

Web Client Standard Plugin Interfaces

Constructor & Destructor

Every plug-in module is derived from a base class which contains two functions that represent constructor and destructor functions. These functions are **i2b2.PLUGINCODE.Init()** and **i2b2.PLUGINCODE.Unload()**; The **Init()** function is called after the plug-in has been selected by the user from the plugin viewer list and has successfully had its initial HTML loaded into the plugin viewer window. The **Unload()** function is called after the user has selected another plug-in from the plugin viewer list. The framework will wait until the **Unload()** function returns before it begins to load the newly selected plug-in. If the **Unload()** function returns false, the framework will cancel loading of the newly selected plug-in.

Function Name	Parameters	Description
Init	refDIV	Executed when a plug-in is loaded into the display container DIV element. A reference to the plug-in's main container DIV is passed as the first parameter.
Unload	(none)	Executed before the plug-in DIV is destroyed. This function must return true for the framework to complete the unload process. Returning false will cancel the unload request.

Example constructor and destructor functions from a hypothetical "Hello World" example would be:

```
i2b2.HELLO.Init = function(loadedDiv) {  
    // this function is called after the HTML is loaded into the viewer DIV  
    i2b2.HELLO.view.containerDiv = loadedDiv; //save DIV reference for later use  
  
    alert("Hello World: This message is from the initialization routine.");  
};  
i2b2.HELLO.Unload = function() {  
    // this function is called before the plugin is unloaded by the framework  
    alert("Hello World: This message is from the unload routine.");  
    // The next line is the boiler plate code that should work in many instances.  
    // This routine can also be used to save the state of the plugin so that work  
    // can seamlessly resume the plugin is loaded.  
    return confirm("Are you sure you want to unload the Hello World plugin?");  
};
```

Visual Functions - Resize, Hide, Show

Certain GUI events may require your plug-in to resize, display or hide its GUI and GUI elements (for example, floating dialog boxes). To facilitate this functionality the plugin viewer framework will execute the following functions at appropriate times:

Function Name	Parameters	Description
wasHidden	(none)	Executed immediately after the plug-in's main display DIV is hidden.
wasShown	(none)	Executed immediately after the plug-in's main display DIV is shown.
Resize	(Object)	Executed immediately after the plug-in's main display is resized. The structure of the data object passed is as follows: { height: <i>number</i> , width: <i>number</i> , left: <i>number</i> , top: <i>number</i> }

Options Button Hook

Whenever a plug-in module is loaded into the Plugin Viewer window it is checked to see if it contains a handler function called **i2b2.PLUGINCODE.ShowOptions()** whereas **PLUGINCODE** is the namespace code for your plug-in module.

If the plug-in module does have a **ShowOptions()** function, it is called by the Plugin Viewer object whenever the "Show Options" icon is clicked while the plug-in is loaded. No parameters are passed into the **ShowOptions()** function.

The "Show Options" icon used to fire the plug-in's ShowOptions() function

