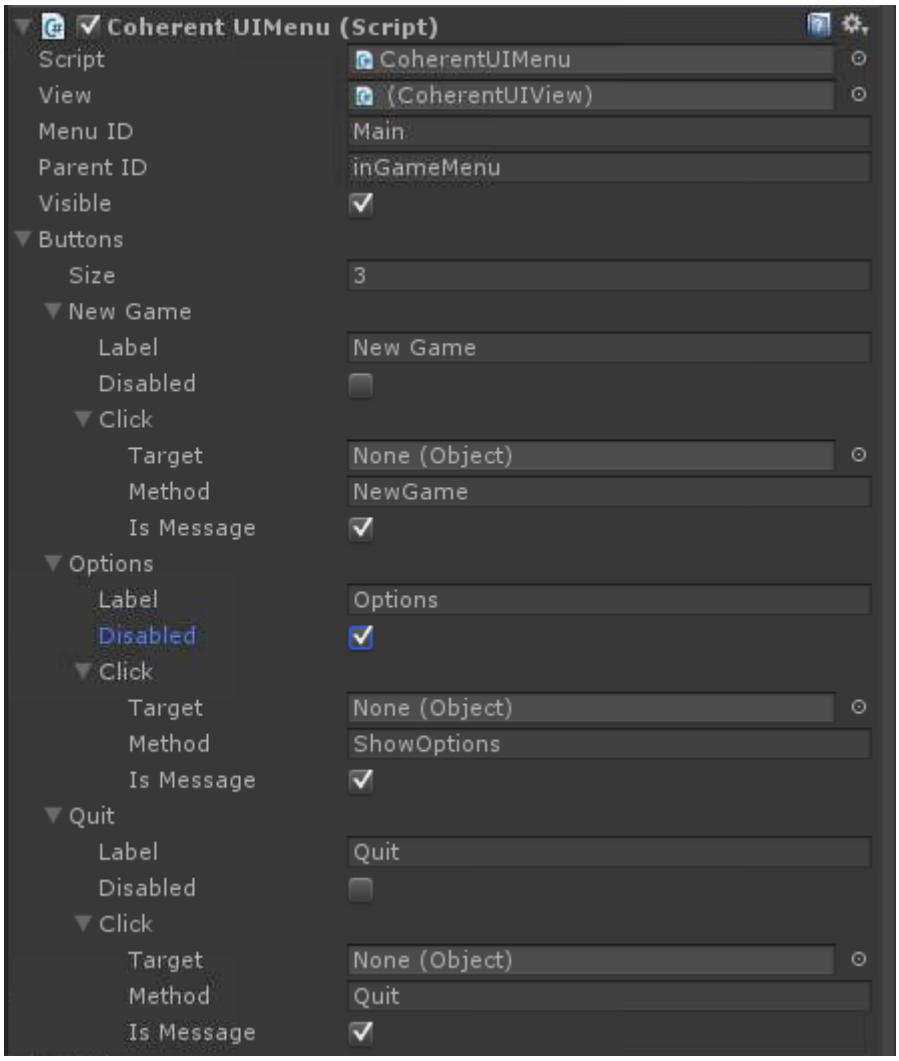



Figure 9.1: Setting up CoherentUIMenu in the Inspector

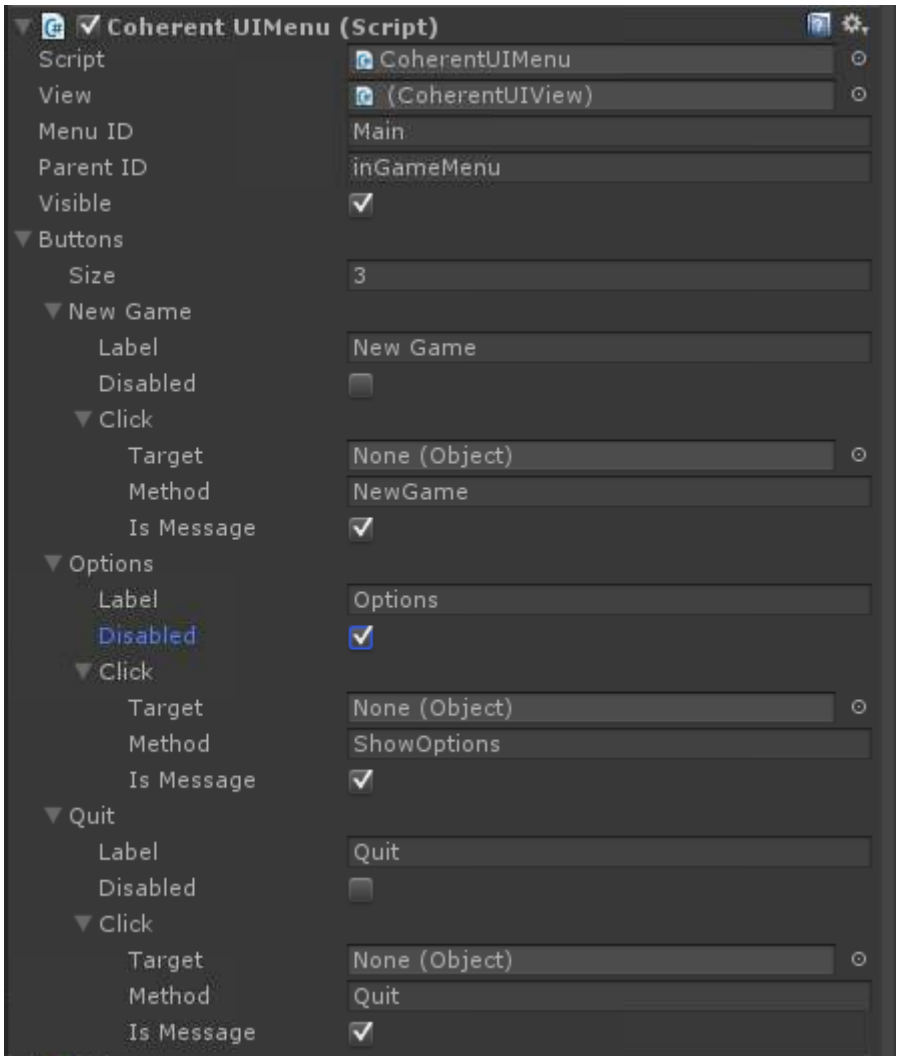



Figure 9.2: Setting up CoherentUIMenu in the Inspector

## 9.2 Setting up the HTML

Using the Coherent UI Menu requires some JavaScript and CSS styles to be present in the view.

The required stylesheets are:

```
<link rel="stylesheet" type="text/css" href="components/flat-ui-official/bootstrap/css/bootstrap.css" />
<link rel="stylesheet" type="text/css" href="components/flat-ui-official/css/flat-ui.css" />
<link rel="stylesheet" type="text/css" href="coherent/css/menu_style.css" />
```

And JavaScript resources:

```
<script type="text/javascript" src="coherent.js"></script>
<script type="text/javascript" src="coherent/js/game_menu.js"></script>
```

**Note**

Please note that *game_menu.js* depends on *coherent.js*, so it is included after *coherent.js*.

## 9.3 JavaScript API

The JavaScript API of the menu kit consist of the following events:

- `cui.MenuButtonClicked` - triggered on button clicked with the *Id* of the menu and the *Label* of the button
- `cui.ShowMenu` - triggered to show the menu with the menu *Id*
- `cui.HideMenu` - triggered to hide the menu with the menu *Id*

You can attach to and trigger these events using the `engine.on` and `engine.trigger` functions.

# Chapter 10

# Namespace Index

## 10.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:


Coherent . . . . . ??
Coherent.UI . . . . . ??
Coherent.UI.Binding . . . . . ??
engine
    Coherent UI JavaScript interface. The `engine` module contains all functions for communication between the UI and the game / application . . . . . ??

# Chapter 11

# Hierarchical Index

## 11.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:


Coherent.UI.Binding.BoundObject . . . ??
Coherent.UI.Button . . . ??
Coherent.UI.ButtonHandler . . . ??
Coherent.UI.Binding.CallbackArguments . . . ??
Coherent.UI.CertificatePrincipal . . . ??
Coherent.UI.ChildViewInfo . . . ??
Coherent.UI.Binding.CoherentProperty . . . ??
Coherent.UI.Binding.CoherentType . . . ??
CoherentUILiveGameView . . . ??
CoherentUISystem . . . ??
CoherentUIView . . . ??
    MobileSurfaceView . . . ??
Coherent.UI.ContextError . . . ??
Coherent.UI.ContextListener . . . ??
Coherent.UI.ContextSettingsBase . . . ??
    Coherent.UI.ContextSettings . . . ??
Coherent.UI.Download . . . ??
Coherent.UI.EventModifiersState . . . ??
Coherent.UI.EventMouseModifiersState . . . ??
Coherent.UI.FactorySettingsBase . . . ??
    Coherent.UI.FactorySettings . . . ??
Coherent.UI.FileHandler . . . ??
Coherent.UI.FileSelectionParams . . . ??
Coherent.UI.FileSelectRequest . . . ??
Coherent.UI.HTTPHeader . . . ??
Coherent.UI.ILogHandler . . . ??
Coherent.UI.ImageData . . . ??
Coherent.UI.Binding.InvalidValueCastException . . . ??
Coherent.UI.KeyEventData . . . ??
Coherent.UI.MediaStreamDevice . . . ??
Coherent.UI.MediaStreamRequest . . . ??
Coherent.UI.MouseEventData . . . ??
Coherent.UI.ResourceData . . . ??
Coherent.UI.ResourceResponse . . . ??
Coherent.UI.SubscriptionError . . . ??
Coherent.UI.SurfaceResponse . . . ??
Coherent.UI.TouchEventData . . . ??

Coherent.UI.Binding.UnsupportedPrimitiveTypeException . . . . . ??
Coherent.UI.URLRequestBase . . . . . ??
Coherent.UI.URLRequest . . . . . ??
Coherent.UI.Binding.Value . . . . . ??
Coherent.UI.Binding.ValueObject . . . . . ??
Coherent.UI.View . . . . . ??
Coherent.UI.BrowserView . . . . . ??
Coherent.UI.ViewContext . . . . . ??
Coherent.UI.ViewError . . . . . ??
Coherent.UI.ViewInfo . . . . . ??
Coherent.UI.ViewListenerBase . . . . . ??
Coherent.UI.BrowserViewListenerBase . . . . . ??
Coherent.UI.BrowserViewListener . . . . . ??
Coherent.UI.ViewListener . . . . . ??

# Chapter 12

# Class Index

## 12.1 Class List

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


Coherent.UI.Binding.BoundObject
: Wrapper class for .Net objects exposed to Coherent UI . . . . . ??

Coherent.UI.BrowserView
: Class that encapsulates a browser Coherent::UI::View . . . . . ??

Coherent.UI.BrowserViewListener
: Interface that allows clients to listen to Coherent::UI::ViewListener . . . . . ??

Coherent.UI.BrowserViewListenerBase
: Interface all browser view listeners inherit. For an easier to use interface inherit instead. Coherent::UI::BrowserViewListener . . . . . ??

Coherent.UI.Button
: Describes a button in a CoherentUIMenu instance . . . . . ??

Coherent.UI.ButtonHandler
: Handler for button related events . . . . . ??

Coherent.UI.Binding.CallbackArguments
: Holds the arguments for the generic callback not handled by a registered delegate . . . . . ??

Coherent.UI.CertificatePrincipal
: Represents certificate principal. . . . . ??

Coherent.UI.ChildViewInfo
: Helper structure for creating child views. . . . . ??

Coherent.UI.Binding.CoherentProperty
: Specify a property / field or method visible to Coherent UI . . . . . ??

Coherent.UI.Binding.CoherentType
: Specify which properties and fields of a type are visible to Coherent UI . . . . . ??

CoherentUILiveGameView
: Component that needs to be attached to a camera and creates a Coherent UI Live Game View ??

CoherentUISystem
: Component controlling the CoherentUI System . . . . . ??

CoherentUIView
: Component containing a Coherent UI view. . . . . ??

Coherent.UI.ContextError
: Encapsulates a context-related error. . . . . ??

Coherent.UI.ContextListener
: Abstract interface to listen to Context-related events. . . . . ??

Coherent.UI.ContextSettings
: Encapsulates the settings of the . . . . . ??

Coherent.UI.ContextSettingsBase
: Encapsulates the settings of the . . . . . ??

Coherent.UI.Download
Encapsulates a download task. . . . ??
Coherent.UI.EventModifiersState
The state of the key modifiers when an event happens. . . . ??
Coherent.UI.EventMouseModifiersState
The state of the mouse modifiers when an event happens. . . . ??
Coherent.UI.FactorySettings
Encapsulates the settings of the UIFactory. . . . ??
Coherent.UI.FactorySettingsBase
Encapsulates the settings of the UIFactory. . . . ??
Coherent.UI.FileHandler
Abstract interface that allows clients to provide their own file-handling functionality. . . . ??
Coherent.UI.FileSelectionParams
Parameters for file selection request. . . . ??
Coherent.UI.FileSelectRequest
File selection request. . . . ??
Coherent.UI.HTTPHeader
Encapsulates a HTTP header field with its content. . . . ??
Coherent.UI.ILogHandler
Interface to allow custom logging. . . . ??
Coherent.UI.ImageData
This class represents a link to a . . . ??
Coherent.UI.Binding.InvalidValueCastException
Thrown when casting a Value to an incompatible type . . . ??
Coherent.UI.KeyEventData
A keyboard event. . . . ??
Coherent.UI.MediaStreamDevice
Represents a media stream device. . . . ??
Coherent.UI.MediaStreamRequest
Represents a request for media stream. . . . ??
MobileSurfaceView
Component containing a Coherent UI view for surfaces on mobile platforms. . . . ??
Coherent.UI.MouseEventData
A mouse event. . . . ??
Coherent.UI.ResourceData
Abstract interface providing data for storing resources. . . . ??
Coherent.UI.ResourceResponse
Abstract interface for responding to read resource requests. . . . ??
Coherent.UI.SubscriptionError
Encapsulates a subscription-check-related error. . . . ??
Coherent.UI.SurfaceResponse
Interface that signals for creation of rendering surfaces. . . . ??
Coherent.UI.TouchEventData
A touch event. . . . ??
Coherent.UI.Binding.UnsupportedPrimitiveTypeException
Thrown when trying to bind a value of unsupported primitive type such as long . . . ??
Coherent.UI.URLRequest
represents a single URL request Allows monitoring and modifications of URL requests . . . ??
Coherent.UI.URLRequestBase
represents a single URL request (read-only) Allows monitoring of URL requests . . . ??
Coherent.UI.Binding.Value
Type for representing generic JavaScript values . . . ??
Coherent.UI.Binding.ValueObject
Class for compound JavaScript objects, behaves like a Dictionary<string, Value> . . . ??
Coherent.UI.View
Class that encapsulates a UI . . . ??
Coherent.UI.ViewContext
Encapsulates basic . . . ??

Coherent.UI.ViewError
Encapsulates a view-related error. . . . . ??
Coherent.UI.ViewInfo
Encapsulates the options of a . . . . . ??
Coherent.UI.ViewListener
Interface that allows clients to listen to . . . . . ??
Coherent.UI.ViewListenerBase
Interface all view listeners inherit. For an easier to use interface inherit instead -. Coherent::UI::-ViewListener . . . . . ??

# Chapter 13

# Namespace Documentation

## 13.1 Package Coherent

### Namespaces

- package UI

## 13.2 Package Coherent.UI

### Namespaces

- package Binding

### Classes

- class HTTPHeader

  *Encapsulates a HTTP header field with its 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::BrowserView-Listener*
- class CertificatePrincipal

  *Represents certificate principal.*
- class ChildViewInfo

  *Helper structure for creating child views.*
- class ContextError

  *Encapsulates a context-related error.*
- class ContextListener

  *Abstract interface to listen to Context-related events.*
- class ContextSettings

  *Encapsulates the settings of the*
- class ContextSettingsBase

  *Encapsulates the settings of the*
- class Download

*Encapsulates a download task.*

- 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 FactorySettings

  *Encapsulates the settings of the UIFactory.*
- class FactorySettingsBase

  *Encapsulates the settings of the UIFactory.*
- class FileHandler

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

  *Parameters for file selection request.*
- class FileSelectRequest

  *File selection request.*
- 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 SubscriptionError

  *Encapsulates a subscription-check-related error.*
- class SurfaceResponse

  *Interface that signals for creation of rendering surfaces.*
- class TouchEventData

  *A touch event.*
- class URLRequest

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

  *represents a single URL request (read-only) Allows monitoring of URL requests*
- class View

  *Class that encapsulates a UI*
- class ViewContext

  *Encapsulates basic*
- 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*
- class ButtonHandler

  *Handler for button related events*
- class Button

  *Describes a button in a CoherentUIMenu instance*

## Enumerations

- enum CertificateStatus

  *Enum containing the possible certificate validation errors.*
- enum ContextErrorType {
  ContextErrorType.CE_Success,
  ContextErrorType.CE_InternalError,
  ContextErrorType.CE_InvalidKey,
  ContextErrorType.CE_MissingComponent,
  ContextErrorType.CE_VersionMismatch,
  ContextErrorType.CE_GPUProcessCrash,
  ContextErrorType.CE_GPUProcessDeviceMismatch }

  *Context-related error-codes.*
- 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_DownloadOperationFailed }
- 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 SubscriptionErrorType {
  SubscriptionErrorType.SET_Success,
  SubscriptionErrorType.SET_SuccessNotify,
  SubscriptionErrorType.SET_OfflineGracePeriod,
  SubscriptionErrorType.SET_Error }
- 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_FailedRenderingCreation,
  ViewErrorType.VE_ClickThroughFailed,
  ViewErrorType.VE_InvalidPath,
  ViewErrorType.VE_ChildCompositionFailed,
  ViewErrorType.VE_InvalidCall,
  ViewErrorType.VE_FrameAlreadyInFlight,
  ViewErrorType.VE_FrameNotReady,
  ViewErrorType.VE_FrameNotRequeted,
  ViewErrorType.VE_AbnormalTermination,
  ViewErrorType.VE_QueryNotReady }

  *View-related error codes.*
- enum ViewType

  *Enum containing the possible view types.*

### 13.2.1 Enumeration Type Documentation

#### 13.2.1.1 enum Coherent.UI.CertificateStatus

Enum containing the possible certificate validation errors.

#### 13.2.1.2 enum Coherent.UI.ContextErrorType

Context-related error-codes.

**Enumerator**

***CE_Success*** Indicates no error.

***CE_InternalError*** Indicates a serious context internal error.

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

***CE_MissingComponent*** Indicates that a component required by Coherent UI is missing.

***CE_VersionMismatch*** Indicates that there is a mismatch between the Coherent UI dynamic library and the Coherent UI Host executable.

***CE_GPUProcessCrash*** Indicates that the GPU process has crashed.

***CE_GPUProcessDeviceMismatch*** Indicates that the GPU process is on diferent GPU device than the current process.

#### 13.2.1.3 enum Coherent.UI.CursorTypes

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

#### 13.2.1.4 enum Coherent.UI.Direct3DInterfaceType

Types of interface used for DirectX.

#### 13.2.1.5 enum Coherent.UI.DownloadErrorType

**Enumerator**

***DET_Success*** Indicates everything went fine.

***DET_DownloadOperationFailed*** Indicates the download operation failed.

#### 13.2.1.6 enum Coherent.UI.FileSelectionMode

Enumeration for file selection modes.

#### 13.2.1.7 enum Coherent.UI.JavaScriptMessageType

Enum containing the possible types of a javascript message.

#### 13.2.1.8 enum Coherent.UI.MediaStreamType

Enumeration of media stream types.

#### 13.2.1.9 enum Coherent.UI.RenderingDeviceType

Types of rendering backend.

#### 13.2.1.10 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.

#### 13.2.1.11 enum Coherent.UI.SubscriptionErrorType

**Enumerator**

***SET_Success*** Indicates everything went fine.

***SET_SuccessNotify*** Indicates everything went fine and the user should be notified.

***SET_OfflineGracePeriod*** Indicates that the subscribed client is offline and is granted a grace period to reactivate.

***SET_Error*** Indicates some type of error. SubscriptionError::Error holds more information.

#### 13.2.1.12 enum Coherent.UI.TextInputControlType

Enum containing the text input types reported for IME.

#### 13.2.1.13 enum Coherent.UI.UserAgentType

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

#### 13.2.1.14 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_FrameNotRequeted*** 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.

#### 13.2.1.15 enum Coherent.UI.ViewType

Enum containing the possible view types.

## 13.3 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,
  ValueType.Char }

  *Specifies the type of Coherent.UI.Binding.Value*

### 13.3.1 Enumeration Type Documentation

#### 13.3.1.1 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

#### 13.3.1.2 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

***Char*** char value

## 13.4 engine Namespace Reference

Coherent UI JavaScript interface. The `engine` module contains all functions for communication between the UI and the game / application.

### Functions

- undefined on (String name, Function callback, Object context)

  *Register handler for and event.*
- undefined off (String name, Function callback, Object context)

  *Remove handler for an event.*
- undefined trigger (String name,...)

  *Trigger an event This function will trigger any C++ handler registered for this event with `Coherent::UI::View::RegisterForEvent`*
- Deferred createDeferred ()

  *Create a new deferred object. Use this to create deferred / promises that can be used together with `engine.call`.*
- Deferred call (String name,...)

  *Call asynchronously a C++ handler and retrieve the result The C++ handler must have been registered with `Coherent::UI::View::BindCall`*

## 13.4.1 Detailed Description

Coherent UI JavaScript interface. The `engine` module contains all functions for communication between the UI and the game / application.

## 13.4.2 Function Documentation

### 13.4.2.1 Deferred engine::call ( String *name*, ... )

Call asynchronously a C++ handler and retrieve the result The C++ handler must have been registered with `Coherent::UI::View::BindCall`

**Parameters**

| | |
|---|---|
| *name* | name of the C++ handler to be called |
| ... | any extra parameters to be passed to the C++ handler |

**Returns**

deferred object whose promise is resolved with the result of the C++ handler

### 13.4.2.2 Deferred engine::createDeferred ( )

Create a new deferred object. Use this to create deferred / promises that can be used together with `engine.call`.

**Returns**

a new deferred object

**See Also**

CustomizingPromises

### 13.4.2.3 undefined engine::off ( String *name*, Function *callback*, Object *context* )

Remove handler for an event.

**Parameters**

| | |
|---|---|
| *name* | name of the event, by default removes all events |
| *callback* | the callback function to be removed, by default removes all callbacks for a given event |
| *context* | *this* context for the function, by default all removes all callbacks, regardless of context |

**Warning**

Removing all handlers for `engine` will remove some *Coherent UI* internal events, breaking some functionality.

#### 13.4.2.4 undefined engine::on ( String *name,* Function *callback,* Object *context* )

Register handler for and event.

**Parameters**

| | |
|---|---|
| *name* | name of the event |
| *callback* | callback function to be executed when the event has been triggered |
| *context* | *this* context for the function, by default the engine object |

#### 13.4.2.5 undefined engine::trigger ( String *name,* ... )

Trigger an event This function will trigger any C++ handler registered for this event with `Coherent::UI::View::RegisterForEvent`

**Parameters**

| | |
|---|---|
| *name* | name of the event |
| ... | any extra arguments to be passed to the event handlers |

# Chapter 14

# Class Documentation

## 14.1 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*

### 14.1.1 Detailed Description

Wrapper class for .Net objects exposed to Coherent UI

### 14.1.2 Member Function Documentation

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

expose object's methods to Coherent UI

**Parameters**

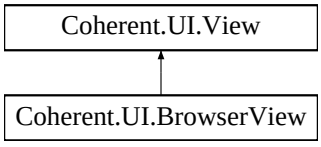
| | |
|---|---|
| *target* | instance |

**Returns**

## 14.2 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*

### 14.2.1 Detailed Description

Class that encapsulates a browser Coherent::UI::View

### 14.2.2 Member Function Documentation

#### 14.2.2.1 virtual void Coherent.UI.BrowserView.GetHistory ( ) [inline], [virtual]

Issues a request for obtaining the navigation history for the view. BrowserViewListener::OnHistoryObtained

#### 14.2.2.2 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.

#### 14.2.2.3 virtual void Coherent.UI.BrowserView.GoBack ( ) [inline], [virtual]

Navigates back to the previous URL, if any.

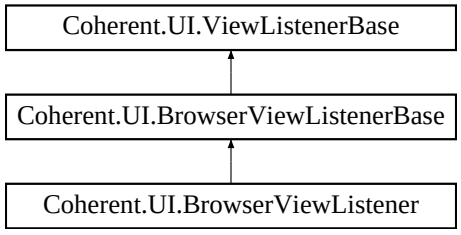
#### 14.2.2.4 virtual void Coherent.UI.BrowserView.GoForward ( ) [inline], [virtual]

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

## 14.3 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 context.*
- 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 (URLRequest request)

  *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 context thread in order to support client applications with multi-threaded rendering.*
- override void OnCertificateError (string url, CertificateStatus status, Certificate certificate, CertificateErrorResponse 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.*
- override bool OnCanCreateChildWindow (string openerUrl, string targetUrl, ChildViewInfo childViewInfo)

  *Called when the view wants to open a new window.*
- override void OnAudioStreamCreated (int streamId, int channels, int bitDepth, int frequency)

  *Called when a new WebAudio stream is created. This is usually when the page is loading (for <audio> HTML tags).*
- override void OnAudioStreamPlay (int streamId)

  *Called when a WebAudio stream is played.*
- override void OnAudioStreamPause (int streamId)

  *Called when a WebAudio stream is stopped.*
- override void OnAudioStreamClose (int streamId)

  *Called when a WebAudio stream is closed. This is usually called when the page is unloading.*

## 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 methof 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.*
- CoherentUI_OnAudioStreamCreated AudioStreamCreated

  *Fired when a new WebAudio stream is created. This is usually when the page is loading (for <audio> HTML tags).*
- CoherentUI_OnAudioStreamPlay AudioStreamPlay

  *Fired when a WebAudio stream is played.*
- CoherentUI_OnAudioStreamPause AudioStreamPause

  *Fired when a WebAudio stream is stopped.*
- CoherentUI_OnAudioStreamClose AudioStreamClose

  *Fired when a WebAudio stream is closed. This is usually called when the page is unloading.*

## Additional Inherited Members

### 14.3.1 Detailed Description

Interface that allows clients to listen to Coherent::UI::ViewListener

### 14.3.2 Member Function Documentation

#### 14.3.2.1 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**

| | |
|---|---|
| *sharedMemory* | true if the surface should be created in shared memory (4 ∗ width ∗ height bytes); false if a shared texture must be created. |
| *width* | the width of the surface in pixels |
| *height* | the height of the surface in pixels |
| *response* | • object to hold the response when the surface is created or fails it's creation - must be signaled |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.2 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 context thread in order to support client applications with multi-threaded rendering.

**Parameters**

| | |
|---|---|
| *surface* | handle to the surface |
| *usesShared-Memory* | determines whether the surface parameter is a handle to shared memory or shared texture |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.3 override void Coherent.UI.BrowserViewListener.OnAudioStreamClose ( int *streamId* ) [inline], [virtual]

Called when a WebAudio stream is closed. This is usually called when the page is unloading.

**Parameters**

| | |
|---|---|
| *streamId* | the ID of the stream that is closed |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.4 override void Coherent.UI.BrowserViewListener.OnAudioStreamCreated ( int *streamId,* int *channels,* int *bitDepth,* int *frequency* ) [inline], [virtual]

Called when a new WebAudio stream is created. This is usually when the page is loading (for <audio> HTML tags).

**Parameters**

| | |
|---|---|
| *streamId* | the generated ID for the new stream |
| *channels* | the number of channels in the stream |
| *bitDepth* | bits per sample in the stream |
| *frequency* | the audio stream frequency, in Hz |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.5 override void Coherent.UI.BrowserViewListener.OnAudioStreamPause ( int *streamId* ) [inline], [virtual]

Called when a WebAudio stream is stopped.

**Parameters**

| | |
|---|---|
| *streamId* | the ID of the stream that is stopped |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.6 override void Coherent.UI.BrowserViewListener.OnAudioStreamPlay ( int *streamId* ) [inline], [virtual]

Called when a WebAudio stream is played.

**Parameters**

| | |
|---|---|
| *streamId* | the ID of the stream that is played |

Reimplemented from Coherent.UI.ViewListenerBase.

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

Called when the bindings for frame are released.

**Parameters**

| | |
|---|---|
| *frameId* | the id of the frame |
| *path* | the path in the frame |
| *isMainFrame* | true if this is the main frame of the view |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.8 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**

| | |
|---|---|
| *eventName* | name of the event |
| *arguments* | arguments of the event invocation |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.9 override bool Coherent.UI.BrowserViewListener.OnCanCreateChildWindow ( string *openerUrl,* string *targetUrl,* ChildViewInfo *childViewInfo* ) [inline],[virtual]

Called when the view wants to open a new window.

**Parameters**

| | |
|---|---|
| *openerUrl* | the URL that wants to open the new window |
| *targetUrl* | the target URL of the new window |
| *childViewInfo* | structure that the user should fill in if she's about to allow new view creation |

**Returns**

true if you allow the creation of a new window, false otherwise (if false, the childViewInfo parameter is ignored)

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.10 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**

| | |
|---|---|
| *the* | x position of the selection caret |
| *the* | y position of the selection caret |
| *the* | width of the selection caret |
| *the* | height of the selection caret |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.11 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**

| | |
|---|---|
| *url* | the url of the request |
| *status* | the error status of the certificate |
| *certificate* | the certificate details. This pointer will be valid only for this call |
| *response* | object to signal whether to continue loading the URL. This pointer will be valid only for this call |

Reimplemented from Coherent.UI.ViewListenerBase.

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

Called when the cursor has changed internally in the

**Parameters**

| | |
|---|---|
| *cursor* | the new cursor |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.13 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**

| | |
|---|---|
| *handle* | 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 View was created. The handle is valid only during this call |
| *usesShared-Memory* | determines whether the handle parameter is a handle to shared memory or shared texture |
| *width* | the width of the surface |
| *height* | the height of the surface |

Reimplemented from Coherent.UI.ViewListenerBase.

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

Called when an error occurs for this specific

**Parameters**

| | |
|---|---|
| *error* | error description |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.15 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**

| | |
|---|---|
| *frameId* | the id of the frame |

| | |
|---|---|
| *validatedPath* | the path in the frame |
| *isMainFrame* | true if this is the main frame of the View |
| *error* | error message for the failure |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.16 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**

| | |
|---|---|
| *request* | contains the file selection params for the request |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.17 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**

| | |
|---|---|
| *frameId* | the id of the loaded frame |
| *validatedPath* | the path loaded in the frame |
| *isMainFrame* | true if this is the main frame of the View |
| *statusCode* | the status code of the response |
| *headers* | an array of header fields |
| *headersCount* | the count of items in the headers array |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.18 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**

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

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.19 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.

#### 14.3.2.20 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**

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

Reimplemented from Coherent.UI.ViewListenerBase.

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

Called when the view starts navigation to a new path.

**Parameters**

| | |
|---|---|
| *path* | URL that the view is navigating to |

Reimplemented from Coherent.UI.ViewListenerBase.

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

Called when a frame is ready for bindings.

**Parameters**

| | |
|---|---|
| *frameId* | the id of the frame |
| *path* | the path in the frame |
| *isMainFrame* | true if this is the main frame of the view |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.23 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**

| | |
|---|---|
| *request* | contains the available media streams for the request |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.24 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**

| | |
|---|---|
| *level* | message level |
| *message* | the text of the message |
| *sourceId* | id of the script (usually file name) |
| *line* | the number of the line in which the message was sent |

Reimplemented from Coherent.UI.ViewListenerBase.

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

Called when a new path has started loading.

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.26 override void Coherent.UI.BrowserViewListener.OnStopLoading ( ) [inline], [virtual]

Called when all load operations have completed.

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.27 override void Coherent.UI.BrowserViewListener.OnTextInputTypeChanged ( TextInputControlType *type*, bool *canComposeInline* ) [inline], [virtual]

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

**Parameters**

| | |
|---|---|
| *type* | the type of the currently focused text input control by the user |
| *canCompose-Inline* | if the IME composition could be performed in-line in the control |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.28 override void Coherent.UI.BrowserViewListener.OnURLRequest ( URLRequest *request* ) [inline], [virtual]

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

**Parameters**

| | |
|---|---|
| *request* | the URLRequest object that can be used to modify the request. This pointer will be valid only for this call |

Reimplemented from Coherent.UI.ViewListenerBase.

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

Called when the requested

**Parameters**

| | |
|---|---|
| *view* | the instance of the view containing all manipulation methods |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.3.2.30 override void Coherent.UI.BrowserViewListener.Release ( ) [inline], [virtual]

Called when the listener is no longer needed by the UI context.

Reimplemented from Coherent.UI.ViewListenerBase.

### 14.3.3 Event Documentation

#### 14.3.3.1 CoherentUI_OnAudioStreamClose Coherent.UI.BrowserViewListener.AudioStreamClose

Fired when a WebAudio stream is closed. This is usually called when the page is unloading.

#### 14.3.3.2 CoherentUI_OnAudioStreamCreated Coherent.UI.BrowserViewListener.AudioStreamCreated

Fired when a new WebAudio stream is created. This is usually when the page is loading (for <audio> HTML tags).

#### 14.3.3.3 CoherentUI_OnAudioStreamPause Coherent.UI.BrowserViewListener.AudioStreamPause

Fired when a WebAudio stream is stopped.

#### 14.3.3.4 CoherentUI_OnAudioStreamPlay Coherent.UI.BrowserViewListener.AudioStreamPlay

Fired when a WebAudio stream is played.

#### 14.3.3.5 CoherentUI_OnBindingsReleased Coherent.UI.BrowserViewListener.BindingsReleased

Fired when the bindings for frame are released.

#### 14.3.3.6 CoherentUI_OnCallback Coherent.UI.BrowserViewListener.Callback

Fired by the UI when there is no registered handler for this event.

#### 14.3.3.7 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.

#### 14.3.3.8 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

#### 14.3.3.9 CoherentUI_OnCursorChanged Coherent.UI.BrowserViewListener.CursorChanged

Fired when the cursor has changed internally in the

#### 14.3.3.10 CoherentUI_OnDraw Coherent.UI.BrowserViewListener.Draw

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

#### 14.3.3.11 CoherentUI_OnError Coherent.UI.BrowserViewListener.Error

Fired when an error occurs for this specific

#### 14.3.3.12 CoherentUI_OnFailLoad Coherent.UI.BrowserViewListener.FailLoad

Fired when a frame has been failed loading.

#### 14.3.3.13 CoherentUI_OnFileSelectRequest Coherent.UI.BrowserViewListener.FileSelectRequest

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

#### 14.3.3.14 CoherentUI_OnFinishLoad Coherent.UI.BrowserViewListener.FinishLoad

Fired when a frame has been successfully loaded.

#### 14.3.3.15 CoherentUI_OnGetAuthCredentials Coherent.UI.BrowserViewListener.GetAuthCredentials

Fired when a view requires authentication credentials.

#### 14.3.3.16 CoherentUI_OnIMEShouldCancelComposition Coherent.UI.BrowserViewListener.IMEShouldCancelComposition

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

#### 14.3.3.17 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.

#### 14.3.3.18 CoherentUI_OnNavigateTo Coherent.UI.BrowserViewListener.NavigateTo

Fired when the view starts navigation to a new path.

#### 14.3.3.19 CoherentUI_OnReadyForBindings Coherent.UI.BrowserViewListener.ReadyForBindings

Fired when a frame is ready for bindings.

#### 14.3.3.20 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.

#### 14.3.3.21 CoherentUI_OnScriptMessage Coherent.UI.BrowserViewListener.ScriptMessage

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

#### 14.3.3.22 CoherentUI_OnStartLoading Coherent.UI.BrowserViewListener.StartLoading

Fired when a new path has started loading.

#### 14.3.3.23 CoherentUI_OnStopLoading Coherent.UI.BrowserViewListener.StopLoading

Fired when all load operations have completed.

#### 14.3.3.24 CoherentUI_OnTextInputTypeChanged Coherent.UI.BrowserViewListener.TextInputTypeChanged

Fired 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

#### 14.3.3.25 CoherentUI_OnURLRequest Coherent.UI.BrowserViewListener.URLRequest

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

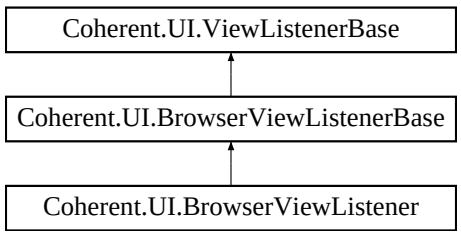
#### 14.3.3.26 CoherentUI_OnViewCreated Coherent.UI.BrowserViewListener.ViewCreated

Fired when the requested

## 14.4 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

### 14.4.1 Detailed Description

Interface all browser view listeners inherit. For an easier to use interface inherit instead. Coherent::UI::BrowserViewListener

## 14.5 Coherent.UI.Button Class Reference

Describes a button in a CoherentUIMenu instance

### Public Member Functions

- void Initialize (CoherentUIMenu owner)

  *Initialize this button*

### Public Attributes

- string Label

  *The label of the button*
- bool Disabled

  *Where the button is enabled or disabled. Currently this property cannot be changed*
- ButtonHandler Click

  *Specifies handler for the click event of the button*

### 14.5.1 Detailed Description

Describes a button in a CoherentUIMenu instance

### 14.5.2 Member Function Documentation

#### 14.5.2.1 void Coherent.UI.Button.Initialize ( CoherentUIMenu *owner* ) [inline]

Initialize this button

**Parameters**

| | |
|---|---|
| *owner* | the menu which contains this button |

### 14.5.3 Member Data Documentation

#### 14.5.3.1 ButtonHandler Coherent.UI.Button.Click

Specifies handler for the click event of the button

#### 14.5.3.2 bool Coherent.UI.Button.Disabled

Where the button is enabled or disabled. Currently this property cannot be changed

#### 14.5.3.3 string Coherent.UI.Button.Label

The label of the button

## 14.6 Coherent.UI.ButtonHandler Class Reference

Handler for button related events

### Public Member Functions

- void Initialize (CoherentUIMenu owner)

  *Initializes the Handler delegate in the Start method. If you want to set a different delegate, you can set it from another component in the Awake method.*
- void Invoke ()

  *Invokes the handler*
- delegate void Invoker ()

  *Delegate signature for the event handler*

### Public Attributes

- Object Target

  *Target GameObject or MonoBehavior for the event. None if using the current GameObject.*
- string Method

  *Method (or message) name to be called for the event*
- bool IsMessage = true

  *Whether Method specifies a method or a message to be called*
- Invoker Handler

  *The delegate that is executed when the event of this handler is triggered*

### 14.6.1 Detailed Description

Handler for button related events

### 14.6.2 Member Function Documentation

#### 14.6.2.1 void Coherent.UI.ButtonHandler.Initialize ( CoherentUIMenu *owner* ) [inline]

Initializes the Handler delegate in the Start method. If you want to set a different delegate, you can set it from another component in the Awake method.

**Parameters**

| | |
|---|---|
| *owner* | the menu that owns our button |

#### 14.6.2.2 void Coherent.UI.ButtonHandler.Invoke ( ) [inline]

Invokes the handler

#### 14.6.2.3 delegate void Coherent.UI.ButtonHandler.Invoker ( )

Delegate signature for the event handler

### 14.6.3 Member Data Documentation

#### 14.6.3.1 Invoker Coherent.UI.ButtonHandler.Handler

The delegate that is executed when the event of this handler is triggered

#### 14.6.3.2 bool Coherent.UI.ButtonHandler.IsMessage = true

Whether Method specifies a method or a message to be called

#### 14.6.3.3 string Coherent.UI.ButtonHandler.Method

Method (or message) name to be called for the event

#### 14.6.3.4 Object Coherent.UI.ButtonHandler.Target

Target GameObject or MonoBehavior for the event. None if using the current GameObject.

## 14.7 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*

### 14.7.1 Detailed Description

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

### 14.7.2 Member Data Documentation

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

Arguments provided by JavaScript

## 14.8 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.*

### 14.8.1 Detailed Description

Represents certificate principal.

### 14.8.2 Property Documentation

#### 14.8.2.1 string Coherent.UI.CertificatePrincipal.CommonName [get]

Null terminated string with the common name of the subject.

#### 14.8.2.2 string Coherent.UI.CertificatePrincipal.CountryName [get]

Null terminated string with the country name of the subject.

#### 14.8.2.3 string [] Coherent.UI.CertificatePrincipal.DomainComponents [get]

The list of domain components storage of the principal

#### 14.8.2.4 string Coherent.UI.CertificatePrincipal.LocalityName [get]

Null terminated string with the locality name of the subject.

#### 14.8.2.5 string [] Coherent.UI.CertificatePrincipal.OrganizationNames [get]

The list of organization names of the principal

#### 14.8.2.6 string [] Coherent.UI.CertificatePrincipal.OrganizationUnitNames [get]

The list of organization unit names of the principal

#### 14.8.2.7 string Coherent.UI.CertificatePrincipal.StateOrProvinceName [get]

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

#### 14.8.2.8 string [] Coherent.UI.CertificatePrincipal.StreetAddresses [get]

The list of street addresses of the principal

## 14.9 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*

### 14.9.1 Detailed Description

Helper structure for creating child views.

### 14.9.2 Property Documentation

#### 14.9.2.1 ViewListenerBase Coherent.UI.ChildViewInfo.Listener [get], [set]

View

#### 14.9.2.2 ViewInfo Coherent.UI.ChildViewInfo.NewViewInfo [get], [set]

An existing

## 14.10 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*

### 14.10.1 Detailed Description

Specify a property / field or method visible to Coherent UI

### 14.10.2 Constructor & Destructor Documentation

#### 14.10.2.1 Coherent.UI.Binding.CoherentProperty.CoherentProperty ( ) [inline]

Make a property / field or method visible to Coherent UI

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

Make a property / field or method visible to Coherent UI

**Parameters**

| | |
|---|---|
| *name* | name of the property / field / method |

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

Specify which properties and fields of a type are visible to Coherent UI

Inherits Attribute.

### 14.11.1 Detailed Description

Specify which properties and fields of a type are visible to Coherent UI

## 14.12 CoherentUILiveGameView Class Reference

Component that needs to be attached to a camera and creates a Coherent UI Live Game View

Inherits MonoBehaviour.

### 14.12.1 Detailed Description

Component that needs to be attached to a camera and creates a Coherent UI Live Game View

## 14.13 CoherentUISystem Class Reference

Component controlling the CoherentUI System

Inherits MonoBehaviour.

### Public Member Functions

- bool IsReady ()

  *Determines whether this instance is ready.*

### Public Attributes

- Camera m_MainCamera = null

  *The main camera. Used for obtaining mouse position over the HUD and raycasting in the world.*

### Static Public Attributes

- static System.Func< FileHandler > FileHandlerFactoryFunc

  *Creates the FileHandler instance for the system. Change to allow usage of custom FileHandler*
- static System.Func
  < System.Action,
  SystemListener > SystemListenerFactoryFunc

  *Creates the SystemListener instance for the system. Change to allow usage of custom EventListener*

### Properties

- bool IsUpdating `[get, set]`

  *Determines whether the Coherent UI System component is currently in its Update() method*
- bool UseURLCache `[get, set]`

  *Sets if the system should use the URL Cache. NOTE: This should almost always be enabled. Disable it if you already set the URL cache yourself for the app*
- int MemoryCacheSize `[get, set]`

  *Sets the in-memory size of the URL cache*
- int DiskCacheSize `[get, set]`

  *Sets the on-disk size of the URL cache*
- bool HasFocusedView `[get]`

  *Determines whether there is an focused Click-to-focus view*
- ViewContext UISystem `[get]`

  *Gets the user interface system.*

### Events

- OnViewFocusedDelegate OnViewFocused

  *Occurs when a Click-to-focus view gains or loses focus*

## 14.13.1 Detailed Description

Component controlling the CoherentUI System

## 14.13.2 Member Function Documentation

#### 14.13.2.1 bool CoherentUISystem.IsReady ( ) `[inline]`

Determines whether this instance is ready.

**Returns**

`true` if this instance is ready; otherwise, `false`.

### 14.13.3 Member Data Documentation

#### 14.13.3.1 System.Func<FileHandler> CoherentUISystem.FileHandlerFactoryFunc [static]

**Initial value:**

```
= () =>
    {


        return new UnityFileHandler();

    }
```

Creates the FileHandler instance for the system. Change to allow usage of custom FileHandler

#### 14.13.3.2 Camera CoherentUISystem.m_MainCamera = null

The main camera. Used for obtaining mouse position over the HUD and raycasting in the world.

#### 14.13.3.3 System.Func<System.Action, SystemListener> CoherentUISystem.SystemListenerFactoryFunc [static]

Creates the SystemListener instance for the system. Change to allow usage of custom EventListener

custom OnSystemReady override must call SystemListener.OnSystemReady

Action to be given to SystemListener constructor

### 14.13.4 Property Documentation

#### 14.13.4.1 int CoherentUISystem.DiskCacheSize [get], [set]

Sets the on-disk size of the URL cache

The maximum size of the on-disk cache

#### 14.13.4.2 bool CoherentUISystem.HasFocusedView [get]

Determines whether there is an focused Click-to-focus view

`true` if there is an focused Click-to-focus view; otherwise, `false`.

#### 14.13.4.3 bool CoherentUISystem.IsUpdating [get], [set]

Determines whether the Coherent UI System component is currently in its Update() method

**Returns**

`true` if this instance is updating; otherwise, `false`.

#### 14.13.4.4 int CoherentUISystem.MemoryCacheSize [get],[set]

Sets the in-memory size of the URL cache

The maximum size of the in-memory cache

#### 14.13.4.5 ViewContext CoherentUISystem.UISystem [get]

Gets the user interface system.

The user interface system.

#### 14.13.4.6 bool CoherentUISystem.UseURLCache [get],[set]

Sets if the system should use the URL Cache. NOTE: This should almost always be enabled. Disable it if you already set the URL cache yourself for the app

If to set the cache

### 14.13.5 Event Documentation

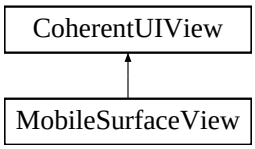
#### 14.13.5.1 OnViewFocusedDelegate CoherentUISystem.OnViewFocused

Occurs when a Click-to-focus view gains or loses focus

## 14.14 CoherentUIView Class Reference

Component containing a Coherent UI view.

Inheritance diagram for CoherentUIView:




### Public Member Functions

- void SetMousePosition (int x, int y)

  *Sets the mouse position. Note Coherent UI (0,0) is the upper left corner of the screen.*
- void DestroyView ()

  *Destroy this view. Destroys the Coherent UI view and removes the CoherentUIView component from its game object. Any usage of the view after this method is undefined behaviour.*
- delegate void ViewDestroyedHandler ()

  *Handler for ViewDestroyed event.*
- void Resize (int width, int height)

  *Resize the view to the specified width and height.*
- void Reload (bool ignoreCache)

  *Request reload of this view.*
- bool GetCamDimensions (out int x, out int y)

  *Returns the camera dimensions of the current view.*

## Properties

- string Page `[get, set]`

  *Gets or sets the URL of the view*
- int Width `[get, set]`

  *Gets or sets the width of the view.*
- int Height `[get, set]`

  *Gets or sets the height of the view.*
- int XPos `[get, set]`

  *Gets or sets the X position of the overlay view.*
- int YPos `[get, set]`

  *Gets or sets the Y position of the overlay view.*
- bool ScaleToFit `[get, set]`

  *Gets or sets if the view will scale it's content to fit it's size.*
- bool EnableWebGLSupport `[get, set]`

  *If enabled, WebGL will be supported in the view.*
- string InitialScript `[get, set]`

  *Gets or sets the initial JavaScript code to be executed when the view JavaScript engine is created.*
- CoherentViewInputState InputState `[get, set]`

  *Gets or sets the inbput state of the overlay view.*
- bool IsTransparent `[get, set]`

  *Gets or sets a value indicating whether the view is transparent.*
- bool SupportClickThrough `[get, set]`

  *Gets or sets a value indicating whether this view CoherentUIView supports click through.*
- float ClickThroughAlphaThreshold `[get, set]`

  *Gets or sets the alpha threshold for click through checks.*
- bool ClickToFocus `[get, set]`

  *When enabled, allows a view to take input focus when clicked with the left mouse button.*
- DrawOrder DrawAfterPostEffects `[get, set]`

  *Gets or sets a value indicating whether this view is drawn after post effects.*
- bool FlipY `[get, set]`

  *Gets or sets a value indicating whether the Y axis of this view should be flipped.*
- virtual bool ReceivesInput `[get, set]`

  *Gets or sets a value indicating whether this view receives input. All automatic processing and reading of this property is done in the `LateUpdate()` / `OnGUI()` callbacks in Unity, letting you do all your logic for View focus in `Update()`.*
- bool InterceptAllEvents `[get, set]`

  *Gets or sets a value indicating whether this view intercepts all events and sends a message for each event.*
- bool EnableBindingAttribute `[get, set]`

  *Gets or sets a value indicating whether this CoherentUIView enables usage of the CoherentMethod attribute in components in the host GameObject. When true, the all components in the host GameObject are inspected for the CoherentMethod attribute (in the Awake() function) and the decorated methods are automatically bound when the ReadyForBindings event is received. When false, the attribute does nothing.*
- bool IsIndependentOfZBuffer `[get, set]`

  *Gets or sets a value indicating whether this view is z-buffer independent. If it is set to true, the view is rendered on top of everything.*
- bool ShowJavaScriptDialogs `[get, set]`

  *Gets or sets a value indicating whether dialogs will be drawn when a JavaScript message is received (e.g. alert, prompt, auth credentials).*
- bool ForceSoftwareRendering `[get, set]`

  *Forces the view to use software rendering. For GPU-bound applications software views might be a good choice. CSS 3D transforms, WebGL and accelerated Canvas don't work with software views. This option doesn't work with OnDemand.*

- bool IgnoreDisplayDensity [get, set]

  *Defines whether the Android device's display density will affect the scale of the View. When set to true, the displayed content in the view will match its pixel size. For example, a device with 1.5 scale factor will display a div of 100px in a 200px View on about 75% of surface if IgnoreDisplayDensity is set to false. If IgnoreDisplayDensity is set to true, the same div of 100px displayed in a 200px View will occupy exactly 50% of the View.*
- bool UseCameraDimensions [get, set]

  *If checked, the view will use the camera's width and height*
- bool CorrectGamma [get, set]

  *Gets or sets a value indicating whether this view should have gamma corrected.*
- bool EnableIME [get, set]

  *Gets or sets a value indicating whether the view should have IME enabled*
- View View [get]

  *Gets the underlying Coherent.UI.View instance.*
- UnityViewListener.CoherentUI_OnViewCreated OnViewCreated

  *Occurs when the underlying Coherent.UI.View is created*
- UnityViewListener.CoherentUI_OnReadyForBindings OnReadyForBindings

  *Occurs when the view bindings should be registered.*
- UnityViewListener Listener [get]

  *Gets the underlying UnityViewListener for this view.*
- virtual bool IsSurfaceView [get]

  *whether this is a mobile surface view*

### Events

- ViewDestroyedHandler OnViewDestroyed

  *Occurs when the view has been destroyed and the CoherentUIView component is going to be removed from the game object.*

### 14.14.1 Detailed Description

Component containing a Coherent UI view.

### 14.14.2 Member Function Documentation

#### 14.14.2.1 void CoherentUIView.DestroyView ( ) [inline]

Destroy this view. Destroys the Coherent UI view and removes the CoherentUIView component from its game object. Any usage of the view after this method is undefined behaviour.

#### 14.14.2.2 bool CoherentUIView.GetCamDimensions ( out int *x*, out int *y* ) [inline]

Returns the camera dimensions of the current view.

#### 14.14.2.3 void CoherentUIView.Reload ( bool *ignoreCache* ) [inline]

Request reload of this view.

**Parameters**

| | |
|---|---|
| *ignoreCache* | Ignore cache for the reload. |

#### 14.14.2.4 void CoherentUIView.Resize ( int *width*, int *height* ) [inline]

Resize the view to the specified width and height.

**Parameters**

| | |
|---|---|
| *width* | New width for the view. |
| *height* | New height for the view. |

#### 14.14.2.5 void CoherentUIView.SetMousePosition ( int *x*, int *y* ) [inline]

Sets the mouse position. Note Coherent UI (0,0) is the upper left corner of the screen.

**Parameters**

| | |
|---|---|
| *x* | X coordinate of the mouse. |
| *y* | Y coordinate of the mouse. |

#### 14.14.2.6 delegate void CoherentUIView.ViewDestroyedHandler ( )

Handler for ViewDestroyed event.

### 14.14.3 Property Documentation

#### 14.14.3.1 float CoherentUIView.ClickThroughAlphaThreshold [get], [set]

Gets or sets the alpha threshold for click through checks.

The alpha threshold for click through checks.

#### 14.14.3.2 bool CoherentUIView.ClickToFocus [get], [set]

When enabled, allows a view to take input focus when clicked with the left mouse button.

`true` if this view takes input focus when clicked; otherwise, `false`.

#### 14.14.3.3 bool CoherentUIView.CorrectGamma [get], [set]

Gets or sets a value indicating whether this view should have gamma corrected.

`true` if gamma is corrected; otherwise, `false`.

#### 14.14.3.4 DrawOrder CoherentUIView.DrawAfterPostEffects [get], [set]

Gets or sets a value indicating whether this view is drawn after post effects.

`AfterPostEffects` if the view is drawn after post effects; otherwise, `false`.

**Exceptions**

| | |
|---|---|
| *System.Application-Exception* | Is thrown when the application exception. |

#### 14.14.3.5 bool CoherentUIView.EnableBindingAttribute [get], [set]

Gets or sets a value indicating whether this CoherentUIView enables usage of the CoherentMethod attribute in components in the host GameObject. When true, the all components in the host GameObject are inspected for the CoherentMethod attribute (in the Awake() function) and the decorated methods are automatically bound when the ReadyForBindings event is received. When false, the attribute does nothing.

`true` if usage of the CoherentMethod is enabled; otherwise, `false`.

#### 14.14.3.6 bool CoherentUIView.EnableIME [get], [set]

Gets or sets a value indicating whether the view should have IME enabled

`true` if IME is enabled; otherwise, `false`.

#### 14.14.3.7 bool CoherentUIView.EnableWebGLSupport [get], [set]

If enabled, WebGL will be supported in the view.

The EnableWebGLSupport property.

#### 14.14.3.8 bool CoherentUIView.FlipY [get], [set]

Gets or sets a value indicating whether the Y axis of this view should be flipped.

`true` if the Y axis is flipped; otherwise, `false`.

#### 14.14.3.9 bool CoherentUIView.ForceSoftwareRendering [get], [set]

Forces the view to use software rendering. For GPU-bound applications software views might be a good choice. CSS 3D transforms, WebGL and accelerated Canvas don't work with software views. This option doesn't work with OnDemand.

Not available for iOS.

`true` if the view is to use software rendering; otherwise `false`.

#### 14.14.3.10 int CoherentUIView.Height [get], [set]

Gets or sets the height of the view.

The height.

#### 14.14.3.11 bool CoherentUIView.IgnoreDisplayDensity [get], [set]

Defines whether the Android device's display density will affect the scale of the View. When set to *true*, the displayed content in the view will match its pixel size. For example, a device with 1.5 scale factor will display a div of 100px in a 200px View on about 75% of surface if IgnoreDisplayDensity is set to false. If IgnoreDisplayDensity is set to true, the same div of 100px displayed in a 200px View will occupy exactly 50% of the View.

`true` if the view ignores the device's display density; otherwise `false`.

#### 14.14.3.12 string CoherentUIView.InitialScript [get], [set]

Gets or sets the initial JavaScript code to be executed when the view JavaScript engine is created.

The script.

#### 14.14.3.13 CoherentViewInputState CoherentUIView.InputState [get], [set]

Gets or sets the inbput state of the overlay view.

The new input state.

#### 14.14.3.14 bool CoherentUIView.InterceptAllEvents [get], [set]

Gets or sets a value indicating whether this view intercepts all events and sends a message for each event.

`true` if view intercepts all events; otherwise, `false`.

**Exceptions**

| | |
|---|---|
| *System.Application-Exception* | Is thrown when the property is modified and the view has already been created |

#### 14.14.3.15 bool CoherentUIView.IsIndependentOfZBuffer [get], [set]

Gets or sets a value indicating whether this view is z-buffer independent. If it is set to true, the view is rendered on top of everything.

`true` if it is independent; otherwise `false`.

#### 14.14.3.16 virtual bool CoherentUIView.IsSurfaceView [get]

whether this is a mobile surface view

#### 14.14.3.17 bool CoherentUIView.IsTransparent [get], [set]

Gets or sets a value indicating whether the view is transparent.

`true` if this instance is transparent; otherwise, `false`.

**Exceptions**

| | |
|---|---|
| *System.Application-Exception* | Is thrown when the property is modified and the view has already been created |

#### 14.14.3.18 UnityViewListener CoherentUIView.Listener [get]

Gets the underlying UnityViewListener for this view.

The listener.

#### 14.14.3.19 UnityViewListener.CoherentUI_OnReadyForBindings CoherentUIView.OnReadyForBindings [add], [remove]

Occurs when the view bindings should be registered.

#### 14.14.3.20 UnityViewListener.CoherentUI_OnViewCreated CoherentUIView.OnViewCreated [add], [remove]

Occurs when the underlying Coherent.UI.View is created

#### 14.14.3.21 string CoherentUIView.Page [get], [set]

Gets or sets the URL of the view

The loaded URL of view

#### 14.14.3.22 virtual bool CoherentUIView.ReceivesInput [get], [set]

Gets or sets a value indicating whether this view receives input. All automatic processing and reading of this property is done in the `LateUpdate()` / `OnGUI()` callbacks in Unity, letting you do all your logic for View focus in `Update()`.

`true` if this view receives input; otherwise, `false`.

#### 14.14.3.23 bool CoherentUIView.ScaleToFit [get], [set]

Gets or sets if the view will scale it's content to fit it's size.

The scale-to-fit property.

#### 14.14.3.24 bool CoherentUIView.ShowJavaScriptDialogs [get], [set]

Gets or sets a value indicating whether dialogs will be drawn when a JavaScript message is received (e.g. alert, prompt, auth credentials).

`true` if a dialog is to be shown automatically; otherwise `false`.

#### 14.14.3.25 bool CoherentUIView.SupportClickThrough [get], [set]

Gets or sets a value indicating whether this view CoherentUIView supports click through.

`true` if supports click through; otherwise, `false`.

**Exceptions**

| | |
|---|---|
| *System.Application-Exception* | Is thrown when the property is modified and the view has already been created |

#### 14.14.3.26 bool CoherentUIView.UseCameraDimensions [get], [set]

If checked, the view will use the camera's width and height

`true` if we want to use camera's width and height; otherwise `false`.

#### 14.14.3.27 View CoherentUIView.View [get]

Gets the underlying Coherent.UI.View instance.

The underlying Coherent.UI.View instance.

#### 14.14.3.28 int CoherentUIView.Width [get], [set]

Gets or sets the width of the view.

The width.

#### 14.14.3.29 int CoherentUIView.XPos [get], [set]

Gets or sets the X position of the overlay view.

The X position.

#### 14.14.3.30 int CoherentUIView.YPos [get], [set]

Gets or sets the Y position of the overlay view.

The Y position.

### 14.14.4 Event Documentation

#### 14.14.4.1 ViewDestroyedHandler CoherentUIView.OnViewDestroyed

Occurs when the view has been destroyed and the CoherentUIView component is going to be removed from the game object.

## 14.15 Coherent.UI.ContextError Class Reference

Encapsulates a context-related error.

Inherits IDisposable.

### Properties

- ContextErrorType ErrorCode [get, set]

  *Indicates an error code.*
- string Error [get, set]

  *Error description.*

### 14.15.1 Detailed Description

Encapsulates a context-related error.

### 14.15.2 Property Documentation

#### 14.15.2.1 string Coherent.UI.ContextError.Error [get], [set]

Error description.

#### 14.15.2.2 ContextErrorType Coherent.UI.ContextError.ErrorCode [get], [set]

Indicates an error code.

## 14.16 Coherent.UI.ContextListener Class Reference

Abstract interface to listen to Context-related events.

Inherits IDisposable.

### Public Member Functions

- virtual void ContextReady ()

  *Called when the context is fully initialized. Creating Views is now permitted.*
- virtual void OnError (ContextError arg0)

  *Called when an error in the context occurs.*
- virtual bool OnDownloadStarted (Download downloadItem)

  *Called when the context 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.*
- virtual void OnSubscriptionCheckDone (SubscriptionError arg0)

  *Called when the check for the subscription version of the product is completed.*
- virtual void OnCookiesSaved (string cookiesResource)

  *Called when the write request for the cookies resource was completed.*

### 14.16.1 Detailed Description

Abstract interface to listen to Context-related events.

### 14.16.2 Member Function Documentation

#### 14.16.2.1 virtual void Coherent.UI.ContextListener.ContextReady ( ) [inline], [virtual]

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

#### 14.16.2.2 virtual void Coherent.UI.ContextListener.OnCookiesSaved ( string *cookiesResource* ) [inline], [virtual]

Called when the write request for the cookies resource was completed.

#### 14.16.2.3 virtual void Coherent.UI.ContextListener.OnDownloadComplete ( Download *downloadItem,* System.IntPtr *data,* uint *size* ) [inline], [virtual]

Called when a download is complete.

#### 14.16.2.4 virtual void Coherent.UI.ContextListener.OnDownloadFailed ( Download *downloadItem,* DownloadErrorType *error* ) [inline], [virtual]

Called when a download has failed.

#### 14.16.2.5 virtual bool Coherent.UI.ContextListener.OnDownloadStarted ( Download *downloadItem* ) [inline], [virtual]

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

#### 14.16.2.6 virtual void Coherent.UI.ContextListener.OnError ( ContextError *arg0* ) [inline], [virtual]

Called when an error in the context occurs.

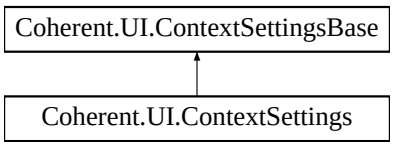
#### 14.16.2.7 virtual void Coherent.UI.ContextListener.OnSubscriptionCheckDone ( SubscriptionError *arg0* ) [inline], [virtual]

Called when the check for the subscription version of the product is completed.

## 14.17 Coherent.UI.ContextSettings Class Reference

Encapsulates the settings of the

Inheritance diagram for Coherent.UI.ContextSettings:




### Properties

- 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 CustomUserAgentString [get, set]

  *The user agent string that will be used if you select "UAT_Custom" for UserAgent.*

### 14.17.1 Detailed Description

Encapsulates the settings of the

### 14.17.2 Property Documentation

#### 14.17.2.1 new string Coherent.UI.ContextSettings.CachePath [get], [set]

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

#### 14.17.2.2 new string Coherent.UI.ContextSettings.CookiesResource [get], [set]

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

#### 14.17.2.3 new string Coherent.UI.ContextSettings.CustomUserAgentString [get], [set]

The user agent string that will be used if you select "UAT_Custom" for UserAgent.

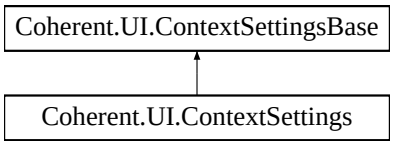
#### 14.17.2.4 new string Coherent.UI.ContextSettings.HTML5LocalStoragePath [get], [set]

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

## 14.18 Coherent.UI.ContextSettingsBase Class Reference

Encapsulates the settings of the

Inheritance diagram for Coherent.UI.ContextSettingsBase:




### 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 EnableClientAudioPlayback [get, set]

  *Enable notifications for client audio playback. When this setting is enabled, the ViewListeners will receive notifications for audio streams and the client can request more PCM data using the corresponding*
- 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 SubscriptionEmail [get, set]

  *The client e-mail for the subscription version of the product. Leave empty for production builds and non-subscription versions.*
- string SubscriptionKey [get, set]

  *The client activation key for the subscription version of the product. Leave empty for production builds and non-subscription versions.*
- string AdditionalHostOptions [get, set]

  *Additional host options that should be supplied to the*

### 14.18.1 Detailed Description

Encapsulates the settings of the

### 14.18.2 Property Documentation

#### 14.18.2.1 string Coherent.UI.ContextSettingsBase.AdditionalHostOptions [get], [set]

Additional host options that should be supplied to the

#### 14.18.2.2 bool Coherent.UI.ContextSettingsBase.AllowCookies [get], [set]

Enables/disables cookies when accessing the web.

#### 14.18.2.3 int Coherent.UI.ContextSettingsBase.DebuggerPort [get], [set]

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

#### 14.18.2.4 bool Coherent.UI.ContextSettingsBase.DisableWebSecurity [get], [set]

Disable Same-Origin Policy. Use with caution.

#### 14.18.2.5 bool Coherent.UI.ContextSettingsBase.EnableClientAudioPlayback [get], [set]

Enable notifications for client audio playback. When this setting is enabled, the ViewListeners will receive notifications for audio streams and the client can request more PCM data using the corresponding

#### 14.18.2.6 bool Coherent.UI.ContextSettingsBase.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.

#### 14.18.2.7 bool Coherent.UI.ContextSettingsBase.ForceDisablePluginFullscreen [get], [set]

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

#### 14.18.2.8 string Coherent.UI.ContextSettingsBase.SubscriptionEmail [get], [set]

The client e-mail for the subscription version of the product. Leave empty for production builds and non-subscription versions.

#### 14.18.2.9 string Coherent.UI.ContextSettingsBase.SubscriptionKey [get], [set]

The client activation key for the subscription version of the product. Leave empty for production builds and non-subscription versions.

#### 14.18.2.10 UserAgentType Coherent.UI.ContextSettingsBase.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.

## 14.19 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-context per-session and can be used to distinguish downloads poinitng 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.*

### 14.19.1 Detailed Description

Encapsulates a download task.

### 14.19.2 Member Function Documentation

#### 14.19.2.1 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

#### 14.19.2.2 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.

#### 14.19.2.3 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

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

The name of the downloaded file.

**Returns**

the name of the file being doenloaded

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

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

**Returns**

a unique id of the download

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

The referrer of the download.

**Returns**

the referrer

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

The estimated size of the download.

**Returns**

the estimated size in bytes

#### 14.19.2.8 virtual string Coherent.UI.Download.GetURL ( ) [inline], [virtual]

The download url.

**Returns**

the url to download

#### 14.19.2.9 virtual bool Coherent.UI.Download.HasFailed ( ) [inline],[virtual]

Tells if the download has failed.

**Returns**

if the download has failed

#### 14.19.2.10 virtual bool Coherent.UI.Download.IsCancelled ( ) [inline],[virtual]

Tells if the download has been cancelled.

**Returns**

if the download has been cancelled

#### 14.19.2.11 virtual bool Coherent.UI.Download.IsComplete ( ) [inline],[virtual]

Tells if the download is complete.

**Returns**

if the download is complete

## 14.20 Coherent.UI.EventModifiersState Class Reference

The state of the key modifiers when an event happens.

Inherits IDisposable.

### 14.20.1 Detailed Description

The state of the key modifiers when an event happens.

## 14.21 Coherent.UI.EventMouseModifiersState Class Reference

The state of the mouse modifiers when an event happens.

Inherits IDisposable.

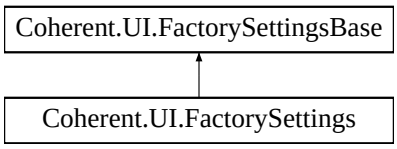
### 14.21.1 Detailed Description

The state of the mouse modifiers when an event happens.

## 14.22 Coherent.UI.FactorySettings Class Reference

Encapsulates the settings of the UIFactory.

Inheritance diagram for Coherent.UI.FactorySettings:




### Properties

- new string HostDirectory [get, set]

  *Points to the folder where the host application and it's libraries*

- new string MinidumpFolder [get, set]

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

- new string AppGroupName [get, set]

  *Set application group name (Mac OS only) If you want to create an sandboxed app, you must set this to the application group identifier*

### 14.22.1 Detailed Description

Encapsulates the settings of the UIFactory.

### 14.22.2 Property Documentation

#### 14.22.2.1 new string Coherent.UI.FactorySettings.AppGroupName [get], [set]

Set application group name (Mac OS only) If you want to create an sandboxed app, you must set this to the application group identifier

#### 14.22.2.2 new string Coherent.UI.FactorySettings.HostDirectory [get], [set]

Points to the folder where the host application and it's libraries

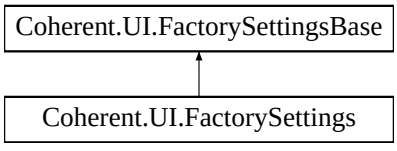
#### 14.22.2.3 new string Coherent.UI.FactorySettings.MinidumpFolder [get], [set]

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

## 14.23 Coherent.UI.FactorySettingsBase Class Reference

Encapsulates the settings of the UIFactory.

Inheritance diagram for Coherent.UI.FactorySettingsBase:



### Properties

- 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 repspective rights holder.*

#### 14.23.1 Detailed Description

Encapsulates the settings of the UIFactory.

#### 14.23.2 Property Documentation

##### 14.23.2.1 bool Coherent.UI.FactorySettingsBase.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 repspective rights holder.

##### 14.23.2.2 bool Coherent.UI.FactorySettingsBase.WriteMinidumps [get],[set]

If a

## 14.24 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, URLRequestBase request, ResourceResponse response)

  *Requests to read a resource.*
- virtual void WriteFile (string url, ResourceData resource)

  *Request to write to a resource.*

#### 14.24.1 Detailed Description

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

#### 14.24.2 Member Function Documentation

##### 14.24.2.1 virtual void Coherent.UI.FileHandler.ReadFile ( string *url,* URLRequestBase *request,* ResourceResponse *response* ) [inline],[virtual]

Requests to read a resource.

**Parameters**

| | |
|---|---|
| *url* | the coui url of the resource |
| *request* | the request object for the resource. |
| *response* | the response object for the resource |
| *response* | the response to be filled with the resource data or notified for failure |

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

Request to write to a resource.

**Parameters**

| | |
|---|---|
| *url* | the coui url of the resource. |
| *resource* | the resource data that has to be written |

## 14.25 Coherent.UI.FileSelectionParams Class Reference

Parameters for file selection request.

Inherits IDisposable.

### Properties

- string[] AcceptedMIMETypes [get]

  *The list of street addresses of the principal*
- FileSelectionMode Mode [get]

  *The file selection mode - it determines whether multiple files or folders can be selected.*
- string DialogTitle [get, set]

  *Title for the dialog.*
- string DefaultFileName [get]

  *Default file name to select in the dialog.*
- uint AcceptedMIMETypesCount [get]

  *The count of items in the AcceptedMIMETypes array.*

### 14.25.1 Detailed Description

Parameters for file selection request.

### 14.25.2 Property Documentation

#### 14.25.2.1 string [] Coherent.UI.FileSelectionParams.AcceptedMIMETypes [get]

The list of street addresses of the principal

#### 14.25.2.2 uint Coherent.UI.FileSelectionParams.AcceptedMIMETypesCount [get]

The count of items in the AcceptedMIMETypes array.

#### 14.25.2.3 string Coherent.UI.FileSelectionParams.DefaultFileName [get]

Default file name to select in the dialog.

#### 14.25.2.4 string Coherent.UI.FileSelectionParams.DialogTitle [get], [set]

Title for the dialog.

#### 14.25.2.5 FileSelectionMode Coherent.UI.FileSelectionParams.Mode [get]

The file selection mode - it determines whether multiple files or folders can be selected.

## 14.26 Coherent.UI.FileSelectRequest Class Reference

File selection request.

Inherits IDisposable.

### Public Member Functions

- virtual FileSelectionParams GetFileSelectionParams ()

  *Get the file selection params for the current request. These are useful for creating the file chooser dialog with accurate params - what mimeTypes are accepted, whether user can choose multiple files or directories, etc.*
- virtual void Add (string fullPath)

  *Adds a file to the current request.*
- virtual void Finish ()

  *Complete the request. This method should be called when you are done with adding files to this file selection request.*

### 14.26.1 Detailed Description

File selection request.

### 14.26.2 Member Function Documentation

#### 14.26.2.1 virtual void Coherent.UI.FileSelectRequest.Add ( string *fullPath* ) [inline], [virtual]

Adds a file to the current request.

**Parameters**

| | |
|---|---|
| *fullPath* | The full path to the chosen file or directory |

#### 14.26.2.2 virtual void Coherent.UI.FileSelectRequest.Finish ( ) [inline], [virtual]

Complete the request. This method should be called when you are done with adding files to this file selection request.

#### 14.26.2.3 virtual FileSelectionParams Coherent.UI.FileSelectRequest.GetFileSelectionParams ( ) [inline], [virtual]

Get the file selection params for the current request. These are useful for creating the file chooser dialog with accurate params - what mimeTypes are accepted, whether user can choose multiple files or directories, etc.

**Returns**

the file selection params for this request

## 14.27 Coherent.UI.HTTPHeader Class Reference

Encapsulates a HTTP header field with its content.

Inherits IDisposable.

### Public Types

- enum FieldType {
  FieldType.Accept,
  FieldType.AcceptCharset,
  FieldType.AcceptEncoding,
  FieldType.AcceptLanguage,
  FieldType.AcceptRanges,
  FieldType.Authorization,
  FieldType.CacheControl,
  FieldType.Connection,
  FieldType.ContentLength,
  FieldType.ContentRange,
  FieldType.ContentType,
  FieldType.Cookie,
  FieldType.Host,
  FieldType.IfModifiedSince,
  FieldType.IfNoneMatch,
  FieldType.IfRange,
  FieldType.Origin,
  FieldType.Pragma,
  FieldType.ProxyAuthorization,
  FieldType.ProxyConnection,
  FieldType.Range,
  FieldType.Referer,
  FieldType.UserAgent,
  FieldType.TransferEncoding }

  *Enumeration that defines various HTTP header types.*

### Static Public Member Functions

- static string GetHeaderField (HTTPHeader.FieldType type)

  *Gets a string representation of the requested header field type.*

### Properties

- string Field [get, set]

  *The name (key) of the HTTP header field.*

- string Content [get, set]

  *The value of the HTTP header field.*

### 14.27.1 Detailed Description

Encapsulates a HTTP header field with its content.

### 14.27.2 Member Enumeration Documentation

#### 14.27.2.1 enum Coherent.UI.HTTPHeader.FieldType

Enumeration that defines various HTTP header types.

**Enumerator**

***Accept*** Key for the Accept header field.

***AcceptCharset*** Key for the AcceptCharset header field.

***AcceptEncoding*** Key for the AcceptEncoding header field.

***AcceptLanguage*** Key for the AcceptLanguage header field.

***AcceptRanges*** Key for the AcceptRanges header field.

***Authorization*** Key for the Authorization header field.

***CacheControl*** Key for the CacheControl header field.

***Connection*** Key for the Connection header field.

***ContentLength*** Key for the ContentLength header field.

***ContentRange*** Key for the ContentRange header field.

***ContentType*** Key for the ContentType header field.

***Cookie*** Key for the Cookie header field.

***Host*** Key for the Host header field.

***IfModifiedSince*** Key for the IfModifiedSince header field.

***IfNoneMatch*** Key for the IfNoneMatch header field.

***IfRange*** Key for the IfRange header field.

***Origin*** Key for the Origin header field.

***Pragma*** Key for the Pragma header field.

***ProxyAuthorization*** Key for the ProxyAuthorization header field.

***ProxyConnection*** Key for the ProxyConnection header field.

***Range*** Key for the Range header field.

***Referer*** Key for the Referer header field.

***UserAgent*** Key for the UserAgent header field.

***TransferEncoding*** Key for the TransferEncoding header field.

### 14.27.3 Member Function Documentation

#### 14.27.3.1 static string Coherent.UI.HTTPHeader.GetHeaderField ( HTTPHeader.FieldType *type* ) [inline], [static]

Gets a string representation of the requested header field type.

**Parameters**

| | |
|---|---|
| *type* | the HTTP header field type |

**Returns**

a string representation for the requested header field type

### 14.27.4 Property Documentation

#### 14.27.4.1 string Coherent.UI.HTTPHeader.Content [get], [set]

The value of the HTTP header field.

#### 14.27.4.2 string Coherent.UI.HTTPHeader.Field [get], [set]

The name (key) of the HTTP header field.

## 14.28 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.*

### 14.28.1 Detailed Description

Interface to allow custom logging.

### 14.28.2 Member Function Documentation

#### 14.28.2.1 virtual void Coherent.UI.ILogHandler.Assert ( string *message* ) [inline], [virtual]

Called when an assert is triggered.

**Parameters**

| | |
|---|---|
| *message* | a message that describes the reason for the assertion |

#### 14.28.2.2 virtual void Coherent.UI.ILogHandler.WriteLog ( Severity *severity,* string *message,* uint *length* ) [inline], [virtual]

Called when a log message has to be written.

**Parameters**

| | |
|---|---|
| *severity* | the severity of the message |
| *message* | the log message itself |
| *length* | the length of the message |

## 14.29 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*

### 14.29.1 Detailed Description

This class represents a link to a

### 14.29.2 Member Function Documentation

#### 14.29.2.1 virtual void Coherent.UI.ImageData.Destroy ( ) [inline], [virtual]

Destroys the

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

Updates the content of the

**Parameters**

| | |
|---|---|
| *data* | a pointer to the new data to copy in the object the size of the data must equal to the total size of the ImageData |
| *flipY* | tells if to flip vertically when copying |

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

Thrown when casting a Value to an incompatible type

Inherits InvalidCastException.

### 14.30.1 Detailed Description

Thrown when casting a Value to an incompatible type

## 14.31 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.*

### 14.31.1 Detailed Description

A keyboard event.

### 14.31.2 Property Documentation

#### 14.31.2.1 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.

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

Is it a key from the num-pad.

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

The key code.

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

The current key modifiers.

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

The type of the event.

## 14.32 Coherent.UI.MediaStreamDevice Class Reference

Represents a media stream device.

Inherits IDisposable.

### Properties

- MediaStreamType Type [get]

  *type of the device*
- string DeviceId [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.*

### 14.32.1 Detailed Description

Represents a media stream device.

### 14.32.2 Property Documentation

#### 14.32.2.1 string Coherent.UI.MediaStreamDevice.DeviceId [get]

Null terminated string with the unique id of the device.

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

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

#### 14.32.2.3 MediaStreamType Coherent.UI.MediaStreamDevice.Type [get]

type of the device

## 14.33 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.*

### 14.33.1 Detailed Description

Represents a request for media stream.

### 14.33.2 Member Function Documentation

#### 14.33.2.1 void Coherent.UI.MediaStreamRequest.Respond ( uint[] *devices* ) [inline]

Respond to the media stream request.

**Parameters**

| | |
|---|---|
| *devices* | 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::ViewContext instance.

### 14.33.3 Property Documentation

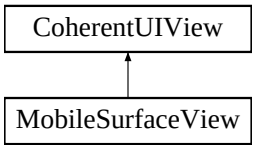
#### 14.33.3.1 MediaStreamDevice [] Coherent.UI.MediaStreamRequest.Devices [get]

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

## 14.34 MobileSurfaceView Class Reference

Component containing a Coherent UI view for surfaces on mobile platforms.

Inheritance diagram for MobileSurfaceView:




### Public Member Functions

- bool UpdateView ()

  *Update the view. This method triggers the view to be rendered to the underlaying texture.*

### Additional Inherited Members

### 14.34.1 Detailed Description

Component containing a Coherent UI view for surfaces on mobile platforms.

### 14.34.2 Member Function Documentation

#### 14.34.2.1 bool MobileSurfaceView.UpdateView ( ) [inline]

Update the view. This method triggers the view to be rendered to the underlaying texture.

**Returns**

true if the view was actually updated

## 14.35 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.*

### 14.35.1 Detailed Description

A mouse event.

### 14.35.2 Property Documentation

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

The mouse button that triggered the event.

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

Key modifiers.

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

Mouse modifiers.

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

The type of the event.

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

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

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

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

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

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

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

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

## 14.36 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.*

### 14.36.1 Detailed Description

Abstract interface providing data for storing resources.

### 14.36.2 Member Function Documentation

#### 14.36.2.1 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

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

Get the size of the data.

**Returns**

the size of the data

#### 14.36.2.3 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.

#### 14.36.2.4 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.

## 14.37 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.*
- virtual void SetResponseHeader (string key, string value)

  *Set a key-value pair in the response headers.*
- virtual void SetStatus (int status)

  *Set the status for the response.*

### 14.37.1 Detailed Description

Abstract interface for responding to read resource requests.

### 14.37.2 Member Function Documentation

#### 14.37.2.1 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**

| | |
|---|---|
| *size* | the size of the resource |

**Returns**

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

#### 14.37.2.2 virtual void Coherent.UI.ResourceResponse.SetResponseHeader ( string *key*, string *value* ) [inline], [virtual]

Set a key-value pair in the response headers.

**Parameters**

| | |
|---|---|
| *key* | the key for the response header |
| *value* | the value for the response header |

#### 14.37.2.3 virtual void Coherent.UI.ResourceResponse.SetStatus ( int *status* ) [inline], [virtual]

Set the status for the response.

**Parameters**

| | |
|---|---|
| *status* | the response status |

#### 14.37.2.4 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.

#### 14.37.2.5 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.

## 14.38 Coherent.UI.SubscriptionError Class Reference

Encapsulates a subscription-check-related error.

Inherits IDisposable.

### Properties

- SubscriptionErrorType ErrorCode [get, set]
  *Indicates an error code.*
- string Error [get, set]
  *Error description.*

### 14.38.1 Detailed Description

Encapsulates a subscription-check-related error.

### 14.38.2 Property Documentation

#### 14.38.2.1 string Coherent.UI.SubscriptionError.Error [get], [set]

Error description.

#### 14.38.2.2 SubscriptionErrorType Coherent.UI.SubscriptionError.ErrorCode [get], [set]

Indicates an error code.

## 14.39 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 context.*

### 14.39.1 Detailed Description

Interface that signals for creation of rendering surfaces.

### 14.39.2 Member Function Documentation

#### 14.39.2.1 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 context.

**Parameters**

| | |
|---|---|
| *handle* | the created surface handle or a NULL handle if the operation fails |

## 14.40 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)*

### 14.40.1 Detailed Description

A touch event.

### 14.40.2 Property Documentation

#### 14.40.2.1 bool Coherent.UI.TouchEventData.HasRadiusInformation `[get],[set]`

Flags if there is contact radius information for this event.

#### 14.40.2.2 uint Coherent.UI.TouchEventData.Id `[get],[set]`

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

#### 14.40.2.3 uint Coherent.UI.TouchEventData.RadiusX `[get],[set]`

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

#### 14.40.2.4 uint Coherent.UI.TouchEventData.RadiusY `[get],[set]`

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

#### 14.40.2.5 TouchEventData.EventType Coherent.UI.TouchEventData.Type `[get],[set]`

The type of the touch event.

#### 14.40.2.6 int Coherent.UI.TouchEventData.X `[get],[set]`

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

#### 14.40.2.7 int Coherent.UI.TouchEventData.Y `[get],[set]`

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

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

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

Inherits Exception.

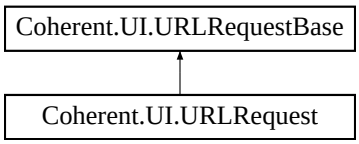
### 14.41.1 Detailed Description

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

## 14.42 Coherent.UI.URLRequest Class Reference

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

Inheritance diagram for Coherent.UI.URLRequest:




### Public Member Functions

- void SetExtraHeaders (HTTPHeader[] headers)

  *Set the extra headers.*
- virtual void SetURL (string url)

  *Replace the URL of the request.*
- virtual void SetMethod (string method)

  *Set the method 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 void Deny ()

  *Deny the request.*

### 14.42.1 Detailed Description

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

### 14.42.2 Member Function Documentation

#### 14.42.2.1 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**

| | |
|---|---|
| *field* | name of the header |
| *content* | content of the header |

#### 14.42.2.2 virtual void Coherent.UI.URLRequest.Deny ( ) [inline], [virtual]

Deny the request.

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

Remove extra header from the request.

**Parameters**

| | |
|---|---|
| *field* | field of the header to be removed |

#### 14.42.2.4 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**

| | |
|---|---|
| *field* | name of the header |
| *content* | new content of the header |

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

Set the extra headers.

**Parameters**

| | |
|---|---|
| *headers* | the array of new headers |

#### 14.42.2.6 virtual void Coherent.UI.URLRequest.SetMethod ( string *method* ) [inline], [virtual]

Set the method of the request.

**Parameters**

| | |
|---|---|
| *method* | the new method as string. method must be a valid HTTP method name |

#### 14.42.2.7 virtual void Coherent.UI.URLRequest.SetReferrer ( string *referrer* ) [inline], [virtual]

Set the referrer of the request.

**Parameters**

| | |
|---|---|
| *referrer* | the new referrer URL for the request |

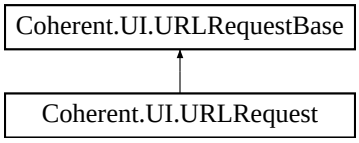
#### 14.42.2.8 virtual void Coherent.UI.URLRequest.SetURL ( string *url* ) [inline], [virtual]

Replace the URL of the request.

## 14.43 Coherent.UI.URLRequestBase Class Reference

represents a single URL request (read-only) Allows monitoring of URL requests

Inheritance diagram for Coherent.UI.URLRequestBase:




### Public Member Functions

- HTTPHeader[] GetExtraHeaders ()
  *Get extra request headers*
- virtual string GetURL ()
  *Get the URL of the request.*
- virtual string GetMethod ()
  *Get the method of the request.*
- virtual string GetReferrer ()
  *Get the referrer of the request.*
- virtual string GetExtraHeader (string field)
  *Get extra header value.*
- virtual int GetExtraHeaderIndex (string field)
  *Get extra header value.*
- virtual bool HasExtraHeader (string field)
  *Checks if a header with the specified key is present in the headers collection.*

### 14.43.1 Detailed Description

represents a single URL request (read-only) Allows monitoring of URL requests

### 14.43.2 Member Function Documentation

#### 14.43.2.1 virtual string Coherent.UI.URLRequestBase.GetExtraHeader ( string *field* ) [inline],[virtual]

Get extra header value.

**Parameters**

| | |
|---|---|
| *field* | field of the header |

**Returns**

the value of the header or nullptr if there is such header in the request

#### 14.43.2.2 virtual int Coherent.UI.URLRequestBase.GetExtraHeaderIndex ( string *field* ) [inline],[virtual]

Get extra header value.

**Parameters**

| | |
|---|---|
| *field* | field of the header |

**Returns**

the index of the header with the specified key in the headers collection, or a negative number if no such key exists

#### 14.43.2.3 HTTPHeader [] Coherent.UI.URLRequestBase.GetExtraHeaders ( ) [inline]

Get extra request headers

**Returns**

array of the extra request headers

#### 14.43.2.4 virtual string Coherent.UI.URLRequestBase.GetMethod ( ) [inline], [virtual]

Get the method of the request.

**Returns**

the method as a string

#### 14.43.2.5 virtual string Coherent.UI.URLRequestBase.GetReferrer ( ) [inline], [virtual]

Get the referrer of the request.

**Returns**

the referrer of the request

#### 14.43.2.6 virtual string Coherent.UI.URLRequestBase.GetURL ( ) [inline], [virtual]

Get the URL of the request.

**Returns**

the URL of the request

#### 14.43.2.7 virtual bool Coherent.UI.URLRequestBase.HasExtraHeader ( string *field* ) [inline], [virtual]

Checks if a header with the specified key is present in the headers collection.

**Parameters**

| | |
|---|---|
| *field* | field of the header |

**Returns**

true if a header with the specified key is present, false otherwise

## 14.44 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 (char value)
  *Create a char 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 char (Value value)

*cast a value to a char*

- 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*

### 14.44.1 Detailed Description

Type for representing generic JavaScript values

### 14.44.2 Constructor & Destructor Documentation

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

Create a null value

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

Create a boolean value

**Parameters**

| | |
|---|---|
| *value* | the integer value of the Value object |

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

Create an integer value

**Parameters**

| | |
|---|---|
| *value* | the integer value of the Value object |

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

Create an unsigned integer value

**Parameters**

| | |
|---|---|
| *value* | the unsigned integer value of the Value object |

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

Create a signed byte value

**Parameters**

| | |
|---|---|
| *value* | the signed byte value of the Value object |

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

Create a byte value

**Parameters**

| | |
|---|---|
| *value* | the byte value of the Value object |

#### 14.44.2.7 Coherent.UI.Binding.Value.Value ( char *value* ) [inline]

Create a char value

**Parameters**

| | |
|---|---|
| *value* | the char value of the Value object |

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

Create a float value

**Parameters**

| | |
|---|---|
| *value* | the float value of the Value object |

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

Create a double value

**Parameters**

| | |
|---|---|
| *value* | the double value of the Value object |

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

Create a string value

**Parameters**

| | |
|---|---|
| *value* | the string value of the Value object |

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

Create an array value

**Parameters**

| | |
|---|---|
| *value* | the array value of the Value object |

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

Create an object value

**Parameters**

| | |
|---|---|
| *value* | dictionary of properties of the Value object |

#### 14.44.2.13 Coherent.UI.Binding.Value.Value ( ValueObject *value* ) [inline]

Create an object value

**Parameters**

| | |
|---|---|
| *value* | dictionary of properties of the Value object |

### 14.44.3 Member Function Documentation

#### 14.44.3.1 override bool Coherent.UI.Binding.Value.Equals ( object *obj* ) [inline]

Compare to another object

**Parameters**

| | |
|---|---|
| *obj* | |

**Returns**

true if obj is a Value instance equal to this one

#### 14.44.3.2 override int Coherent.UI.Binding.Value.GetHashCode ( ) [inline]

Hash code for Value

**Returns**

hash code for the current instance

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

cast a value to a boolean

**Parameters**

| | |
|---|---|
| *value* | value to be casted |

**Returns**

the boolean stored in the value

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

cast a value to a byte

**Parameters**

| | |
|---|---|
| *value* | value to be cast |

**Returns**

the byte stored in the value

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

cast a value to a char

**Parameters**

| | |
|---|---|
| *value* | value to be cast |

**Returns**

the char stored in the value

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

cast a value to a dictionary of properties

**Parameters**

| | |
|---|---|
| *value* | value to be casted |

**Returns**

the dictionary of properties of the value

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

cast a value to a double

**Parameters**

| | |
|---|---|
| *value* | value to be casted |

**Returns**

the double stored in the value

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

cast a value to a float

**Parameters**

| | |
|---|---|
| *value* | value to be casted |

**Returns**

the float stored in the value

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

cast a value to an int

**Parameters**

| | |
|---|---|
| *value* | value to be casted |

**Returns**

the int stored in the value

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

cast a value to a signed byte

**Parameters**

| | |
|---|---|
| *value* | value to be cast |

**Returns**

the signed byte stored in the value

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

cast a value to a string

**Parameters**

| | |
|---|---|
| *value* | value to be casted |

**Returns**

the string stored in the value

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

cast a value to an uint

**Parameters**

| | |
|---|---|
| *value* | value to be casted |

**Returns**

the unsigned int stored in the value

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

cast a value to an array

**Parameters**

| | |
|---|---|
| *value* | value to be casted |

**Returns**

the array of Value objects stored in the value

#### 14.44.3.14 static bool Coherent.UI.Binding.Value.operator!= ( Value *lhs*, Value *rhs* ) [inline], [static]

Compares two Value objects

**Parameters**

| | |
|---|---|
| *lhs* | object to be compared |
| *rhs* | object to be compared |

**Returns**

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

#### 14.44.3.15 static bool Coherent.UI.Binding.Value.operator== ( Value *lhs*, Value *rhs* ) [inline],[static]

Compares two Value objects

**Parameters**

| | |
|---|---|
| *lhs* | object to be compared |
| *rhs* | object to be compared |

**Returns**

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

### 14.44.4 Member Data Documentation

#### 14.44.4.1 ValueType Coherent.UI.Binding.Value.Type

The type of the data stored in this Value instance

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

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

Inherits IDictionary< string, Value >, and IDictionary.

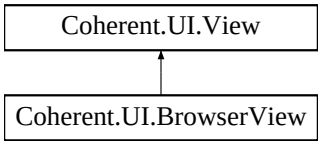
### 14.45.1 Detailed Description

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

## 14.46 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 ViewContext GetContext ()

  *Get the view context.*
- 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 This function must be called on the ViewContext thread. This overload let's Coherent UI control the time in the view. You should have set ViewInfo::ControlTimeOnDemand to false to use it. If you need to control timing - use the other overload and create the view with ViewInfo::ControlTimeOnDemand set to true.*
- 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 This function must be called on the ViewContext thread. This overload let's Coherent UI control the time in the view. You should have set ViewInfo::ControlTimeOnDemand to false to use it. If you need to control timing - use the other overload and create the view with ViewInfo::ControlTimeOnDemand set to true.*
- 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 context'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*
- virtual int GetAudioData (int streamId, IntPtr buffer, int bufferSize, int timeoutMs)

  *Obtains audio data for the specified stream in the current view and copies it into the client-provided buffer.*
- virtual void SetZoomLevel (double zoomLevel)

  *Sets the page zoom level. Specifying 0.0 will reset to the default zoom level. The API works with "levels", not factors. Each level represents 20% zoom. Negative values mean "zoom out", while positive values mean "zoom in". You can pass fractional levels as well. If you want to convert a factor to a level that you can pass to the API, you can use the following formula: level = logN(1.2, factor). In terms of C++ that would be the same as "log(factor) / log(1.2)". For example, if you want to make the content 50% smaller, you'll need a level of logN(1.2, 0.5) ~= -3,8. If you want to make it 50% larger, you'll have logN(1.2, 1.5) ~= 2.2. The zoom factor operates on protocol and host level, meaning that all views with the same protocol and host as the current URL will be scaled as well.*
- virtual void SetMasterVolume (double volume)

  *Sets the master volume for all audio and video streams.*

### 14.46.1 Detailed Description

Class that encapsulates a UI

### 14.46.2 Member Function Documentation

#### 14.46.2.1 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**

| | |
|---|---|
| *username* | the username |
| *password* | the password |
| *success* | defines whether the response is valid, i.e. if the request should be answered or canceled |

#### 14.46.2.2 BoundEventHandle Coherent.UI.View.BindCall ( string *name,* System.Delegate *handler* ) [inline]

Expose C++ handler to be called from UI.

**Parameters**

| | |
|---|---|
| *name* | name for the handler in the UI |
| *handler* | handler to be executed |

**Returns**

bound handle so the handler can be unbound later

#### 14.46.2.3 virtual ImageData Coherent.UI.View.CreateImageData ( string *name,* int *width,* int *height,* IntPtr *data,* bool *flipY* ) [inline], [virtual]

Create an

**Parameters**

| | |
|---|---|
| *name* | the name of the image data object |
| *width* | the width of the image |
| *height* | the height of the image |
| *data* | initial data to fill the image data with |
| *flipY* | flips vertically the content of the source image |

**Returns**

the image data object created in the DOM of this View

#### 14.46.2.4 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

#### 14.46.2.5 virtual void Coherent.UI.View.DownloadUrl ( string *path* ) [inline], [virtual]

Initiates a download request on the specified path.

#### 14.46.2.6 virtual ViewErrorType Coherent.UI.View.DrawSurface ( ) [inline], [virtual]

[EXPERIMENTAL - DO NOT USE]

#### 14.46.2.7 virtual void Coherent.UI.View.ExecuteScript ( string *script* ) [inline], [virtual]

Execute arbitrary JavaScript code.

**Parameters**

| | |
|---|---|
| *script* | code to be evaluated in the context of the main frame of the view |

#### 14.46.2.8 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

#### 14.46.2.9 virtual ViewErrorType Coherent.UI.View.FetchSurface ( ) [inline], [virtual]

Returns the requested surface. Works only for on-demand views. All buffered view should use the context's FetchSurfaces method. This call will block until the surface is ready. (available only in Full versions of

**Returns**

the outcome of the operation

#### 14.46.2.10 virtual int Coherent.UI.View.GetAudioData ( int *streamId,* IntPtr *buffer,* int *bufferSize,* int *timeoutMs* ) [inline], [virtual]

Obtains audio data for the specified stream in the current view and copies it into the client-provided buffer.

**Parameters**

| | |
|---|---|
| *streamId* | the stream for which to get PCM data |
| *buffer* | the client buffer into which the data is copied |
| *bufferSize* | the size of the client buffer |
| *timeoutMs* | the timeout in milliseconds to wait for audio data. Use 0 to wait indefinitely. |

**Returns**

the actual number of bytes read. A negative number is returned if the stream is exhausted or does not exist.

#### 14.46.2.11 virtual float Coherent.UI.View.GetClickThroughAlphaThreshold ( ) [inline], [virtual]

Get the currently set alpha threshold for click-through queries.

**Returns**

the currently set alpha threshold

#### 14.46.2.12 virtual ViewContext Coherent.UI.View.GetContext ( ) [inline], [virtual]

Get the view context.

**Returns**

pointer to the view context used for creating this view

#### 14.46.2.13 virtual string Coherent.UI.View.GetCurentViewPath ( ) [inline], [virtual]

Get the last requested or navigated-to path.

**Returns**

the path as null-terminated string

#### 14.46.2.14 virtual int Coherent.UI.View.GetHeight ( ) [inline], [virtual]

Gets the height of the

**Returns**

the height of the View

#### 14.46.2.15 virtual string Coherent.UI.View.GetLastFailedPath ( ) [inline], [virtual]

Get the last failed path for the

**Returns**

the path as null-terminated string

#### 14.46.2.16 virtual string Coherent.UI.View.GetLastLoadedPath ( ) [inline], [virtual]

Get the last successfully loaded path for the

**Returns**

the path as null-terminated string

#### 14.46.2.17 virtual string Coherent.UI.View.GetLastRequestedPath ( ) [inline], [virtual]

Get the last requested path to load for the

**Returns**

the path as null-terminated string

#### 14.46.2.18 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)

#### 14.46.2.19 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.

#### 14.46.2.20 virtual int Coherent.UI.View.GetWidth ( ) [inline], [virtual]

Gets the width of the

**Returns**

the width of the View

#### 14.46.2.21 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.

#### 14.46.2.22 virtual void Coherent.UI.View.IMEActivate ( bool *active* ) [inline], [virtual]

Sets if the

**Parameters**

| | |
|---|---|
| *active* | whether to activate IME handling for this view |

#### 14.46.2.23 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)

#### 14.46.2.24 virtual bool Coherent.UI.View.IMEConfirmComposition ( string *composition* ) [inline], [virtual]

Confirms the current IME composition.

**Parameters**

| | |
|---|---|
| *composition* | the string to confirm |

**Returns**

the outcome of the operation (will fail if IME is inactive for this view)

#### 14.46.2.25 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

#### 14.46.2.26 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**

| | |
|---|---|
| *composition* | the composition string so far |
| *the* | cursor position in the composition |
| *targetStart* | the index at the start of the selection |
| *targetEnd* | the index at the end of the selection |

**Returns**

the outcome of the operation (will fail if IME is inactive for this view)

#### 14.46.2.27 virtual void Coherent.UI.View.InterceptURLRequests ( bool *intercept* ) [inline], [virtual]

Controls intercepting URL requests. For each request

**Parameters**

| | |
|---|---|
| *intercept* | whether to intercept all URL requests |

#### 14.46.2.28 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

#### 14.46.2.29 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

#### 14.46.2.30 virtual void Coherent.UI.View.IssueMouseOnUIQuery ( float *normX,* float *normY* ) [inline], [virtual]

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.

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**

| | |
|---|---|
| *normX* | the x coordinate of the mouse in the normalized space of the view [0..1] |

| | |
|---|---|
| *normY* | the y coordinate of the mouse in the normalized space of the view [0..1] |

#### 14.46.2.31 virtual bool Coherent.UI.View.IsTransparent ( ) [inline], [virtual]

Tells if this view is transparent or not.

**Returns**

true if the view is transparent

#### 14.46.2.32 virtual void Coherent.UI.View.KeyEvent ( KeyEventData *arg0* ) [inline], [virtual]

Sends a key event to the UI renderer.

**Parameters**

| | |
|---|---|
| *event* | the key event |

#### 14.46.2.33 virtual void Coherent.UI.View.KillFocus ( ) [inline], [virtual]

Removes the focus from this view.

#### 14.46.2.34 virtual void Coherent.UI.View.Load ( string *path* ) [inline], [virtual]

Loads a new path in the

**Parameters**

| | |
|---|---|
| *path* | the path to load |

#### 14.46.2.35 virtual void Coherent.UI.View.MouseActivate ( ) [inline], [virtual]

Sends a mouse event to the UI renderer.

#### 14.46.2.36 virtual void Coherent.UI.View.MouseEvent ( MouseEventData *arg0* ) [inline], [virtual]

Sends a mouse event to the UI renderer.

**Parameters**

| | |
|---|---|
| *event* | the mouse event |

#### 14.46.2.37 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

#### 14.46.2.38 virtual void Coherent.UI.View.Redraw ( ) [inline], [virtual]

Request a redraw of the current view.

#### 14.46.2.39 BoundEventHandle Coherent.UI.View.RegisterForEvent ( string *name,* System.Delegate *handler* ) [inline]

Expose C++ handler to be called when a specific event occurs.

**Parameters**

| | |
|---|---|
| *name* | name of the event |
| *handler* | handler to be executed |

**Returns**

bound handle so the handler can be unregistered

#### 14.46.2.40 virtual void Coherent.UI.View.Reload ( bool *ignoreCache* ) [inline], [virtual]

Reloads the last requested or navigated-to path.

**Parameters**

| | |
|---|---|
| *ignoreCache* | whether to ignore the cache for the reload |

#### 14.46.2.41 virtual void Coherent.UI.View.ReplyToJavaScriptMessage ( bool *success,* string *userInput* ) [inline], [virtual]

Sends a reply to a javascript message triggered for the view.

**Parameters**

| | |
|---|---|
| *success* | defines whether the user replies positively to the message (i.e. clicks the OK button) |
| *userInput* | the user reply message (only used in prompt messages) |

#### 14.46.2.42 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 This function must be called on the ViewContext thread. This overload let's Coherent UI control the time in the view. You should have set ViewInfo::ControlTimeOnDemand to false to use it. If you need to control timing - use the other overload and create the view with ViewInfo::ControlTimeOnDemand set to true.

**Returns**

the outcome of the operation

#### 14.46.2.43 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 This function must be called on the ViewContext thread. This overload let's Coherent UI control the time in the view. You should have set ViewInfo::ControlTimeOnDemand to false to use it. If you need to control timing - use the other overload and create the view with ViewInfo::ControlTimeOnDemand set to true.

**Returns**

the outcome of the operation

#### 14.46.2.44 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**

| | |
|---|---|
| *width* | the new width of the View |
| *height* | the new height of the View |

#### 14.46.2.45 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**

| | |
|---|---|
| *threshold* | sets the new alpha threshold |

#### 14.46.2.46 virtual void Coherent.UI.View.SetFocus ( ) [inline], [virtual]

Sets this view on focus.

#### 14.46.2.47 virtual void Coherent.UI.View.SetMasterVolume ( double *volume* ) [inline], [virtual]

Sets the master volume for all audio and video streams.

**Parameters**

| | |
|---|---|
| *volume* | coefficient between 0.0 and 1.0 |

#### 14.46.2.48 virtual void Coherent.UI.View.SetScriptError ( ScriptCallErrorType *error*, string *message* ) [inline], [virtual]

Sets an error for the currently executing handler.

**Parameters**

| | |
|---|---|
| *error* | the type of the error |
| *message* | the error message |

#### 14.46.2.49 virtual void Coherent.UI.View.SetTargetFramerate ( int *target* ) [inline], [virtual]

Sets the target frame-rate for the

**Parameters**

| | |
|---|---|
| *target* | the new target frame-rate |

#### 14.46.2.50 virtual void Coherent.UI.View.SetZoomLevel ( double *zoomLevel* ) [inline], [virtual]

Sets the page zoom level. Specifying 0.0 will reset to the default zoom level. The API works with "levels", not factors. Each level represents 20% zoom. Negative values mean "zoom out", while positive values mean "zoom in". You can pass fractional levels as well. If you want to convert a factor to a level that you can pass to the API, you can use the following formula: level = logN(1.2, factor). In terms of C++ that would be the same as "log(factor) / log(1.2)". For example, if you want to make the content 50% smaller, you'll need a level of logN(1.2, 0.5) ∼= -3,8. If you want to make it 50% larger, you'll have logN(1.2, 1.5) ∼= 2.2. The zoom factor operates on protocol and host level, meaning that all views with the same protocol and host as the current URL will be scaled as well.

#### 14.46.2.51 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

#### 14.46.2.52 virtual void Coherent.UI.View.TouchEvent ( TouchEventData *events*, uint *count* ) [inline], [virtual]

Sends a touch event to the UI renderer.

**Parameters**

| | |
|---|---|
| *events* | array of touch events |
| *count* | the count of the passed touch events in the array |

#### 14.46.2.53 virtual void Coherent.UI.View.TriggerEvent ( string *eventName* ) [inline], [virtual]

Triggers a UI event.

#### 14.46.2.54 void Coherent.UI.View.UnbindCall ( BoundEventHandle *handle* ) [inline]

Remove a bound C++ handler.

**Parameters**

| | |
|---|---|
| *handle* | the handle of the handler to be unbound |

#### 14.46.2.55 void Coherent.UI.View.UnbindObject ( object *target* ) [inline]

Remove all handlers bound to a specific object.

**Parameters**

| | |
|---|---|
| *object* | the object whose handlers will be removed |

#### 14.46.2.56 void Coherent.UI.View.UnregisterFromEvent ( BoundEventHandle *handle* ) [inline]

Remove a registered C++ handler.

**Parameters**

| | |
|---|---|
| *handle* | the handle of the event handler to be unregistered |

#### 14.46.2.57 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

## 14.47 Coherent.UI.ViewContext Class Reference

Encapsulates basic

Inherits IDisposable.

### Public Member Functions

- virtual bool Uninitialize ()

  *Frees all resources used by the event system. the ViewContext object is unusable after this call*
- virtual void Update ()

  *Communicates with the UI context process and performs all updates. Must be called in the thread that owns the context. 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 context 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 context has signaled that it is ready via Coherent::UI::ContextListener::-ContextReady*
- virtual void CreateView (ViewInfo info, string path, ViewListenerBase listener, string scriptToExecute)

  *Creates a new UI Must be called after the context has signaled that it is ready via Coherent::UI::ContextListener::-ContextReady*
- virtual bool SaveCookies ()

  *Saves the cookies to disk. Must be called after the context has signaled that it is ready via Coherent::UI::Context-Listener::ContextReady*
- virtual void AddCrossOriginWhitelistEntry (string sourceOrigin, string targetProtocol, string targetDomain, bool allowTargetSubdomains)

  *Add an entry to the cross-origin access whitelist. Must be called after the context has signaled that it is ready via Coherent::UI::ContextListener::ContextReady*
- virtual void RemoveCrossOriginWhitelistEntry (string sourceOrigin, string targetProtocol, string target-Domain, bool allowTargetSubdomains)

  *Remove an entry from the cross-origin access whitelist. Must be called after the context has signaled that it is ready via Coherent::UI::ContextListener::ContextReady*
- virtual void ClearCrossOriginWhitelist ()

  *Remove all entries from the cross-origin access whitelist. Must be called after the context has signaled that it is ready via Coherent::UI::ContextListener::ContextReady*

### 14.47.1 Detailed Description

Encapsulates basic

### 14.47.2 Member Function Documentation

#### 14.47.2.1 virtual void Coherent.UI.ViewContext.AddCrossOriginWhitelistEntry ( string *sourceOrigin,* string *targetProtocol,* string *targetDomain,* bool *allowTargetSubdomains* ) [inline], [virtual]

Add an entry to the cross-origin access whitelist. Must be called after the context has signaled that it is ready via Coherent::UI::ContextListener::ContextReady

#### 14.47.2.2 virtual void Coherent.UI.ViewContext.ClearCrossOriginWhitelist ( ) [inline], [virtual]

Remove all entries from the cross-origin access whitelist. Must be called after the context has signaled that it is ready via Coherent::UI::ContextListener::ContextReady

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

Creates a new UI Must be called after the context has signaled that it is ready via Coherent::UI::ContextListener::-ContextReady

**Parameters**

| | |
|---|---|
| *info* | an instance of the ViewInfo struct that describes all the parameters of the view |
| *path* | the page to load when the view is created |
| *listener* | an instance of the ViewListenerBase interface to allow listening to all the events raised for this view |

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

Creates a new UI Must be called after the context has signaled that it is ready via Coherent::UI::ContextListener::-ContextReady

**Parameters**

| | |
|---|---|
| *info* | an instance of the ViewInfo struct that describes all the parameters of the view |
| *path* | the page to load when the view is created |
| *listener* | an instance of the ViewListenerBase interface to allow listening to all the events raised for this view |

#### 14.47.2.5 virtual void Coherent.UI.ViewContext.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 context 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

#### 14.47.2.6 virtual void Coherent.UI.ViewContext.RemoveCrossOriginWhitelistEntry ( string *sourceOrigin,* string *targetProtocol,* string *targetDomain,* bool *allowTargetSubdomains* ) [inline], [virtual]

Remove an entry from the cross-origin access whitelist. Must be called after the context has signaled that it is ready via Coherent::UI::ContextListener::ContextReady

#### 14.47.2.7 virtual bool Coherent.UI.ViewContext.SaveCookies ( ) [inline], [virtual]

Saves the cookies to disk. Must be called after the context has signaled that it is ready via Coherent::UI::ContextListener::ContextReady

**Returns**

true if cookies are enabled and a request for saving has been sent, false otherwise.

#### 14.47.2.8 virtual bool Coherent.UI.ViewContext.Uninitialize ( ) [inline], [virtual]

Frees all resources used by the event system. the ViewContext object is unusable after this call

**Returns**

true if the context has been properly uninitialized

#### 14.47.2.9 virtual void Coherent.UI.ViewContext.Update ( ) [inline], [virtual]

Communicates with the UI context process and performs all updates. Must be called in the thread that owns the context. All registered callbacks will be called in here

## 14.48 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.*

### 14.48.1 Detailed Description

Encapsulates a view-related error.

### 14.48.2 Property Documentation

#### 14.48.2.1 string Coherent.UI.ViewError.Error [get], [set]

Error description.

#### 14.48.2.2 ViewErrorType Coherent.UI.ViewError.ErrorCode [get], [set]

Indicates an error code.

## 14.49 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. Transparent Views are not supported in the Coherent In-Game Browser product*
- 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 Context 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.*

### 14.49.1 Detailed Description

Encapsulates the options of a

### 14.49.2 Property Documentation

#### 14.49.2.1 float Coherent.UI.ViewInfo.ClickThroughAlphaThreshold [get],[set]

The alpha threshold for click-through queries.

#### 14.49.2.2 bool Coherent.UI.ViewInfo.ControlTimeOnDemand [get],[set]

Defines if the client controls the time of on-demand views himself. If not

#### 14.49.2.3 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.

#### 14.49.2.4 int Coherent.UI.ViewInfo.Height [get],[set]

The Height of a

#### 14.49.2.5 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

#### 14.49.2.6 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. Transparent Views are not supported in the Coherent In-Game Browser product

#### 14.49.2.7 bool Coherent.UI.ViewInfo.SupportClickThrough [get],[set]

Defines if the view should support click-through queries. Not supporting click-through increases performance slightly.

#### 14.49.2.8 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 Context Update is called. (available only in Full versions of

#### 14.49.2.9 bool Coherent.UI.ViewInfo.UsesSharedMemory [get],[set]

Flags if a

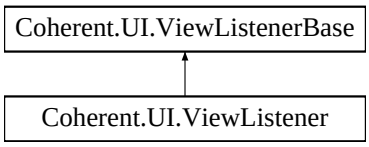
#### 14.49.2.10 int Coherent.UI.ViewInfo.Width [get],[set]

The Width of a

## 14.50 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 context.*
- 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 (URLRequest request)

  *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 context thread in order to support client applications with multi-threaded rendering.*

- override void OnCertificateError (string url, CertificateStatus status, Certificate certificate, CertificateErrorResponse 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 bool OnCanCreateChildWindow (string openerUrl, string targetUrl, ChildViewInfo childViewInfo)

  *Called when the view wants to open a new window.*
- override void OnFileSelectRequest (FileSelectRequest request)

  *Called when the view requests file selection. It could be either single file, directory or multiple files.*
- override void OnAudioStreamCreated (int streamId, int channels, int bitDepth, int frequency)

  *Called when a new WebAudio stream is created. This is usually when the page is loading (for <audio> HTML tags).*
- override void OnAudioStreamPlay (int streamId)

  *Called when a WebAudio stream is played.*
- override void OnAudioStreamPause (int streamId)

  *Called when a WebAudio stream is stopped.*
- override void OnAudioStreamClose (int streamId)

  *Called when a WebAudio stream is closed. This is usually called when the page is unloading.*

## 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 methof 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.*

- CoherentUI_OnAudioStreamCreated AudioStreamCreated

  *Fired when a new WebAudio stream is created. This is usually when the page is loading (for <audio> HTML tags).*

- CoherentUI_OnAudioStreamPlay AudioStreamPlay

  *Fired when a WebAudio stream is played.*

- CoherentUI_OnAudioStreamPause AudioStreamPause

  *Fired when a WebAudio stream is stopped.*

- CoherentUI_OnAudioStreamClose AudioStreamClose

  *Fired when a WebAudio stream is closed. This is usually called when the page is unloading.*

## Additional Inherited Members

### 14.50.1 Detailed Description

Interface that allows clients to listen to

### 14.50.2 Member Function Documentation

#### 14.50.2.1 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**

| | |
|---|---|
| *sharedMemory* | true if the surface should be created in shared memory (4 ∗ width ∗ height bytes); false if a shared texture must be created. |
| *width* | the width of the surface in pixels |
| *height* | the height of the surface in pixels |
| *response* | • object to hold the response when the surface is created or fails it's creation - must be signaled |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.2 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 context thread in order to support client applications with multi-threaded rendering.

**Parameters**

| | |
|---|---|
| *surface* | handle to the surface |
| *usesShared-Memory* | determines whether the surface parameter is a handle to shared memory or shared texture |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.3 override void Coherent.UI.ViewListener.OnAudioStreamClose ( int *streamId* ) [inline], [virtual]

Called when a WebAudio stream is closed. This is usually called when the page is unloading.

**Parameters**

| | |
|---|---|
| *streamId* | the ID of the stream that is closed |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.4 override void Coherent.UI.ViewListener.OnAudioStreamCreated ( int *streamId,* int *channels,* int *bitDepth,* int *frequency* ) [inline], [virtual]

Called when a new WebAudio stream is created. This is usually when the page is loading (for <audio> HTML tags).

**Parameters**

| | |
|---|---|
| *streamId* | the generated ID for the new stream |
| *channels* | the number of channels in the stream |
| *bitDepth* | bits per sample in the stream |
| *frequency* | the audio stream frequency, in Hz |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.5 override void Coherent.UI.ViewListener.OnAudioStreamPause ( int *streamId* ) [inline], [virtual]

Called when a WebAudio stream is stopped.

**Parameters**

| | |
|---|---|
| *streamId* | the ID of the stream that is stopped |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.6 override void Coherent.UI.ViewListener.OnAudioStreamPlay ( int *streamId* ) [inline],[virtual]

Called when a WebAudio stream is played.

**Parameters**

| | |
|---|---|
| *streamId* | the ID of the stream that is played |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.7 override void Coherent.UI.ViewListener.OnBindingsReleased ( int *frameId,* string *path,* bool *isMainFrame* ) [inline],[virtual]

Called when the bindings for frame are released.

**Parameters**

| | |
|---|---|
| *frameId* | the id of the frame |
| *path* | the path in the frame |
| *isMainFrame* | true if this is the main frame of the view |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.8 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**

| | |
|---|---|
| *eventName* | name of the event |
| *arguments* | arguments of the event invocation |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.9 override bool Coherent.UI.ViewListener.OnCanCreateChildWindow ( string *openerUrl,* string *targetUrl,* ChildViewInfo *childViewInfo* ) [inline],[virtual]

Called when the view wants to open a new window.

**Parameters**

| | |
|---|---|
| *openerUrl* | the URL that wants to open the new window |
| *targetUrl* | the target URL of the new window |
| *childViewInfo* | structure that the user should fill in if she's about to allow new view creation |

**Returns**

true if you allow the creation of a new window, false otherwise (if false, the childViewInfo parameter is ignored)

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.10 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**

| | |
|---|---|
| *the* | x position of the selection caret |
| *the* | y position of the selection caret |
| *the* | width of the selection caret |
| *the* | height of the selection caret |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.11 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**

| | |
|---|---|
| *url* | the url of the request |
| *status* | the error status of the certificate |
| *certificate* | the certificate details. This pointer will be valid only for this call |
| *response* | object to signal whether to continue loading the URL. This pointer will be valid only for this call |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.12 override void Coherent.UI.ViewListener.OnCursorChanged ( **CursorTypes** *cursor* ) [inline], [virtual]

Called when the cursor has changed internally in the

**Parameters**

| | |
|---|---|
| *cursor* | the new cursor |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.13 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**

| | |
|---|---|
| *handle* | 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 View was created. The handle is valid only during this call |
| *usesShared-Memory* | determines whether the handle parameter is a handle to shared memory or shared texture |
| *width* | the width of the surface |
| *height* | the height of the surface |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.14 override void Coherent.UI.ViewListener.OnError ( **ViewError** *error* ) [inline],[virtual]

Called when an error occurs for this specific

**Parameters**

| | |
|---|---|
| *error* | error description |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.15 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**

| | |
|---|---|
| *frameId* | the id of the frame |
| *validatedPath* | the path in the frame |
| *isMainFrame* | true if this is the main frame of the View |
| *error* | error message for the failure |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.16 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**

| | |
|---|---|
| *request* | contains the file selection params for the request |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.17 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**

| | |
|---|---|
| *frameId* | the id of the loaded frame |
| *validatedPath* | the path loaded in the frame |
| *isMainFrame* | true if this is the main frame of the View |
| *statusCode* | the status code of the response |
| *headers* | an array of header fields |
| *headersCount* | the count of items in the headers array |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.18 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**

| | |
|---|---|
| *isProxy* | whether the request came from a server or a proxy |
| *host* | the host which triggered the request |

| | |
|---|---|
| *port* | the port at which the request was triggered |
| *realm* | realm of the authentication challenge. Encoded in UTF-8 |
| *scheme* | the authentication scheme used, e.g. "basic" or "digest". Encoded in ASCII |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.19 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.

#### 14.50.2.20 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**

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

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.21 override void Coherent.UI.ViewListener.OnNavigateTo ( string *path* ) [inline], [virtual]

Called when the view starts navigation to a new path.

**Parameters**

| | |
|---|---|
| *path* | URL that the view is navigating to |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.22 override void Coherent.UI.ViewListener.OnReadyForBindings ( int *frameId,* string *path,* bool *isMainFrame* ) [inline], [virtual]

Called when a frame is ready for bindings.

**Parameters**

| | |
|---|---|
| *frameId* | the id of the frame |
| *path* | the path in the frame |
| *isMainFrame* | true if this is the main frame of the view |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.23 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**

| | |
|---|---|
| *request* | contains the available media streams for the request |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.24 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**

| | |
|---|---|
| *level* | message level |
| *message* | the text of the message |
| *sourceId* | id of the script (usually file name) |
| *line* | the number of the line in which the message was sent |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.25 override void Coherent.UI.ViewListener.OnStartLoading ( ) [inline], [virtual]

Called when a new path has started loading.

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.26 override void Coherent.UI.ViewListener.OnStopLoading ( ) [inline], [virtual]

Called when all load operations have completed.

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.27 override void Coherent.UI.ViewListener.OnTextInputTypeChanged ( TextInputControlType *type*, bool *canComposeInline* ) [inline], [virtual]

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

**Parameters**

| | |
|---|---|
| *type* | the type of the currently focused text input control by the user |
| *canCompose-Inline* | if the IME composition could be performed in-line in the control |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.28 override void Coherent.UI.ViewListener.OnURLRequest ( URLRequest *request* ) [inline], [virtual]

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

**Parameters**

| | |
|---|---|
| *request* | the URLRequest object that can be used to modify the request. This pointer will be valid only for this call |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.29 override void Coherent.UI.ViewListener.OnViewCreated ( View *view* ) [inline],[virtual]

Called when the requested

**Parameters**

| | |
|---|---|
| *view* | the instance of the view containing all manipulation methods |

Reimplemented from Coherent.UI.ViewListenerBase.

#### 14.50.2.30 override void Coherent.UI.ViewListener.Release ( ) [inline], [virtual]

Called when the listener is no longer needed by the UI context.

Reimplemented from Coherent.UI.ViewListenerBase.

### 14.50.3 Event Documentation

#### 14.50.3.1 CoherentUI_OnAudioStreamClose Coherent.UI.ViewListener.AudioStreamClose

Fired when a WebAudio stream is closed. This is usually called when the page is unloading.

#### 14.50.3.2 CoherentUI_OnAudioStreamCreated Coherent.UI.ViewListener.AudioStreamCreated

Fired when a new WebAudio stream is created. This is usually when the page is loading (for <audio> HTML tags).

#### 14.50.3.3 CoherentUI_OnAudioStreamPause Coherent.UI.ViewListener.AudioStreamPause

Fired when a WebAudio stream is stopped.

#### 14.50.3.4 CoherentUI_OnAudioStreamPlay Coherent.UI.ViewListener.AudioStreamPlay

Fired when a WebAudio stream is played.

#### 14.50.3.5 CoherentUI_OnBindingsReleased Coherent.UI.ViewListener.BindingsReleased

Fired when the bindings for frame are released.

#### 14.50.3.6 CoherentUI_OnCallback Coherent.UI.ViewListener.Callback

Fired by the UI when there is no registered handler for this event.

#### 14.50.3.7 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.

#### 14.50.3.8 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

#### 14.50.3.9 CoherentUI_OnCursorChanged Coherent.UI.ViewListener.CursorChanged

Fired when the cursor has changed internally in the

#### 14.50.3.10 CoherentUI_OnDraw Coherent.UI.ViewListener.Draw

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

#### 14.50.3.11 CoherentUI_OnError Coherent.UI.ViewListener.Error

Fired when an error occurs for this specific

#### 14.50.3.12 CoherentUI_OnFailLoad Coherent.UI.ViewListener.FailLoad

Fired when a frame has been failed loading.

#### 14.50.3.13 CoherentUI_OnFileSelectRequest Coherent.UI.ViewListener.FileSelectRequest

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

#### 14.50.3.14 CoherentUI_OnFinishLoad Coherent.UI.ViewListener.FinishLoad

Fired when a frame has been successfully loaded.

#### 14.50.3.15 CoherentUI_OnGetAuthCredentials Coherent.UI.ViewListener.GetAuthCredentials

Fired when a view requires authentication credentials.

#### 14.50.3.16 CoherentUI_OnIMEShouldCancelComposition Coherent.UI.ViewListener.IMEShouldCancelComposition

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

#### 14.50.3.17 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.

#### 14.50.3.18 CoherentUI_OnNavigateTo Coherent.UI.ViewListener.NavigateTo

Fired when the view starts navigation to a new path.

#### 14.50.3.19 CoherentUI_OnReadyForBindings Coherent.UI.ViewListener.ReadyForBindings

Fired when a frame is ready for bindings.

#### 14.50.3.20 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.

#### 14.50.3.21 CoherentUI_OnScriptMessage Coherent.UI.ViewListener.ScriptMessage

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

#### 14.50.3.22 CoherentUI_OnStartLoading Coherent.UI.ViewListener.StartLoading

Fired when a new path has started loading.

#### 14.50.3.23 CoherentUI_OnStopLoading Coherent.UI.ViewListener.StopLoading

Fired when all load operations have completed.

#### 14.50.3.24 CoherentUI_OnTextInputTypeChanged Coherent.UI.ViewListener.TextInputTypeChanged

Fired 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

#### 14.50.3.25 CoherentUI_OnURLRequest Coherent.UI.ViewListener.URLRequest

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

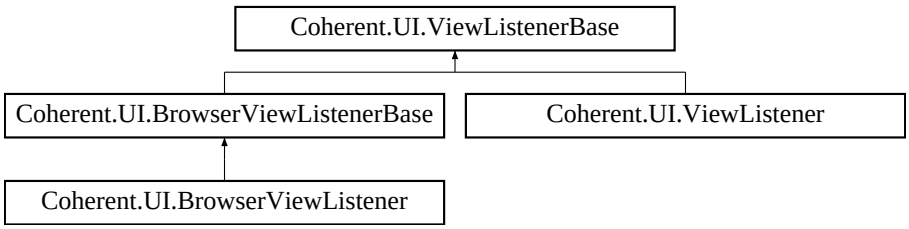
#### 14.50.3.26 CoherentUI_OnViewCreated Coherent.UI.ViewListener.ViewCreated

Fired when the requested

## 14.51 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.*

### 14.51.1 Detailed Description

Interface all view listeners inherit. For an easier to use interface inherit instead -. Coherent::UI::ViewListener

### 14.51.2 Member Enumeration Documentation

#### 14.51.2.1 enum Coherent.UI.ViewListenerBase.MessageLevel

Levels of script messages.

# Index


Accept
    Coherent::UI::HTTPHeader, 111
AcceptCharset
    Coherent::UI::HTTPHeader, 111
AcceptEncoding
    Coherent::UI::HTTPHeader, 111
AcceptLanguage
    Coherent::UI::HTTPHeader, 111
AcceptRanges
    Coherent::UI::HTTPHeader, 111
AcceptedMIMETypes
    Coherent::UI::FileSelectionParams, 108
AcceptedMIMETypesCount
    Coherent::UI::FileSelectionParams, 108
Add
    Coherent::UI::FileSelectRequest, 109
AddCrossOriginWhitelistEntry
    Coherent::UI::ViewContext, 150
AdditionalHostOptions
    Coherent::UI::ContextSettingsBase, 102
All
    Coherent::UI::Binding, 63
AllowCookies
    Coherent::UI::ContextSettingsBase, 102
AppGroupName
    Coherent::UI::FactorySettings, 106
AppendExtraHeaderValue
    Coherent::UI::URLRequest, 123
Arguments
    Coherent::UI::Binding::CallbackArguments, 85
Array
    Coherent::UI::Binding, 63
Assert
    Coherent::UI::ILogHandler, 112
AudioStreamClose
    Coherent::UI::BrowserViewListener, 78
    Coherent::UI::ViewListener, 165
AudioStreamCreated
    Coherent::UI::BrowserViewListener, 78
    Coherent::UI::ViewListener, 165
AudioStreamPause
    Coherent::UI::BrowserViewListener, 79
    Coherent::UI::ViewListener, 165
AudioStreamPlay
    Coherent::UI::BrowserViewListener, 79
    Coherent::UI::ViewListener, 165
AuthCredentialsResponse
    Coherent::UI::View, 139
Authorization
    Coherent::UI::HTTPHeader, 111

BindCall
    Coherent::UI::View, 140
BindMethods
    Coherent::UI::Binding::BoundObject, 67
BindingsReleased
    Coherent::UI::BrowserViewListener, 79
    Coherent::UI::ViewListener, 165
Boolean
    Coherent::UI::Binding, 63
Button
    Coherent::UI::MouseEventData, 117
Byte
    Coherent::UI::Binding, 63

CE_GPUProcessCrash
    Coherent::UI, 60
CE_GPUProcessDeviceMismatch
    Coherent::UI, 60
CE_InternalError
    Coherent::UI, 60
CE_InvalidKey
    Coherent::UI, 60
CE_MissingComponent
    Coherent::UI, 60
CE_Success
    Coherent::UI, 60
CE_VersionMismatch
    Coherent::UI, 60
CacheControl
    Coherent::UI::HTTPHeader, 111
CachePath
    Coherent::UI::ContextSettings, 100
call
    engine, 64
Callback
    Coherent::UI::BrowserViewListener, 79
    Coherent::UI::ViewListener, 165
Cancel
    Coherent::UI::Download, 103
CaretRectChanged
    Coherent::UI::BrowserViewListener, 79
    Coherent::UI::ViewListener, 165
CertificateError
    Coherent::UI::BrowserViewListener, 79
    Coherent::UI::ViewListener, 165
CertificateStatus
    Coherent::UI, 60
Char

Coherent::UI::Binding, 63
ClearCrossOriginWhitelist
Coherent::UI::ViewContext, 150
Click
Coherent::UI::Button, 83
ClickThroughAlphaThreshold
Coherent::UI::ViewInfo, 152
CoherentUIView, 94
ClickToFocus
CoherentUIView, 94
Coherent, 57
Coherent.UI, 57
Coherent.UI.Binding, 62
Coherent.UI.Binding.BoundObject, 67
Coherent.UI.Binding.CallbackArguments, 85
Coherent.UI.Binding.CoherentProperty, 87
Coherent.UI.Binding.CoherentType, 88
Coherent.UI.Binding.InvalidValueCastException, 113
Coherent.UI.Binding.UnsupportedPrimitiveTypeException, 122
Coherent.UI.Binding.Value, 127
Coherent.UI.Binding.ValueObject, 136
Coherent.UI.BrowserView, 67
Coherent.UI.BrowserViewListener, 68
Coherent.UI.BrowserViewListenerBase, 81
Coherent.UI.Button, 81
Coherent.UI.ButtonHandler, 83
Coherent.UI.CertificatePrincipal, 86
Coherent.UI.ChildViewInfo, 87
Coherent.UI.ContextError, 98
Coherent.UI.ContextListener, 99
Coherent.UI.ContextSettings, 100
Coherent.UI.ContextSettingsBase, 101
Coherent.UI.Download, 103
Coherent.UI.EventModifiersState, 105
Coherent.UI.EventMouseModifiersState, 105
Coherent.UI.FactorySettings, 105
Coherent.UI.FactorySettingsBase, 106
Coherent.UI.FileHandler, 107
Coherent.UI.FileSelectRequest, 109
Coherent.UI.FileSelectionParams, 108
Coherent.UI.HTTPHeader, 110
Coherent.UI.ILogHandler, 112
Coherent.UI.ImageData, 113
Coherent.UI.KeyEventData, 114
Coherent.UI.MediaStreamDevice, 115
Coherent.UI.MediaStreamRequest, 115
Coherent.UI.MouseEventData, 117
Coherent.UI.ResourceData, 118
Coherent.UI.ResourceResponse, 119
Coherent.UI.SubscriptionError, 120
Coherent.UI.SurfaceResponse, 121
Coherent.UI.TouchEventData, 121
Coherent.UI.URLRequest, 123
Coherent.UI.URLRequestBase, 125
Coherent.UI.View, 136
Coherent.UI.ViewContext, 149
Coherent.UI.ViewError, 151
Coherent.UI.ViewInfo, 152
Coherent.UI.ViewListener, 153
Coherent.UI.ViewListenerBase, 167
Coherent::UI
CE_GPUProcessCrash, 60
CE_GPUProcessDeviceMismatch, 60
CE_InternalError, 60
CE_InvalidKey, 60
CE_MissingComponent, 60
CE_Success, 60
CE_VersionMismatch, 60
DET_DownloadOperationFailed, 60
DET_Success, 60
SCE_ArgumentType, 61
SCE_NoResult, 61
SCE_NoSuchMethod, 61
SCE_Success, 61
SET_Error, 61
SET_OfflineGracePeriod, 61
SET_Success, 61
SET_SuccessNotify, 61
VE_AbnormalTermination, 62
VE_ChildCompositionFailed, 62
VE_ClickThroughFailed, 62
VE_FailedCreation, 61
VE_FailedRenderingCreation, 62
VE_FrameAlreadyInFlight, 62
VE_FrameNotReady, 62
VE_FrameNotRequeted, 62
VE_InvalidCall, 62
VE_InvalidPath, 62
VE_QueryNotReady, 62
VE_Success, 61
Coherent::UI::Binding
All, 63
Array, 63
Boolean, 63
Byte, 63
Char, 63
Explicit, 63
Instance, 63
Integer, 63
Null, 63
Number, 63
Object, 63
SByte, 63
Static, 63
String, 63
UInteger, 63
Coherent::UI::HTTPHeader
Accept, 111
AcceptCharset, 111
AcceptEncoding, 111
AcceptLanguage, 111
AcceptRanges, 111
Authorization, 111
CacheControl, 111
Connection, 111

ContentLength, 111
ContentRange, 111
ContentType, 111
Cookie, 111
Host, 111
IfModifiedSince, 111
IfNoneMatch, 111
IfRange, 111
Origin, 111
Pragma, 111
ProxyAuthorization, 111
ProxyConnection, 111
Range, 111
Referer, 111
TransferEncoding, 111
UserAgent, 111
Coherent::UI
CertificateStatus, 60
ContextErrorType, 60
CursorTypes, 60
Direct3DInterfaceType, 60
DownloadErrorType, 60
FileSelectionMode, 60
JavaScriptMessageType, 61
MediaStreamType, 61
RenderingDeviceType, 61
ScriptCallErrorType, 61
SubscriptionErrorType, 61
TextInputControlType, 61
UserAgentType, 61
ViewErrorType, 61
ViewType, 62
Coherent::UI::Binding
PropertyBindingFlags, 63
ValueType, 63
Coherent::UI::Binding::BoundObject
BindMethods, 67
Coherent::UI::Binding::CallbackArguments
Arguments, 85
Coherent::UI::Binding::CoherentProperty
CoherentProperty, 88
Coherent::UI::Binding::Value
Equals, 131
GetHashCode, 131
operator bool, 131
operator byte, 131
operator char, 132
operator Dictionary< string, Value >, 132
operator double, 132
operator float, 132
operator int, 132
operator sbyte, 134
operator string, 134
operator uint, 134
operator==, 136
Type, 136
Value, 128, 129, 131
Coherent::UI::BrowserView
GetHistory, 68
GetViewType, 68
GoBack, 68
GoForward, 68
Coherent::UI::BrowserViewListener
AudioStreamClose, 78
AudioStreamCreated, 78
AudioStreamPause, 79
AudioStreamPlay, 79
BindingsReleased, 79
Callback, 79
CaretRectChanged, 79
CertificateError, 79
CreateSurface, 71
CursorChanged, 79
DestroySurface, 71
Draw, 79
Error, 79
FailLoad, 79
FileSelectRequest, 79
FinishLoad, 79
GetAuthCredentials, 80
IMEShouldCancelComposition, 80
JavaScriptMessage, 80
NavigateTo, 80
OnAudioStreamClose, 73
OnAudioStreamCreated, 73
OnAudioStreamPause, 73
OnAudioStreamPlay, 73
OnBindingsReleased, 73
OnCallback, 74
OnCanCreateChildWindow, 74
OnCaretRectChanged, 74
OnCertificateError, 74
OnCursorChanged, 75
OnDraw, 75
OnError, 75
OnFailLoad, 75
OnFileSelectRequest, 76
OnFinishLoad, 76
OnGetAuthCredentials, 76
OnIMEShouldCancelComposition, 76
OnJavaScriptMessage, 76
OnNavigateTo, 77
OnReadyForBindings, 77
OnRequestMediaStream, 77
OnScriptMessage, 77
OnStartLoading, 77
OnStopLoading, 78
OnTextInputTypeChanged, 78
OnURLRequest, 78
OnViewCreated, 78
ReadyForBindings, 80
Release, 78
RequestMediaStream, 80
ScriptMessage, 80
StartLoading, 80
StopLoading, 80

TextInputTypeChanged, 80
URLRequest, 80
ViewCreated, 80
Coherent::UI::Button
Click, 83
Disabled, 83
Initialize, 82
Label, 83
Coherent::UI::ButtonHandler
Handler, 85
Initialize, 84
Invoke, 85
Invoker, 85
IsMessage, 85
Method, 85
Target, 85
Coherent::UI::CertificatePrincipal
CommonName, 86
CountryName, 86
DomainComponents, 86
LocalityName, 86
OrganizationNames, 86
OrganizationUnitNames, 86
StateOrProvinceName, 87
StreetAddresses, 87
Coherent::UI::ChildViewInfo
Listener, 87
NewViewInfo, 87
Coherent::UI::ContextError
Error, 98
ErrorCode, 98
Coherent::UI::ContextListener
ContextReady, 99
OnCookiesSaved, 99
OnDownloadComplete, 99
OnDownloadFailed, 99
OnDownloadStarted, 99
OnError, 100
OnSubscriptionCheckDone, 100
Coherent::UI::ContextSettings
CachePath, 100
CookiesResource, 100
CustomUserAgentString, 101
HTML5LocalStoragePath, 101
Coherent::UI::ContextSettingsBase
AdditionalHostOptions, 102
AllowCookies, 102
DebuggerPort, 102
DisableWebSecurity, 102
EnableClientAudioPlayback, 102
EnableProxy, 102
ForceDisablePluginFullscreen, 102
SubscriptionEmail, 102
SubscriptionKey, 102
UserAgent, 102
Coherent::UI::Download
Cancel, 103
Destroy, 103
GetDownloadedBytesSoFar, 103
GetFilename, 104
GetId, 104
GetReferrer, 104
GetTotalBytes, 104
GetURL, 104
HasFailed, 104
IsCancelled, 105
IsComplete, 105
Coherent::UI::FactorySettings
AppGroupName, 106
HostDirectory, 106
MinidumpFolder, 106
Coherent::UI::FactorySettingsBase
EnableSupportForProprietaryCodecs, 107
WriteMinidumps, 107
Coherent::UI::FileHandler
ReadFile, 107
WriteFile, 108
Coherent::UI::FileSelectRequest
Add, 109
Finish, 109
GetFileSelectionParams, 109
Coherent::UI::FileSelectionParams
AcceptedMIMETypes, 108
AcceptedMIMETypesCount, 108
DefaultFileName, 108
DialogTitle, 109
Mode, 109
Coherent::UI::HTTPHeader
Content, 112
Field, 112
FieldType, 111
GetHeaderField, 111
Coherent::UI::ILogHandler
Assert, 112
WriteLog, 112
Coherent::UI::ImageData
Destroy, 113
Update, 113
Coherent::UI::KeyEventData
IsAutoRepeat, 114
IsNumPad, 114
KeyCode, 114
Modifiers, 114
Type, 114
Coherent::UI::MediaStreamDevice
DeviceId, 115
Name, 115
Type, 115
Coherent::UI::MediaStreamRequest
Devices, 116
Respond, 116
Coherent::UI::MouseEventData
Button, 117
Modifiers, 117
MouseModifiers, 117
Type, 117

WheelX, 117
WheelY, 118
X, 118
Y, 118
Coherent::UI::ResourceData
GetBuffer, 118
GetSize, 118
SignalFailure, 119
SignalSuccess, 119
Coherent::UI::ResourceResponse
GetBuffer, 119
SetResponseHeader, 120
SetStatus, 120
SignalFailure, 120
SignalSuccess, 120
Coherent::UI::SubscriptionError
Error, 121
ErrorCode, 121
Coherent::UI::SurfaceResponse
Signal, 121
Coherent::UI::TouchEventData
HasRadiusInformation, 122
Id, 122
RadiusX, 122
RadiusY, 122
Type, 122
X, 122
Y, 122
Coherent::UI::URLRequest
AppendExtraHeaderValue, 123
Deny, 124
RemoveExtraHeader, 124
SetExtraHeader, 124
SetExtraHeaders, 124
SetMethod, 124
SetReferrer, 124
SetURL, 124
Coherent::UI::URLRequestBase
GetExtraHeader, 125
GetExtraHeaderIndex, 125
GetExtraHeaders, 126
GetMethod, 126
GetReferrer, 126
GetURL, 126
HasExtraHeader, 126
Coherent::UI::View
AuthCredentialsResponse, 139
BindCall, 140
CreateImageData, 140
Destroy, 140
DownloadUrl, 140
DrawSurface, 140
ExecuteScript, 140
FetchMouseOnUIQuery, 141
FetchSurface, 141
GetAudioData, 141
GetClickThroughAlphaThreshold, 141
GetContext, 141
GetCurentViewPath, 141
GetHeight, 142
GetLastFailedPath, 142
GetLastLoadedPath, 142
GetLastRequestedPath, 142
GetTargetFramerate, 142
GetViewType, 142
GetWidth, 142
HasMouseQueryFinished, 143
IMEActivate, 143
IMECancelComposition, 143
IMEConfirmComposition, 143
IMEIsActive, 143
IMESetComposition, 143
InterceptURLRequests, 144
IsMouseOnView, 144
IsOnDemand, 144
IsTransparent, 145
IssueMouseOnUIQuery, 144
KeyEvent, 145
KillFocus, 145
Load, 145
MouseActivate, 145
MouseEvent, 145
PeekFrameReady, 145
Redraw, 145
RegisterForEvent, 146
Reload, 146
ReplyToJavaScriptMessage, 146
RequestFrame, 146
Resize, 147
SetClickThroughAlphaThreshold, 147
SetFocus, 147
SetMasterVolume, 147
SetScriptError, 147
SetTargetFramerate, 147
SetZoomLevel, 148
SupportsClickThrough, 148
TouchEvent, 148
TriggerEvent, 148
UnbindCall, 148
UnbindObject, 148
UnregisterFromEvent, 148
UsesSharedMemory, 149
Coherent::UI::ViewContext
AddCrossOriginWhitelistEntry, 150
ClearCrossOriginWhitelist, 150
CreateView, 150
FetchSurfaces, 150
RemoveCrossOriginWhitelistEntry, 150
SaveCookies, 150
Uninitialize, 151
Update, 151
Coherent::UI::ViewError
Error, 151
ErrorCode, 151
Coherent::UI::ViewInfo
ClickThroughAlphaThreshold, 152

ControlTimeOnDemand, 152
ForceSoftwareRendering, 152
Height, 153
IsOnDemand, 153
IsTransparent, 153
SupportClickThrough, 153
TargetFrameRate, 153
UsesSharedMemory, 153
Width, 153
Coherent::UI::ViewListener
AudioStreamClose, 165
AudioStreamCreated, 165
AudioStreamPause, 165
AudioStreamPlay, 165
BindingsReleased, 165
Callback, 165
CaretRectChanged, 165
CertificateError, 165
CreateSurface, 156
CursorChanged, 165
DestroySurface, 157
Draw, 165
Error, 166
FailLoad, 166
FileSelectRequest, 166
FinishLoad, 166
GetAuthCredentials, 166
IMEShouldCancelComposition, 166
JavaScriptMessage, 166
NavigateTo, 166
OnAudioStreamClose, 157
OnAudioStreamCreated, 157
OnAudioStreamPause, 157
OnAudioStreamPlay, 158
OnBindingsReleased, 158
OnCallback, 158
OnCanCreateChildWindow, 158
OnCaretRectChanged, 158
OnCertificateError, 160
OnCursorChanged, 160
OnDraw, 160
OnError, 160
OnFailLoad, 161
OnFileSelectRequest, 161
OnFinishLoad, 161
OnGetAuthCredentials, 161
OnIMEShouldCancelComposition, 162
OnJavaScriptMessage, 162
OnNavigateTo, 162
OnReadyForBindings, 162
OnRequestMediaStream, 162
OnScriptMessage, 163
OnStartLoading, 163
OnStopLoading, 163
OnTextInputTypeChanged, 163
OnURLRequest, 163
OnViewCreated, 163
ReadyForBindings, 166
Release, 165
RequestMediaStream, 166
ScriptMessage, 166
StartLoading, 166
StopLoading, 167
TextInputTypeChanged, 167
URLRequest, 167
ViewCreated, 167
Coherent::UI::ViewListenerBase
MessageLevel, 167
CoherentProperty
Coherent::UI::Binding::CoherentProperty, 88
CoherentUILiveGameView, 88
CoherentUISystem, 88
DiskCacheSize, 90
FileHandlerFactoryFunc, 90
HasFocusedView, 90
IsReady, 89
IsUpdating, 90
m_MainCamera, 90
MemoryCacheSize, 90
OnViewFocused, 91
SystemListenerFactoryFunc, 90
UISystem, 91
UseURLCache, 91
CoherentUIView, 91
ClickThroughAlphaThreshold, 94
ClickToFocus, 94
CorrectGamma, 94
DestroyView, 93
DrawAfterPostEffects, 94
EnableBindingAttribute, 95
EnableIME, 95
EnableWebGLSupport, 95
FlipY, 95
ForceSoftwareRendering, 95
GetCamDimensions, 93
Height, 95
IgnoreDisplayDensity, 95
InitialScript, 95
InputState, 96
InterceptAllEvents, 96
IsIndependentOfZBuffer, 96
IsSurfaceView, 96
IsTransparent, 96
Listener, 96
OnReadyForBindings, 96
OnViewCreated, 96
OnViewDestroyed, 98
Page, 97
ReceivesInput, 97
Reload, 93
Resize, 94
ScaleToFit, 97
SetMousePosition, 94
ShowJavaScriptDialogs, 97
SupportClickThrough, 97
UseCameraDimensions, 97

View, 97
ViewDestroyedHandler, 94
Width, 97
XPos, 98
YPos, 98
CommonName
Coherent::UI::CertificatePrincipal, 86
Connection
Coherent::UI::HTTPHeader, 111
Content
Coherent::UI::HTTPHeader, 112
ContentLength
Coherent::UI::HTTPHeader, 111
ContentRange
Coherent::UI::HTTPHeader, 111
ContentType
Coherent::UI::HTTPHeader, 111
ContextErrorType
Coherent::UI, 60
ContextReady
Coherent::UI::ContextListener, 99
ControlTimeOnDemand
Coherent::UI::ViewInfo, 152
Cookie
Coherent::UI::HTTPHeader, 111
CookiesResource
Coherent::UI::ContextSettings, 100
CorrectGamma
CoherentUIView, 94
CountryName
Coherent::UI::CertificatePrincipal, 86
createDeferred
engine, 64
CreateImageData
Coherent::UI::View, 140
CreateSurface
Coherent::UI::BrowserViewListener, 71
Coherent::UI::ViewListener, 156
CreateView
Coherent::UI::ViewContext, 150
CursorChanged
Coherent::UI::BrowserViewListener, 79
Coherent::UI::ViewListener, 165
CursorTypes
Coherent::UI, 60
CustomUserAgentString
Coherent::UI::ContextSettings, 101

DET_DownloadOperationFailed
Coherent::UI, 60
DET_Success
Coherent::UI, 60
DebuggerPort
Coherent::UI::ContextSettingsBase, 102
DefaultFileName
Coherent::UI::FileSelectionParams, 108
Deny
Coherent::UI::URLRequest, 124
Destroy
Coherent::UI::Download, 103
Coherent::UI::ImageData, 113
Coherent::UI::View, 140
DestroySurface
Coherent::UI::BrowserViewListener, 71
Coherent::UI::ViewListener, 157
DestroyView
CoherentUIView, 93
DeviceId
Coherent::UI::MediaStreamDevice, 115
Devices
Coherent::UI::MediaStreamRequest, 116
DialogTitle
Coherent::UI::FileSelectionParams, 109
Direct3DInterfaceType
Coherent::UI, 60
DisableWebSecurity
Coherent::UI::ContextSettingsBase, 102
Disabled
Coherent::UI::Button, 83
DiskCacheSize
CoherentUISystem, 90
DomainComponents
Coherent::UI::CertificatePrincipal, 86
DownloadErrorType
Coherent::UI, 60
DownloadUrl
Coherent::UI::View, 140
Draw
Coherent::UI::BrowserViewListener, 79
Coherent::UI::ViewListener, 165
DrawAfterPostEffects
CoherentUIView, 94
DrawSurface
Coherent::UI::View, 140

EnableBindingAttribute
CoherentUIView, 95
EnableClientAudioPlayback
Coherent::UI::ContextSettingsBase, 102
EnableIME
CoherentUIView, 95
EnableProxy
Coherent::UI::ContextSettingsBase, 102
EnableSupportForProprietaryCodecs
Coherent::UI::FactorySettingsBase, 107
EnableWebGLSupport
CoherentUIView, 95
engine, 63
call, 64
createDeferred, 64
off, 64
on, 65
trigger, 65
Equals
Coherent::UI::Binding::Value, 131
Error
Coherent::UI::BrowserViewListener, 79
Coherent::UI::ContextError, 98

Coherent::UI::SubscriptionError, 121
Coherent::UI::ViewError, 151
Coherent::UI::ViewListener, 166
ErrorCode
Coherent::UI::ContextError, 98
Coherent::UI::SubscriptionError, 121
Coherent::UI::ViewError, 151
ExecuteScript
Coherent::UI::View, 140
Explicit
Coherent::UI::Binding, 63

FailLoad
Coherent::UI::BrowserViewListener, 79
Coherent::UI::ViewListener, 166
FetchMouseOnUIQuery
Coherent::UI::View, 141
FetchSurface
Coherent::UI::View, 141
FetchSurfaces
Coherent::UI::ViewContext, 150
Field
Coherent::UI::HTTPHeader, 112
FieldType
Coherent::UI::HTTPHeader, 111
FileHandlerFactoryFunc
CoherentUISystem, 90
FileSelectRequest
Coherent::UI::BrowserViewListener, 79
Coherent::UI::ViewListener, 166
FileSelectionMode
Coherent::UI, 60
Finish
Coherent::UI::FileSelectRequest, 109
FinishLoad
Coherent::UI::BrowserViewListener, 79
Coherent::UI::ViewListener, 166
FlipY
CoherentUIView, 95
ForceDisablePluginFullscreen
Coherent::UI::ContextSettingsBase, 102
ForceSoftwareRendering
Coherent::UI::ViewInfo, 152
CoherentUIView, 95

GetAudioData
Coherent::UI::View, 141
GetAuthCredentials
Coherent::UI::BrowserViewListener, 80
Coherent::UI::ViewListener, 166
GetBuffer
Coherent::UI::ResourceData, 118
Coherent::UI::ResourceResponse, 119
GetCamDimensions
CoherentUIView, 93
GetClickThroughAlphaThreshold
Coherent::UI::View, 141
GetContext
Coherent::UI::View, 141
GetCurentViewPath
Coherent::UI::View, 141
GetDownloadedBytesSoFar
Coherent::UI::Download, 103
GetExtraHeader
Coherent::UI::URLRequestBase, 125
GetExtraHeaderIndex
Coherent::UI::URLRequestBase, 125
GetExtraHeaders
Coherent::UI::URLRequestBase, 126
GetFileSelectionParams
Coherent::UI::FileSelectRequest, 109
GetFilename
Coherent::UI::Download, 104
GetHashCode
Coherent::UI::Binding::Value, 131
GetHeaderField
Coherent::UI::HTTPHeader, 111
GetHeight
Coherent::UI::View, 142
GetHistory
Coherent::UI::BrowserView, 68
GetId
Coherent::UI::Download, 104
GetLastFailedPath
Coherent::UI::View, 142
GetLastLoadedPath
Coherent::UI::View, 142
GetLastRequestedPath
Coherent::UI::View, 142
GetMethod
Coherent::UI::URLRequestBase, 126
GetReferrer
Coherent::UI::Download, 104
Coherent::UI::URLRequestBase, 126
GetSize
Coherent::UI::ResourceData, 118
GetTargetFramerate
Coherent::UI::View, 142
GetTotalBytes
Coherent::UI::Download, 104
GetURL
Coherent::UI::Download, 104
Coherent::UI::URLRequestBase, 126
GetViewType
Coherent::UI::BrowserView, 68
Coherent::UI::View, 142
GetWidth
Coherent::UI::View, 142
GoBack
Coherent::UI::BrowserView, 68
GoForward
Coherent::UI::BrowserView, 68

HTML5LocalStoragePath
Coherent::UI::ContextSettings, 101
Handler
Coherent::UI::ButtonHandler, 85
HasExtraHeader

Coherent::UI::URLRequestBase, 126
HasFailed
Coherent::UI::Download, 104
HasFocusedView
CoherentUISystem, 90
HasMouseQueryFinished
Coherent::UI::View, 143
HasRadiusInformation
Coherent::UI::TouchEventData, 122
Height
Coherent::UI::ViewInfo, 153
CoherentUIView, 95
Host
Coherent::UI::HTTPHeader, 111
HostDirectory
Coherent::UI::FactorySettings, 106

IMEActivate
Coherent::UI::View, 143
IMECancelComposition
Coherent::UI::View, 143
IMEConfirmComposition
Coherent::UI::View, 143
IMEIsActive
Coherent::UI::View, 143
IMESetComposition
Coherent::UI::View, 143
IMEShouldCancelComposition
Coherent::UI::BrowserViewListener, 80
Coherent::UI::ViewListener, 166
Id
Coherent::UI::TouchEventData, 122
IfModifiedSince
Coherent::UI::HTTPHeader, 111
IfNoneMatch
Coherent::UI::HTTPHeader, 111
IfRange
Coherent::UI::HTTPHeader, 111
IgnoreDisplayDensity
CoherentUIView, 95
InitialScript
CoherentUIView, 95
Initialize
Coherent::UI::Button, 82
Coherent::UI::ButtonHandler, 84
InputState
CoherentUIView, 96
Instance
Coherent::UI::Binding, 63
Integer
Coherent::UI::Binding, 63
InterceptAllEvents
CoherentUIView, 96
InterceptURLRequests
Coherent::UI::View, 144
Invoke
Coherent::UI::ButtonHandler, 85
Invoker
Coherent::UI::ButtonHandler, 85
IsAutoRepeat
Coherent::UI::KeyEventData, 114
IsCancelled
Coherent::UI::Download, 105
IsComplete
Coherent::UI::Download, 105
IsIndependentOfZBuffer
CoherentUIView, 96
IsMessage
Coherent::UI::ButtonHandler, 85
IsMouseOnView
Coherent::UI::View, 144
IsNumPad
Coherent::UI::KeyEventData, 114
IsOnDemand
Coherent::UI::View, 144
Coherent::UI::ViewInfo, 153
IsReady
CoherentUISystem, 89
IsSurfaceView
CoherentUIView, 96
IsTransparent
Coherent::UI::View, 145
Coherent::UI::ViewInfo, 153
CoherentUIView, 96
IsUpdating
CoherentUISystem, 90
IssueMouseOnUIQuery
Coherent::UI::View, 144

JavaScriptMessage
Coherent::UI::BrowserViewListener, 80
Coherent::UI::ViewListener, 166
JavaScriptMessageType
Coherent::UI, 61

KeyCode
Coherent::UI::KeyEventData, 114
KeyEvent
Coherent::UI::View, 145
KillFocus
Coherent::UI::View, 145

Label
Coherent::UI::Button, 83
Listener
Coherent::UI::ChildViewInfo, 87
CoherentUIView, 96
Load
Coherent::UI::View, 145
LocalityName
Coherent::UI::CertificatePrincipal, 86

m_MainCamera
CoherentUISystem, 90
MediaStreamType
Coherent::UI, 61
MemoryCacheSize
CoherentUISystem, 90

MessageLevel
Coherent::UI::ViewListenerBase, 167
Method
Coherent::UI::ButtonHandler, 85
MinidumpFolder
Coherent::UI::FactorySettings, 106
MobileSurfaceView, 116
UpdateView, 116
Mode
Coherent::UI::FileSelectionParams, 109
Modifiers
Coherent::UI::KeyEventData, 114
Coherent::UI::MouseEventData, 117
MouseActivate
Coherent::UI::View, 145
MouseEvent
Coherent::UI::View, 145
MouseModifiers
Coherent::UI::MouseEventData, 117

Name
Coherent::UI::MediaStreamDevice, 115
NavigateTo
Coherent::UI::BrowserViewListener, 80
Coherent::UI::ViewListener, 166
NewViewInfo
Coherent::UI::ChildViewInfo, 87
Null
Coherent::UI::Binding, 63
Number
Coherent::UI::Binding, 63

Object
Coherent::UI::Binding, 63
off
engine, 64
on
engine, 65
OnAudioStreamClose
Coherent::UI::BrowserViewListener, 73
Coherent::UI::ViewListener, 157
OnAudioStreamCreated
Coherent::UI::BrowserViewListener, 73
Coherent::UI::ViewListener, 157
OnAudioStreamPause
Coherent::UI::BrowserViewListener, 73
Coherent::UI::ViewListener, 157
OnAudioStreamPlay
Coherent::UI::BrowserViewListener, 73
Coherent::UI::ViewListener, 158
OnBindingsReleased
Coherent::UI::BrowserViewListener, 73
Coherent::UI::ViewListener, 158
OnCallback
Coherent::UI::BrowserViewListener, 74
Coherent::UI::ViewListener, 158
OnCanCreateChildWindow
Coherent::UI::BrowserViewListener, 74
Coherent::UI::ViewListener, 158
OnCaretRectChanged
Coherent::UI::BrowserViewListener, 74
Coherent::UI::ViewListener, 158
OnCertificateError
Coherent::UI::BrowserViewListener, 74
Coherent::UI::ViewListener, 160
OnCookiesSaved
Coherent::UI::ContextListener, 99
OnCursorChanged
Coherent::UI::BrowserViewListener, 75
Coherent::UI::ViewListener, 160
OnDownloadComplete
Coherent::UI::ContextListener, 99
OnDownloadFailed
Coherent::UI::ContextListener, 99
OnDownloadStarted
Coherent::UI::ContextListener, 99
OnDraw
Coherent::UI::BrowserViewListener, 75
Coherent::UI::ViewListener, 160
OnError
Coherent::UI::BrowserViewListener, 75
Coherent::UI::ContextListener, 100
Coherent::UI::ViewListener, 160
OnFailLoad
Coherent::UI::BrowserViewListener, 75
Coherent::UI::ViewListener, 161
OnFileSelectRequest
Coherent::UI::BrowserViewListener, 76
Coherent::UI::ViewListener, 161
OnFinishLoad
Coherent::UI::BrowserViewListener, 76
Coherent::UI::ViewListener, 161
OnGetAuthCredentials
Coherent::UI::BrowserViewListener, 76
Coherent::UI::ViewListener, 161
OnIMEShouldCancelComposition
Coherent::UI::BrowserViewListener, 76
Coherent::UI::ViewListener, 162
OnJavaScriptMessage
Coherent::UI::BrowserViewListener, 76
Coherent::UI::ViewListener, 162
OnNavigateTo
Coherent::UI::BrowserViewListener, 77
Coherent::UI::ViewListener, 162
OnReadyForBindings
Coherent::UI::BrowserViewListener, 77
Coherent::UI::ViewListener, 162
CoherentUIView, 96
OnRequestMediaStream
Coherent::UI::BrowserViewListener, 77
Coherent::UI::ViewListener, 162
OnScriptMessage
Coherent::UI::BrowserViewListener, 77
Coherent::UI::ViewListener, 163
OnStartLoading
Coherent::UI::BrowserViewListener, 77
Coherent::UI::ViewListener, 163

OnStopLoading
Coherent::UI::BrowserViewListener, 78
Coherent::UI::ViewListener, 163
OnSubscriptionCheckDone
Coherent::UI::ContextListener, 100
OnTextInputTypeChanged
Coherent::UI::BrowserViewListener, 78
Coherent::UI::ViewListener, 163
OnURLRequest
Coherent::UI::BrowserViewListener, 78
Coherent::UI::ViewListener, 163
OnViewCreated
Coherent::UI::BrowserViewListener, 78
Coherent::UI::ViewListener, 163
CoherentUIView, 96
OnViewDestroyed
CoherentUIView, 98
OnViewFocused
CoherentUISystem, 91
operator bool
Coherent::UI::Binding::Value, 131
operator byte
Coherent::UI::Binding::Value, 131
operator char
Coherent::UI::Binding::Value, 132
operator Dictionary< string, Value >
Coherent::UI::Binding::Value, 132
operator double
Coherent::UI::Binding::Value, 132
operator float
Coherent::UI::Binding::Value, 132
operator int
Coherent::UI::Binding::Value, 132
operator sbyte
Coherent::UI::Binding::Value, 134
operator string
Coherent::UI::Binding::Value, 134
operator uint
Coherent::UI::Binding::Value, 134
operator==
Coherent::UI::Binding::Value, 136
OrganizationNames
Coherent::UI::CertificatePrincipal, 86
OrganizationUnitNames
Coherent::UI::CertificatePrincipal, 86
Origin
Coherent::UI::HTTPHeader, 111

Page
CoherentUIView, 97
PeekFrameReady
Coherent::UI::View, 145
Pragma
Coherent::UI::HTTPHeader, 111
PropertyBindingFlags
Coherent::UI::Binding, 63
ProxyAuthorization
Coherent::UI::HTTPHeader, 111
ProxyConnection
Coherent::UI::HTTPHeader, 111

RadiusX
Coherent::UI::TouchEventData, 122
RadiusY
Coherent::UI::TouchEventData, 122
Range
Coherent::UI::HTTPHeader, 111
ReadFile
Coherent::UI::FileHandler, 107
ReadyForBindings
Coherent::UI::BrowserViewListener, 80
Coherent::UI::ViewListener, 166
ReceivesInput
CoherentUIView, 97
Redraw
Coherent::UI::View, 145
Referer
Coherent::UI::HTTPHeader, 111
RegisterForEvent
Coherent::UI::View, 146
Release
Coherent::UI::BrowserViewListener, 78
Coherent::UI::ViewListener, 165
Reload
Coherent::UI::View, 146
CoherentUIView, 93
RemoveCrossOriginWhitelistEntry
Coherent::UI::ViewContext, 150
RemoveExtraHeader
Coherent::UI::URLRequest, 124
RenderingDeviceType
Coherent::UI, 61
ReplyToJavaScriptMessage
Coherent::UI::View, 146
RequestFrame
Coherent::UI::View, 146
RequestMediaStream
Coherent::UI::BrowserViewListener, 80
Coherent::UI::ViewListener, 166
Resize
Coherent::UI::View, 147
CoherentUIView, 94
Respond
Coherent::UI::MediaStreamRequest, 116

SByte
Coherent::UI::Binding, 63
SCE_ArgumentType
Coherent::UI, 61
SCE_NoResult
Coherent::UI, 61
SCE_NoSuchMethod
Coherent::UI, 61
SCE_Success
Coherent::UI, 61
SET_Error
Coherent::UI, 61
SET_OfflineGracePeriod

Coherent::UI, 61
SET_Success
Coherent::UI, 61
SET_SuccessNotify
Coherent::UI, 61
SaveCookies
Coherent::UI::ViewContext, 150
ScaleToFit
CoherentUIView, 97
ScriptCallErrorType
Coherent::UI, 61
ScriptMessage
Coherent::UI::BrowserViewListener, 80
Coherent::UI::ViewListener, 166
SetClickThroughAlphaThreshold
Coherent::UI::View, 147
SetExtraHeader
Coherent::UI::URLRequest, 124
SetExtraHeaders
Coherent::UI::URLRequest, 124
SetFocus
Coherent::UI::View, 147
SetMasterVolume
Coherent::UI::View, 147
SetMethod
Coherent::UI::URLRequest, 124
SetMousePosition
CoherentUIView, 94
SetReferrer
Coherent::UI::URLRequest, 124
SetResponseHeader
Coherent::UI::ResourceResponse, 120
SetScriptError
Coherent::UI::View, 147
SetStatus
Coherent::UI::ResourceResponse, 120
SetTargetFramerate
Coherent::UI::View, 147
SetURL
Coherent::UI::URLRequest, 124
SetZoomLevel
Coherent::UI::View, 148
ShowJavaScriptDialogs
CoherentUIView, 97
Signal
Coherent::UI::SurfaceResponse, 121
SignalFailure
Coherent::UI::ResourceData, 119
Coherent::UI::ResourceResponse, 120
SignalSuccess
Coherent::UI::ResourceData, 119
Coherent::UI::ResourceResponse, 120
StartLoading
Coherent::UI::BrowserViewListener, 80
Coherent::UI::ViewListener, 166
StateOrProvinceName
Coherent::UI::CertificatePrincipal, 87
Static
Coherent::UI::Binding, 63
StopLoading
Coherent::UI::BrowserViewListener, 80
Coherent::UI::ViewListener, 167
StreetAddresses
Coherent::UI::CertificatePrincipal, 87
String
Coherent::UI::Binding, 63
SubscriptionEmail
Coherent::UI::ContextSettingsBase, 102
SubscriptionErrorType
Coherent::UI, 61
SubscriptionKey
Coherent::UI::ContextSettingsBase, 102
SupportClickThrough
Coherent::UI::ViewInfo, 153
CoherentUIView, 97
SupportsClickThrough
Coherent::UI::View, 148
SystemListenerFactoryFunc
CoherentUISystem, 90

Target
Coherent::UI::ButtonHandler, 85
TargetFrameRate
Coherent::UI::ViewInfo, 153
TextInputControlType
Coherent::UI, 61
TextInputTypeChanged
Coherent::UI::BrowserViewListener, 80
Coherent::UI::ViewListener, 167
TouchEvent
Coherent::UI::View, 148
TransferEncoding
Coherent::UI::HTTPHeader, 111
trigger
engine, 65
TriggerEvent
Coherent::UI::View, 148
Type
Coherent::UI::Binding::Value, 136
Coherent::UI::KeyEventData, 114
Coherent::UI::MediaStreamDevice, 115
Coherent::UI::MouseEventData, 117
Coherent::UI::TouchEventData, 122

UInteger
Coherent::UI::Binding, 63
UISystem
CoherentUISystem, 91
URLRequest
Coherent::UI::BrowserViewListener, 80
Coherent::UI::ViewListener, 167
UnbindCall
Coherent::UI::View, 148
UnbindObject
Coherent::UI::View, 148
Uninitialize
Coherent::UI::ViewContext, 151

UnregisterFromEvent
Coherent::UI::View, 148
Update
Coherent::UI::ImageData, 113
Coherent::UI::ViewContext, 151
UpdateView
MobileSurfaceView, 116
UseCameraDimensions
CoherentUIView, 97
UseURLCache
CoherentUISystem, 91
UserAgent
Coherent::UI::HTTPHeader, 111
UserAgent
Coherent::UI::ContextSettingsBase, 102
UserAgentType
Coherent::UI, 61
UsesSharedMemory
Coherent::UI::View, 149
Coherent::UI::ViewInfo, 153

VE_AbnormalTermination
Coherent::UI, 62
VE_ChildCompositionFailed
Coherent::UI, 62
VE_ClickThroughFailed
Coherent::UI, 62
VE_FailedCreation
Coherent::UI, 61
VE_FailedRenderingCreation
Coherent::UI, 62
VE_FrameAlreadyInFlight
Coherent::UI, 62
VE_FrameNotReady
Coherent::UI, 62
VE_FrameNotRequeted
Coherent::UI, 62
VE_InvalidCall
Coherent::UI, 62
VE_InvalidPath
Coherent::UI, 62
VE_QueryNotReady
Coherent::UI, 62
VE_Success
Coherent::UI, 61
Value
Coherent::UI::Binding::Value, 128, 129, 131
ValueType
Coherent::UI::Binding, 63
View
CoherentUIView, 97
ViewCreated
Coherent::UI::BrowserViewListener, 80
Coherent::UI::ViewListener, 167
ViewDestroyedHandler
CoherentUIView, 94
ViewErrorType
Coherent::UI, 61
ViewType
Coherent::UI, 62

WheelX
Coherent::UI::MouseEventData, 117
WheelY
Coherent::UI::MouseEventData, 118
Width
Coherent::UI::ViewInfo, 153
CoherentUIView, 97
WriteFile
Coherent::UI::FileHandler, 108
WriteLog
Coherent::UI::ILogHandler, 112
WriteMinidumps
Coherent::UI::FactorySettingsBase, 107

X
Coherent::UI::MouseEventData, 118
Coherent::UI::TouchEventData, 122
XPos
CoherentUIView, 98

Y
Coherent::UI::MouseEventData, 118
Coherent::UI::TouchEventData, 122
YPos
CoherentUIView, 98