

# file DefaultBrowserStreamHandler.h

DefaultBrowserStreamHandler.h

```
1 #pragma once
2 #ifndef DefaultBrowserStreamHandler_h__
3 #define DefaultBrowserStreamHandler_h__
4
5 /*****\
6 Original Author: Matthias (nitrogenycs)
7
8 Created: Feb 28, 2010
9 License: Dual license model; choose one of two:
10 New BSD License
11 http://www.opensource.org/licenses/bsd-license.php
12 - or -
13 GNU Lesser General Public License, version 2.1
14 http://www.gnu.org/licenses/lgpl-2.1.html
15
16 Copyright 2010 PacketPass, Inc and the Firebreath development team
17 \*****/
18
19 #include "BrowserStream.h"
20 #include "PluginEvents/AttachedEvent.h"
21
22 namespace FB {
23     class DefaultBrowserStreamHandler : public PluginEventSink
24     {
25     public:
26         BEGIN_PLUGIN_EVENT_MAP()
27         EVENTTYPE_CASE(FB::AttachedEvent, onStreamAttached, FB::BrowserStream)
28         EVENTTYPE_CASE(FB::DetachedEvent, onStreamDetached, FB::BrowserStream)
29         EVENTTYPE_CASE(FB::StreamCreatedEvent, onStreamCreated, FB::BrowserStream)
30         EVENTTYPE_CASE(FB::StreamDestroyedEvent, onStreamDestroyed, FB::BrowserStream)
31         EVENTTYPE_CASE(FB::StreamFailedOpenEvent, onStreamFailedOpen, FB::BrowserStream)
32         EVENTTYPE_CASE(FB::StreamOpenedEvent, onStreamOpened, FB::BrowserStream)
33         EVENTTYPE_CASE(FB::StreamDataArrivedEvent, onStreamDataArrived, FB::BrowserStream)
34         EVENTTYPE_CASE(FB::StreamCompletedEvent, onStreamCompleted, FB::BrowserStream)
35         END_PLUGIN_EVENT_MAP()
36
37         DefaultBrowserStreamHandler();
38
39         virtual
40         ~DefaultBrowserStreamHandler();
41
42         virtual bool onStreamAttached(FB::AttachedEvent *evt, FB::BrowserStream * Stream);
43         virtual bool onStreamDetached(FB::DetachedEvent *evt, FB::BrowserStream * Stream);
44         virtual bool onStreamCreated(FB::StreamCreatedEvent *evt, FB::BrowserStream * Stream);
45         virtual bool onStreamDestroyed(FB::StreamDestroyedEvent *evt, FB::BrowserStream *);
46         virtual bool onStreamDataArrived(FB::StreamDataArrivedEvent *evt, FB::BrowserStream *);
47         virtual bool onStreamFailedOpen(FB::StreamFailedOpenEvent *evt, FB::BrowserStream *);
48         virtual bool onStreamOpened(FB::StreamOpenedEvent *evt, FB::BrowserStream *);
49         virtual bool onStreamCompleted(FB::StreamCompletedEvent *evt, FB::BrowserStream *);
50         virtual const FB::BrowserStreamPtr& getStream() const;
51         virtual bool cancel();
52     protected:
53         virtual void setStream(const FB::BrowserStreamPtr& stream);
54         virtual void clearStream();
55     };
56     BrowserStreamPtr stream;
57 }
58 #endif // DefaultBrowserStreamHandler_h__
59
```

**FB::DefaultBrowserStreamHandler::~DefaultBrowserStreamHandler**  
virtual ~DefaultBrowserStreamHandler()  
Destructor.

**Definition:** DefaultBrowserStreamHandler.cpp:25

**FB::DefaultBrowserStreamHandler::onStreamCompleted**  
virtual bool onStreamCompleted(FB::StreamCompletedEvent \*evt, FB::BrowserStream \*)  
Called when the stream finished downloading successfully.

**Definition:** DefaultBrowserStreamHandler.cpp:75

**FB::DefaultBrowserStreamHandler::DefaultBrowserStreamHandler**  
DefaultBrowserStreamHandler()  
Default constructor.

**Definition:** DefaultBrowserStreamHandler.cpp:21

**FB::StreamCreatedEvent**  
This event is fired when the given stream was created.

**Definition:** StreamEvents.h:48

**FB::DefaultBrowserStreamHandler::onStreamAttached**  
virtual bool onStreamAttached(FB::AttachedEvent \*evt, FB::BrowserStream \*Stream)  
Called when the stream is attached to this handler (may have already been created).

**Definition:** DefaultBrowserStreamHandler.cpp:29

**FB::DefaultBrowserStreamHandler::onStreamDetached**  
virtual bool onStreamDetached(FB::DetachedEvent \*evt, FB::BrowserStream \*Stream)  
Called when the stream is detached from this handler (may not have been destroyed).

**Definition:** DefaultBrowserStreamHandler.cpp:36

**FB::BrowserStream**  
This is the abstract base class (interface class) for a browser stream.

**Definition:** BrowserStream.h:41

**FB::DefaultBrowserStreamHandler::setStream**  
virtual void setStream(const FB::BrowserStreamPtr &stream)  
Call to set the stream associated with this handler.

**Definition:** DefaultBrowserStreamHandler.cpp:85

**FB::DefaultBrowserStreamHandler::onStreamFailedOpen**  
virtual bool onStreamFailedOpen(FB::StreamFailedOpenEvent \*evt, FB::BrowserStream \*)  
Called when the stream failed to open.

**Definition:** DefaultBrowserStreamHandler.cpp:65

**FB::DefaultBrowserStreamHandler::getStream**  
virtual const FB::BrowserStreamPtr & getStream() const  
Call to get the stream associated with this handler. Returns empty shared\_ptr if none.

**Definition:** DefaultBrowserStreamHandler.cpp:80

**FB::DefaultBrowserStreamHandler::onStreamDestroyed**  
virtual bool onStreamDestroyed(FB::StreamDestroyedEvent \*evt, FB::BrowserStream \*)  
Called when the stream is destroyed.

**Definition:** DefaultBrowserStreamHandler.cpp:53

**FB::DefaultBrowserStreamHandler::onStreamDataArrived**  
virtual bool onStreamDataArrived(FB::StreamDataArrivedEvent \*evt, FB::BrowserStream \*)  
Called when data arrives.

**Definition:** DefaultBrowserStreamHandler.cpp:59

**FB::DefaultBrowserStreamHandler**  
Simple implementation of a stream event handler from which you can derive your own stream events hand...

**Definition:** DefaultBrowserStreamHandler.h:36

**FB::DefaultBrowserStreamHandler::stream**  
BrowserStreamPtr stream  
stream instance, don't use yourself

**Definition:** DefaultBrowserStreamHandler.h:181

**FB::StreamFailedOpenEvent**  
This event is fired when a stream failed to open, e.g. the url was invalid or a seekable stream was r...

**Definition:** StreamEvents.h:120

**FB::DefaultBrowserStreamHandler::onStreamOpened**  
virtual bool onStreamOpened(FB::StreamOpenedEvent \*evt, FB::BrowserStream \*)  
Called when the stream was opened successfully.

**Definition:** DefaultBrowserStreamHandler.cpp:70

**FB::PluginEventSink**  
Plugin event sink; all objects that can receive PluginEvent s should inherit from this class...

**Definition:** PluginEventSink.h:36

**FB::AttachedEvent**  
Fired when a PluginEventSink is attached to a PluginEventSource (such as a PluginCore derived plugin ...

**Definition:** AttachedEvent.h:29

**FB::DefaultBrowserStreamHandler::cancel**  
virtual bool cancel()  
Cancels the stream.

**Definition:** DefaultBrowserStreamHandler.cpp:42

**FB::DefaultBrowserStreamHandler::clearStream**  
virtual void clearStream()  
Call to clear the stream associated with this handler.

**Definition:** DefaultBrowserStreamHandler.cpp:95

**FB::DetachedEvent**  
Fired when a PluginEventSink is detached from a PluginEventSource (such as a PluginCore derived plugi...

**Definition:** AttachedEvent.h:39

**FB::StreamCompletedEvent**  
This event is fired when a stream has completed downloading.

**Definition:** StreamEvents.h:148

**FB::DefaultBrowserStreamHandler::onStreamCreated**

virtual bool onStreamCreated(FB::StreamCreatedEvent \*evt, FB::BrowserStream \*Stream)

Called when the stream was created.

**Definition:** [DefaultBrowserStreamHandler.cpp:48](#)

---