Classes and Structures

Class List

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

- N FB — Primary location of FireBreath classes and utility functions
  - N Npapi
    - C NpapiBrowserHost — Provides a FB::BrowserHost implementation for Npapi
    - C NPObjectAPI — Provides a FB::JSObject implementation that wraps a NPObject*
    - C NpapiAsyncDrawService — Provides a D3d10AsyncDrawService implementation for NPAPI
  - N DOM — These are convenience objects used to access and manipulate DOM objects. They should never be created directly; instead, get the Window object or the Document object from the BrowserHost or create it with DOM::Element::create(obj) (or similar)
    - C Document — Abstraction for accessing and manipulating a DOM Document
    - C Element — DOM Element wrapper
    - C Node — DOM Node wrapper
    - C Window — DOM Window abstraction for manipulating and accessing the javascript window object that the plugin is contained in
  - N ActiveX
    - N AXDOM
      - C Document — ActiveX specific implementation of DOM::Document
      - C Element — ActiveX specific implementation of DOM::Element
      - C Node — Provides an ActiveX specific implementation of DOM::Node
      - C Window — ActiveX specific implementation of DOM::Window
    - C ActiveXAsyncDrawService — Provides a D3d10AsyncDrawService implementation for ActiveX
    - C ActiveXBrowserHost — Provides a BrowserHost implementation for ActiveX
    - C IDispatchAPI — Provide a JSObject implementation to wrap a IDispatch ActiveX object
  - C AsyncDrawService — Asynchronous drawing service base class
  - C BrowserPlugin — Browser-specific plugin base class

BrowserStream	This is the abstract base class (interface class) for a browser stream
Range	Specifies the range for a read range request (start to end) in bytes
BrowserStreamRequest	Information about an HTTP request to be made
DefaultBrowserStreamHandler	Simple implementation of a stream event handler from which you can derive your own stream events handler
FactoryBase	This is the base factory for a plugin. Every plugin *must* implement this class and override at least the `createPlugin()` method. To futher customize your plugin, you can override other methods to replace the `PluginWindow` or the NpapiPlugin class
PluginCore	Base class for all FireBreath Plugins
PluginEvent	Plugin event base class
AttachedEvent	Fired when a `PluginEventSink` is attached to a `PluginEventSource` (such as a `PluginCore` derived plugin object being attached to a `PluginWindow` to get events)
DetachedEvent	Fired when a `PluginEventSink` is detached from a `PluginEventSource` (such as a `PluginCore` derived plugin object being detached from a `PluginWindow` to get events)
ChangedEvent	Fired when a `PluginEventSink` has changed in some fundamental way that the plugin needs to know about, such as a different drawing context being provided by the browser to a `PluginWindow`
ResizedEvent	Fired when the plugin window is resized
ClipChangedEvent	Fired when the clipping of the plugin drawing area changes
RefreshEvent	Fired when the plugin should repaint itself (such as on windows when WM_PAINT is received)
QuickDrawDraw	Mac QuickDraw Draw event
CoreGraphicsDraw	Mac CoreGraphics Draw event
FocusChangedEvent	Fired when the focus changes
TimerEvent	Fired when a timer event is received; currently this only works on windows and is fired when WM_TIMER is received
KeyEvent	Fired for a key event
KeyUpEvent	Fired for a key up event
KeyDownEvent	Fired for a key down event
TextEvent	Used at least on Mac, possibly elsewhere; gives us text
MacEventCarbon	Carbon Mac event
MacEventCocoa	Cocoa mac event

`MouseEvent`	Fired when a mouse event occurs
`MouseMoveEvent`	Fired when the mouse moves
`MouseButtonEvent`	Fired when a mouse button event occurs
`MouseDownEvent`	Fired when a mouse down event occurs
`MouseDoubleClickEvent`	Fired when a mouse double click event occurs
`MouseUpEvent`	Fired when a mouse up event occurs
`MouseScrollEvent`	Fired when the user moves the scrollwheel
`MouseEnteredEvent`	Fired when the user moves mouse over the plugin
`MouseExitedEvent`	Fired when the user moves mouse away from the plugin
`StreamEvent`	Base class for all stream events
`StreamCreatedEvent`	This event is fired when the given stream was created
`StreamFailedOpenEvent`	This event is fired when a stream failed to open, e.g. the url was invalid or a seekable stream was requested while the server provided only a non-seekable stream
`StreamCompletedEvent`	This event is fired when a stream has completed downloading
`WindowsEvent`	Generic windows event. All windows events going through the winproc get first fired as a WindowEvent and then fired as the specific type, allowing you to handle them either way
`X11Event`	Generic X11 event
`X11NativeGdkEvent`	Class encapsulation for native Gdk event
`X11NativeGdkEventExpose`	Class encapsulation for native Gdk expose event
`PluginEventSink`	Plugin event sink; all objects that can receive `PluginEvent` s should inherit from this class. Most notably, `PluginCore` extends this class
`PluginEventSource`	Base class for any object, such as `BrowserStream` or `PluginWindow`, that needs to fire events to a `PluginEventSink` object (such as a `PluginCore` derived plugin class)
`PluginWindow`	`PluginWindow` is the base class for all `PluginWindow` objects
`HttpStreamResponse`	Data structure to hold the response to an HTTP Get request
`SimpleStreamHelper`	Helper class for making simple HTTP requests
`Timer`	`Timer` Utility

TimerService	TimerService Utility
WinMessageWindow	Creates a message window. Don't touch this if you don't understand what you are doing
CatchAll	When used as a parameter on a JSAPIAuto function this matches 0 or more variants – in other words, all other parameters from this point on regardless of type
MethodInfo	Used by FB::JSAPISimple to store information about a method
PropertyInfo	Used by FB::JSAPISimple to store information about a property
PropertyFunctors	Used by FB::JSAPIAuto to store property implementation details, created by FB::make_property()
AsyncLogRequest	This class is used by BrowserHost for the BrowserHost::AsyncHtmlLog method
BrowserHost	Browser host base class
JSAPI	JavaScript API class – provides a javascript interface that can be exposed to the browser
scoped_zonelock	Provides a helper class for locking
JSAPIAuto	JSAPI class with automatic argument type enforcement
JSAPIImpl	JavaScript API base class implementation – provides basic functionality for C++ JSAPI objects
JSAPIProxy	JavaScript API Wrapper – this can wrap any type of JSAPI object and be passed back to any browser that
JSAPISimple	Simple JSAPI implementation for those who for whatever reason don't want to use JSAPIAuto
script_error	Exception type; when thrown in a JSAPI method, a javascript exception will be thrown
invalid_arguments	Thrown by a JSAPI object when the argument(s) provided to a SetProperty or Invoke call are found to be invalid. JSAPIAuto will throw this automatically if the argument cannot be convert_cast to the type expected by the function
object_invalidated	Thrown by a JSAPI object when a call is made on it after the object has been invalidated
invalid_member	Thrown when an Invoke, SetProperty, or GetProperty call is made for a member that is invalid (does not exist, not accessible, only supports Get or Set, etc)
JSObject	Wraps a Javascript Object
SafeQueue	Basic thread-safe queue class
TypeIDMap	Bidirectional map between an identifier and a variant
URI	Data structure for dealing with URI strings
bad_variant_cast	Thrown when variant::cast<type> or variant::convert_cast<type> fails because the type of the value stored in the variant is not compatible with the operation
variant	Accepts any datatype, used in all interactions with javascript. Provides tools for getting back out the type you put in or for coercing that type into another type (if possible)

PluginEventMacCarbon	Mac OS X Carbon specific implementation of PluginEventMac
PluginEventMacCocoa	Mac OS X Cocoa specific implementation of PluginEventMac
PluginWindowlessWin	Windows specific implementation of PluginWindow
PluginWindowWin	Windows specific implementation of PluginWindow
PluginWindowX11	X11 specific implementation of PluginWindow
Content	Fired when DPI settings change (e.g. when moving webbrowser window between Retina and non-Retina displays)