

SmileTiger WebPresentation™ Plug-in Developer Guide



SmileTiger Software

SmileTiger Software Corporation

11615 Sir Francis Drake Drive
Charlotte, NC 28277 USA
Phone: + 1 704 321 9068
Email: info@smiletiger.com
<http://www.smiletiger.com>

Revision 5.4, Oct. 2003

Table of Contents

1	OVERVIEW	3
2	API AND LIBRARY.....	4
3	CREATE A CLIENT PLUG-IN COMPONENT	5
4	CREATE A SERVER PLUG-IN COMPONENT	6
5	DEPLOY PLUG-IN COMPONENTS	7
5.1	To deploy a client plug-in.....	7
5.2	To deploy a server plug-in	7

1 Overview

This documentation is the guide for developers who are interested in developing plug-in components in the SmileTiger WebPresentation Software.

SmileTiger WebPresentation offers a lot of benefits for developers. The system provides rich collaboration functions, firewall-friendly communication mechanism, automatic deployment mechanism, redundancy mechanism, and load balancing mechanism for developers to easily create enterprise web-based applications. Developers will be able to create plug-ins in both the server side and the client side to accomplish their application's business logic. Some application examples are: multiple-user games, group collaboration services and e-learning system.

A client plug-in is a component embedded in the main window of the SmileTiger WebPresentation client. This component is developed in Java language and is packaged as a JAR file.

A server plug-in is a component loaded by SmileTiger WebPresentation Server on demand. This component is developed in Java language and is packaged as a JAR file. The server plug-in and the client plug-in can work together to accomplish an application's task.

A complex application developed on SmileTiger WebPresentation System may contain multiple client plug-ins and server plug-ins. Each plug-in is identified by a unique ID, and other plug-ins communicate with it using this ID. All plug-ins collaborating to accomplish an application task are deployed to the same virtual conference room.

2 API and Library

To develop plug-ins, you will need to use the SmileTiger WebPresentation API and its corresponding library to develop and compile your Java project. The API and library are packaged in a ZIP file, and you should use WinZip and a similar application to open it.

The detailed API information is available under the javadoc directory of the zip file. The library, stwpapi.jar, is under the root directory of the zip file

3 Create a Client Plug-in Component

A client plug-in component contains main class, helper classes and resources. The main class is a subclass of **com.smiletiger.wpapi.WPPanel**, which is a subclass of **java.awt.Panel**.

To create a plug-in main class:

1. Create a new class, which extends **com.smiletiger.wpapi.WPPanel**
2. Override the following methods in **com.smiletiger.wpapi.WPPanel**

```
public char[] getProgramID();
    // Must return a unique four-character array.

public String getProgramName();
    // Return the short name of this component. This name is shown on
    the bottom tab of the main window.

public String getProgramImage();
    // Return the image path of this component. The image path
    represents the location of this image in the JAR file, and it is
    written in a format like "/com/company/product/image.gif". This
    image is shown on the bottom tab of the main window.

public void open(String param);
    // called when the plug-in is loaded

public void close();
    // called when the plug-in is unloaded
```

3. Override the message handles:

```
public void serverMessageReceived(byte type, byte[] data);

public void messageReceived(short src, byte type, byte[] data);

public void memberListReceived(Hashtable h);
```

4. Develop the plug-in logic by using the functions available in the API, Java and the platform.
5. Create a JAR file to package the plug-in files.

Please read the Java Doc file and the sample plug-in for more information.

4 Create a Server Plug-in Component

A server plug-in component contains main class, helper classes. The main class is a subclass of **com.smiletiger.wpsapi.WPSPlugin**.

To create a plug-in main class:

1. Create a new class, which extends **com.smiletiger.wpsapi.WPSPlugin**
2. Override the following method of **com.smiletiger.wpsapi.WPSPlugin**

```
public char[] getProgramID();
    // Must return a unique four-character array.

public void open(String param);
    // called when the plug-in is loaded

public void close();
    // called when the plug-in is unloaded
```

3. Override the message handles:

```
public void messageReceived(short src, byte type, byte[] bs);

public void userJoined(ConferenceEvent e);

public void userLeft(ConferenceEvent e);
```

4. Develop the plug-in logic by using the functions available in the API, Java and the platform.
5. Create a JAR file to package the plug-in files.

Please read the Java Doc file and the sample plug-in for more information.

5 Deploy Plug-in Components

After developing, compiling and packaging of the plug-ins of an application, a developer or a SmileTiger WebPresentation server administrator needs to deploy the plug-ins to the SmileTiger WebPresentation server. All plug-ins for an application should be deployed to the same virtual conference room.

5.1 To deploy a client plug-in

Each virtual conference room has a configuration file under directory <STWPS HOME>/wwhome/default/conferences/. To deploy a client plug-in to a virtual room, you need insert a <plugin> tag under the parent tag <conference> in the virtual room configuration file. Below is a sample:

```
<plugin name="Sample"
  type="client"
  param=""
  file="cpsample.jar"
  class="com.smiletiger.plugin.sample.CPSample"/>
```

The type parameter must be 'home' or 'client'. Type 'home' means that the plug-in is automatically loaded when a user enters this conference room. Type 'client' means that the plug-in is sent to the user's machine, but is not activated until the user enables it.

The packaged Jar file need to be put under directory <STWPS HOME>/cplugins/ .

5.2 To deploy a server plug-in

To deploy a client plug-in to a virtual room, you need insert a <plugin> tag under the parent tag <conference> in the virtual room configuration file. Below is a sample:

```
<plugin name="Sample"
  type="server"
  param=""
  file="cssample.jar"
  class="com.smiletiger.plugin.sample.CSSample"/>
```

The type parameter must be 'server'.

The packaged Jar file need to be put under directory <STWPS HOME>/splugins/ .