

March 6th

08

# Hacking Servoy's Frame Title & Icon

Tom Parry

Prospect IT Consulting, Inc. [venture@compmore.net](mailto:venture@compmore.net)

Java for Servoy Developers

*Community support for all things Java and Servoy*

<http://java-servoy.blogspot.com>

## **This is a Blog Post!**

This is a PDF version of a blog post located at <http://java-servoy.blogspot.com>. It is not a formal whitepaper or other similar guide and is intended to be a casual writing. Apart from styling and formatting differences, and the possible addition of graphics it is typically identical to its online counterpart.

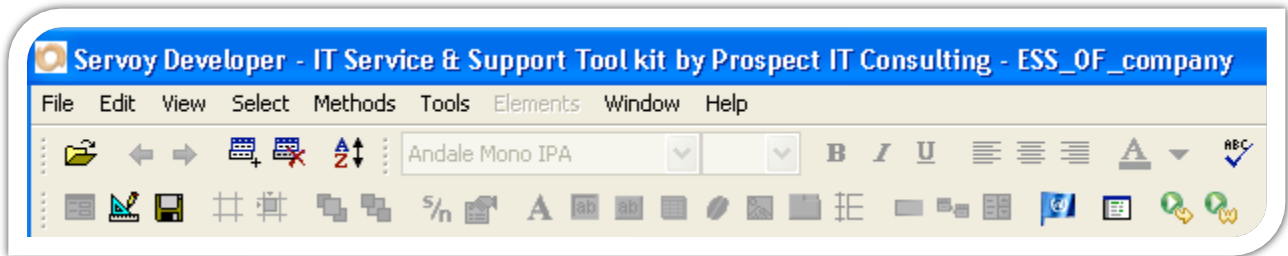
Sometimes things just look better in print 😊

## Hacking Servoy's Frame Title & Icon

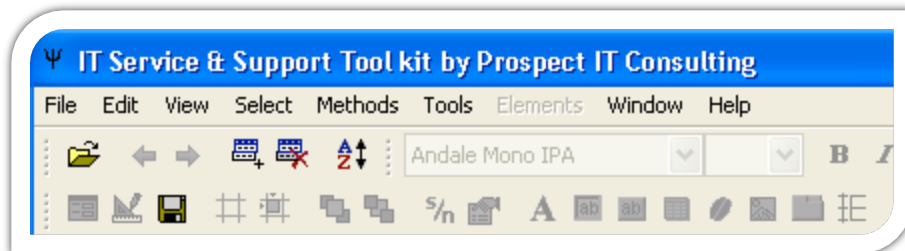
Have you ever wanted to change the title or the icon image when you deploy your Servoy solution?

In this blog post I will show you how it can be done using some Java knowledge.

Before:



After:



If you were to develop a Java application you would be able to set both the icon image as well as the title of any window. However for your Servoy solution we will deal only with the top-level window since it is the one the client will see loading first.

The first Java class to be aware of is the JFrame which is an implementation of the Frame class from the java.awt.Frame package (see <http://java.sun.com/javase/6/docs/api/java/awt/Frame.html>). However we will not be using JFrame but rather a static method of the Frame class:

```
public static Frame[] getFrames()
```

Note this this method returns an array of Frames created by this application. By hacking (actually examining the contents of each array element) I was able to determine that array element zero is the top level window used by Servoy by comparing the title property to what is displayed. Now this may change in future if Servoy decides to design differently – but until then this seems to work (Let me know if there is a problem and then we can all share).

In Servoy javascript method get the array of frames in use:

```
var frames = Packages.java.awt.Frame.getFrames();// [1]
```

Once having obtained the frame to use we can then invoke any method for that frame.

For this experiment I was interested in the title and the icon. The relevant setter and getter methods are:

```
getTitle() setTitle( String title )  
getIconImage() setIconImage( Image image )
```

So the title can be changed as in:

```
frames[0].setTitle('some other title');// [2]
```

There have it – a two line change of title ( [1] and [2])!

In my case I had only a gif as the icon of my choice (the greek letter psi) and this is not the type of object that the setIconImage method wants – it wants an Image. So a little more research was needed. After doing some web searches and reading the Java tutorials we finally get to a simple approach.

```
var kit = new Packages.java.awt.Toolkit.getDefaultToolkit();//[3]
var imageToUse = kit.getImage(myIconImageLoc);//[4]
frames[0].setIconImage(imageToUse);//[5]
```

First we need to use the “toolkit” from Java. Then we create an image type of object from the gif using the toolkit `getImage()` method. Finally we can set the icon image into the frame. A three line fix ( [3][4][5]).

The `myIconImageLoc` refers to a gif or other image file on your system. The location of this could be specified as the directory containing Servoy files. I will let you experiment with that part.

I shall leave it as a challenge for others out there if they wish to experiment with the other interesting methods such as:

```
getMenuBar setMenuBar( Menubar mb )
isResizable setResizable( boolean resizable )
```

According to the Frame class description if you use `setMenuBar(null)` then any menu bar is removed – which could be useful.

Perhaps setting the `resizable` property to not resizable by the user is useful for solutions that should not be changed in size by the user.

In my own solutions I like to create an all purpose method, and because of Servoy design it has to be a global method, to do all the hard work. I then create another global variable so that I hide the implementation and only deal with the interface.

My generic method is like this screen snapshot:

```
//2008.March.01
/*
   Sets the top level window title and/or icon.
   Ensure this is called after servoy has finished messing with the title.
4
5
6   Default image is/was greek letter psi: 'd:/temp/icons/images/Math/psi.gif';
7   Default title is/was: 'IT Service & Support Toolkit by Prospect IT Consulting'
8
9   The place where it is best to use this is...
10 */
11 var myTitle = null; //no change
12 var myIconImageLoc = null; //no change
13 if (arguments && (arguments.length > 0)) {
14     if (arguments[0]) {
15         myTitle = arguments[0];
16     }
17     if (arguments[1]) {
18         var myIconImageLoc = arguments[1];
19     }
20
21     var frames = Packages.java.awt.Frame.getFrames();
22     if (myTitle) {
23         var prevTitle = frames[0].getTitle(); //save it or do something with it
24         frames[0].setTitle(myTitle); // {0} is the top most frame - set a new title
25     }
26     if (myIconImageLoc) {
27         //var myIconImage = new Packages.java.awt.image.BufferedImage;
28
29         var kit = new Packages.java.awt.Toolkit.getDefaultToolkit();
30         var imageToUse = kit.getImage(myIconImageLoc);
31         frames[0].setIconImage(imageToUse);
32     }
33 } //if arguments
34
35 return;
```

I added a couple of optional arguments so that either or both the title and the icon can be changed.

My interface method is like this:

```
1 //2008.Mar.01
2 /*
3     set default title means setting the title and/or the icon default values.
4 */
5 globals.PITC_setTopTitle(
6     'IT Service & Support Tool kit by Prospect IT Consulting',
7     'd:/temp/icons/images/Math/psi.gif'
8 );
9 //application.updateUI();
10
```

The `application.updateUI()` method may or may not be required – try it out.

Now the next hardest part was trying to figure out why the generic interface method calling the generic method works in the developer debugger session but not otherwise. Well I have not quite figured out the whole story and no doubt there might be some one who can find a better approach but this is what I finally figured out would work for me.

In the Servoy developer the top level title changes as you change forms – which is nice to have. However when we run smart client we have the option of a title in the solutions settings part.

Through some experimentation I concluded that I would have to call my default interface method every “time an onShow event occurs for “every” form in order not to have the Servoy title pop back. Is there a better event that can trigger calling the default title method? Please let me know!

So although this is not foolproof method (the icon seems to be set but the title does not seem to work in developer very well or not at all but both work in smart client).

I would like to hear from others who might try out this idea some more and improve it.

I have not tried this in the web client – so not sure of the effect there.

Regards,

Tom