

## Busy Plugin - Defect #44

### Other dialog won't show up after unlocking busy indicator

06/26/2010 09:26 PM - Patrick Talbot

<b>Status:</b>	Rejected	<b>Start date:</b>	06/26/2010
<b>Priority:</b>	None	<b>Due date:</b>	
<b>Assignee:</b>	Patrick Talbot	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Browser (if web client):</b>			
<b>Description</b>			
Reported by <a href="mailto:lareina@kelman.com">lareina@kelman.com</a> , Sep 10, 2009			
What steps will reproduce the problem? 1. use busy indicator for a servoy function 2. when errors happen, unlock the busy indicator and call "application.showFormInDialog()" in that function. The dialog won't show up.			
What is the expected output? What do you see instead? The error dialog should show up but I didn't see anything.			
What version of the product are you using? On what operating system? I'm using the latest version of busy.jar			
Please provide any additional information below. I know this issue is caused by the separate thread as running busy indicator will be on a separate thread. What I need to know is if there is a workaround to solve this issue?			
Thanks,			
Comment 1 by <a href="mailto:lareina@kelman.com">lareina@kelman.com</a> , Sep 10, 2009 My solution is a web client solution. Users submit a form and then a function will do data processing. This might take long time so the busy indicator is needed.			
Comment 2 by <a href="mailto:lareina@kelman.com">lareina@kelman.com</a> , Sep 30, 2009 The same problem happens to the other situation: I wrote a plugin to use IWebClientPluginAccess.serveResource(file, byte[], mimeType) and IWebClientPluginAccess.showURL() to download data. This plugin will be called in a function of a form. It works fine without busy plugin. But it threw out java null pointer exception when calling IWebClientPluginAccess.serveResource(file, byte[], mimeType) after the busy plugin was implemented to this function.			
Delete comment Comment 3 by <a href="mailto:lareina@kelman.com">lareina@kelman.com</a> , Sep 30, 2009 My app is web client. The example code for "Comment 2": 1. Without busy plugin: function test() { plugins.myplugin.download(Object[]); } 2. With busy plugin: function test1() { var params = { processFunctionName: 'test', message: 'Exporting Grid.. Please wait.....', opacity: 0.7, paneColor: '#000000', showCancelButton: true, cancelButtonText: 'Cancel', callBackFunction: globals.callBack }; plugins.busy.block( params );			

```
}  
function test()  
{  
    plugins.myplugin.download(Object[]);  
    plugins.busy.unblock();  
}
```

Comment 4 by alicelis.hurtado, Feb 09, 2010

How I can do work the busy plugin in a Form showed in Dialog mode?, I'm using the latest version of busy.jar

Comment 5 by ptalbotCA, Feb 17, 2010

Unfortunately, due to limitations of the web client and the fact that you cannot have more than one synchronous call to the Servoy server, I have to mark this problem as not fixable.

No workaround has been found, so you will not be able to use the plugin in such a case. Sorry.

## History

---

### #1 - 07/17/2010 07:28 AM - Patrick Talbot

- Priority changed from Normal to None

### #2 - 08/01/2010 07:28 AM - Patrick Talbot

- Status changed from Rejected to Reopened

- Assignee set to Patrick Talbot

I'm reopening this one as the change of calls using a specific Thread introduced with v2.0 might have solved it. Please try and give me some feedback here.

### #3 - 08/06/2010 07:15 PM - Paul Coward

- Status changed from Reopened to Feedback

Patrick Talbot wrote:

I'm reopening this one as the change of calls using a specific Thread introduced with v2.0 might have solved it. Please try and give me some feedback here.

Hi Patrick,

I am testing the latest release v2.0 and am getting the following error, so it appears as if the thread issue is not solved.

Error setting form foundset, There is no application attached to current thread Thread-1107  
> org.apache.wicket.WicketRuntimeException: There is no application attached to current thread Thread-1107  
Error setting form foundset, There is no application attached to current thread Thread-1107  
> org.apache.wicket.WicketRuntimeException: There is no application attached to current thread Thread-1107

### #4 - 08/06/2010 07:50 PM - Patrick Talbot

Paul Coward wrote:

Patrick Talbot wrote:

I'm reopening this one as the change of calls using a specific Thread introduced with v2.0 might have solved it. Please try and give me some feedback here.

Hi Patrick,

I am testing the latest release v2.0 and am getting the following error, so it appears as if the thread issue is not solved.

Error setting form foundset, There is no application attached to current thread Thread-1107

org.apache.wicket.WicketRuntimeException: There is no application attached to current thread Thread-1107

Error setting form foundset, There is no application attached to current thread Thread-1107

org.apache.wicket.WicketRuntimeException: There is no application attached to current thread Thread-1107

Thanks for the feedback!

This is a different error though. Someone reported something similar to this, and I will see what I can do: basically the processFunction is called inside a different Thread and it seems that this Thread has no access a certain number of Servoy objects... I will talk about this with Johan Companer to see what can be done about it.

**#5 - 08/18/2010 08:03 PM - Paul Coward**

Hi Patrick,

Thank you for the update.

I was wondering what the progress might be on this particular issue? Our solution currently requires that functionality and I am needing to give my superiors an update on this issue.

Thanks again

**#6 - 08/19/2010 05:54 AM - Patrick Talbot**

- Status changed from Feedback to Rejected

- % Done changed from 0 to 100

This is actually a much bigger issue than expected.  
Basically, everything I attempted to address this have failed.

The problem is that once a method is called from a Wicket Request, Wicket will block all events until this method is terminated.

This was already the case before 5.2, but the asynchronous call coming in the meantime were still set to execute, while this is not the case anymore, Servoy 5.2 is now adding them to a queue of to be executed in a ScheduledThreadExecutor.

I first noticed that from Servoy 5.2 the call from the cancel method never reached the component, even though the Ajax called was received by the server.

Discussing this with Johan, he explained to me that this was actually a whole in previous implementation and that now every Ajax call was put in a List and queued, until the first Request was finished, meaning that even though I execute the callback method asynchronously, this code is no more executed before the first method has finished.

Meaning that the cancel call is being executed AFTER the whole process is finished (kind of pointless, really!)

Johan told me that the only way to workaround this was to launch the first method in my own Thread, thus 'by-passing' the normal threading roadblocks. He warned me that there might be potential problem with this, which I understood in terms of concurrency problems, so I decided to give it a shot, and why I thought this could resolve your issue too.

The process method in this implementation was done in a non-Servoy managed Thread, while the cancel call was now going through the normal Servoy threading model. It turns out that launching the 'processFunction' in a non-Servoy managed Thread prevents from executing sensitive access to the database and the UI, meaning that some methods will work and some just won't. I suppose all databaseManager methods are expecting to be executed by a Servoy client Thread, which is not the case here.

I must say that I exhausted all possibilities here, and the worst thing is that even less complex calls using a cancel callback will cease to work anyway with Servoy 5.2.

The cancel callback will always be scheduled to execute after the end of the process function.

So the cancel option will no longer be available with Servoy 5.2 in web client...

Only possible way to work now with long process will be to use the headless\_client plugin to create another client that will do the work and callback when done.

Then again, because the headless client will not be the client, some of the things done in this new client will not be immediately visible in the main client, but most options will be and the database updates will hopefully trigger databroadcasting to the main client.