class FB FactoryBase

FB::FactoryBase Class Referenceabstract

Public Member Functions | List of all members

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. More...

`#include "FactoryBase.h"`

Inherits noncopyable.

Public Member Functions

virtual **FB::PluginCorePtr**	**createPlugin** (const std::string &mimetype)=0 Creates a **FB::PluginCore** derived user plugin object. This must be implemented for all plugin projects. More...
virtual void	**globalPluginInitialize** () Global plugin initialization. More...
virtual void	**globalPluginDeinitialize** () Global plugin deinitialization. More...
std::string	**getPluginName** () Returns the name of the plugin. To change the name of your plugin edit PluginConfig.cmake. More...
std::string	**getPluginName** (const std::string &mimetype)
std::string	**getPluginDescription** () Returns the description of the plugin. To change the description of your plugin edit PluginConfig.cmake. More...
std::string	**getPluginDescription** (const std::string &mimetype)
virtual FB::Npapi::NpapiPluginPtr	**createNpapiPlugin** (const FB::Npapi::NpapiBrowserHostPtr &host, const std::string &mimetype) Creates a npapi plugin. The default implementation creates a NpapiPluginWin, NpapiPluginX11, or NpapiPluginMac depending on the platform. Only in very very rare cases will you need to override this method. More...
virtual void	**getLoggingMethods** (FB::Log::LogMethodList &outMethods) Called by the logger to discover which log methods should be used. More...
virtual FB::Log::LogLevel	**getLogLevel** () Called by the logger to discover what loglevel to use. More...
virtual **PluginWindowWin** *	**createPluginWindowWin** (const WindowContextWin &ctx) Creates a **PluginWindowWin** derived plugin window object. More...
virtual **PluginWindowlessWin** *	**createPluginWindowless** (const WindowContextWindowless &ctx) Creates a **PluginWindowlessWin** derived plugin window object. More...
virtual PluginWindowMacICA *	**createPluginWindowMacICA** () Creates a **PluginWindow** derived plugin window object for Invalidating CoreAnimation. More...
virtual PluginWindowMacCA *	**createPluginWindowMacCA** () Creates a **PluginWindow** derived plugin window object for CoreAnimation. More...
virtual PluginWindowMacCG *	**createPluginWindowMacCG** () Creates a **PluginWindow** derived plugin window object for CoreGraphics. More...
virtual **PluginWindowX11** *	**createPluginWindowX11** (const WindowContextX11 &ctx) Creates a **PluginWindowX11** derived plugin window object for X11. More...

Detailed Description

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.

```
// Example implementation:
class PluginFactory : public FB::FactoryBase
{
public:
FB::PluginCorePtr createPlugin(const std::string& mimetype)
{
return boost::make_shared<MyPluginObject>();
}
void globalPluginInitialize()
{
MyPluginObject::StaticInitialize();
}
void globalPluginDeinitialize()
{
MyPluginObject::StaticDeinitialize();
}
};
FB::FactoryBasePtr getFactoryInstance()
{
static boost::shared_ptr<PluginFactory> factory = boost::make_shared<PluginFactory>();
return factory;
}
```

Since

1.3 RC2

Definition at line **100** of file **FactoryBase.h**.

The documentation for this class was generated from the following files:

- **FactoryBase.h**
- **FactoryBase.cpp**
