Busy Plugin - Defect #475

Plugin does not block the screen -when plugins.busy.unblock() function is called. If I remove it, it works!

03/28/2012 02:59 AM - Juan-Carlos Sanchez

Status:	Closed	**Start date:**	03/28/2012
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
Browser (if web client):			

Description

*I have this code. The plugin do not block the screen ~~when plugins.busy.unblock() function is called. If I remove it (comment it out), the blocking of the screen works -- but of course, stay blocked :~~)

/** * @properties={typeid:24,uuid:"F87FD628-DB37-4BDC-9A15-F3CB07842246"}
*/
function goToLastRecord_busyNotice(){
// busy plugin needs some instructions:
var busyParameters = {
// function that contains the lengthy process:
processFunction: processGoToLastRecord,
// setup the look of the blocking
message: 'Uploading entire database and going to last record\n... please WAIT!',
opacity: 0.5,
paneColor: '#000000',
textColor: '#f5f5f5',
// use a cancel button:
showCancelButton: true,
cancelButtonText: 'Stop process'
};
// launch UI blocking and process at the same time:
plugins.busy.block(busyParameters);
}

/** * @properties={typeid:24,uuid:"D79B0CE3-DA1D-437E-AA24-A05EDB165DE0"}
*/
function processGoToLastRecord() {
application.output("process started"); *

```
// always put your process in a try/catch/finally block:
    try {
        // your long process in a loop:
          while(currentcontroller.recordIndex < currentcontroller.getMaxRecordIndex())
         {
            // do something here
            currentcontroller.recordIndex = currentcontroller.getMaxRecordIndex();
            application.output("process end");
            // check for each iteration if the user has canceled:
            if (plugins.busy.isCanceled()) {
                application.output("process canceled");
                break;
            }
         }
    // trap any exception here (or throw)
    } catch (e) {
        application.output("exception caught");
        application.output(e);

// always unblock in a finally block (this will always happen!)
```

```
    } finally {
        plugins.busy.unblock();
    }
    application.output("process end");
}
```

The other strange thing is that this happens only in this solution. I checked running the method as globals, I created new forms, etc. Just an strange thing to keep in mind.

History

#1 - 03/28/2012 04:14 AM - Patrick Talbot

And this is Smart client?
I know of some troubles with Web client, but that's the first time I hear there is any in Smart...

If your code is really:

```
while(currentcontroller.recordIndex < currentcontroller.getMaxRecordIndex())
{
   // do something here
   currentcontroller.recordIndex = currentcontroller.getMaxRecordIndex();
   // ...
```

Then basically you are exiting the loop as soon as you enter it, thus the busy screen doesn't even have the time to show ;) Because you set record index to the max, thus it's not < max anymore. Then the rest of the time you are waiting is happening while Servoy is retrieving the foundset from the database (and the finally block has already been called).

Unless you are doing something instead of the "// do something here" that you are showing there, in which case, I'd like to know what exactly...

Do you have a sample???

#2 - 03/30/2012 01:06 AM - Juan-Carlos Sanchez

Patrick Talbot wrote:

> And this is Smart client?
> I know of some troubles with Web client, but that's the first time I hear there is any in Smart...
>
> If your code is really:
> [...]
>
> Then basically you are exiting the loop as soon as you enter it, thus the busy screen doesn't even have the time to show ;) Because you set record index to the max, thus it's not < max anymore. Then the rest of the time you are waiting is happening while Servoy is retrieving the foundset from the database (and the finally block has already been called).
>
> Unless you are doing something instead of the "// do something here" that you are showing there, in which case, I'd like to know what exactly...
>
> Do you have a sample???

Man, all I want to do is to load the entire dataset and go to the last row of a table with 300000 records (yep, this is a user request. The user wants to see all the records loaded before doing and export ...)

This is the function I use to do that:

function processGoToLastRecord(){
while(currentcontroller.recordIndex < currentcontroller.getMaxRecordIndex())		{currentcontroller.recordIndex = currentcontroller.getMaxRecordIndex();}
}

Can you help me with this? I don’t need the 'try', capture an error etc. All I need to do is the ‘unblock’ after I load all records and get to the last record. Thanks! JC

#3 - 03/30/2012 08:19 PM - Juan-Carlos Sanchez

- File notes_js.pdf added

Patrick,

The attached code used to work just fine before I updated from Servoy 5.2.9 to 5.2.12.

If I try this while loop take longer to go to the last record:

```
while(currentcontroller.recordIndex < currentcontroller.getMaxRecordIndex())
         {currentcontroller.recordIndex ++;}
```

It doesn't work either anyway. What am I doing wrong? Did something change in Servoy 5.2.12 parser that makes my code bad now? As I said, it used to work just fine with the Busy Plugin.

Also, I think it’s my code, because the plugins still works fine in other solution where is used - e.g. while the SmartDoc does its job.

Any idea?

JC

#4 - 03/30/2012 08:24 PM - Patrick Talbot

- Status changed from New to Closed

Apart from the fact that this is just crazy requirement and you should tell that to the user (like he's going to review 300.000 records, yeah!)

To make it work, change your processGoToLastRecord() method to:

```
function processGoToLastRecord() {
    application.output("process started");
    application.updateUI(); // <-- Give a chance to the plugin to show!!!
    // always put your process in a try/catch/finally block:
    try {
        // your long process in a loop:
        while (currentcontroller.recordIndex < currentcontroller.getMaxRecordIndex()) {
            currentcontroller.recordIndex = currentcontroller.getMaxRecordIndex();
            application.output(currentcontroller.getMaxRecordIndex());
            // check for each iteration if the user has canceled:
            application.sleep(120); // Give some time to the plugin to process a click on cancel button!
            if (plugins.busy.isCanceled()) {
                application.output("process canceled");
                break;
            }
        }
        // trap any exception here (or throw)
    } catch (e) {
        application.output("exception caught");
        application.output(e);
        // always unblock in a finally block (this will always happen!)
    } finally {
        plugins.busy.unblock();
    }
    application.output("process end");
}
```

#5 - 03/30/2012 08:54 PM - Juan-Carlos Sanchez

Patrick Talbot wrote:

> Apart from the fact that this is just crazy requirement and you should tell that to the user (like he's going to review 300.000 records, yeah!)
>
> To make it work, change your processGoToLastRecord() method to:
>
> [...]

Thank you Patrick. You code worked perfectly. I guess the 'application.updateUI()' did the trick here!

Without giving the option to the user to cancel (I commented out application.sleep), it took less than 15 second to Servoy to get to the last record (MySQL 5.5 backend). I think this is pretty fast. I'll do this for this specific 87 years old user (one big fish faculty). All others don't care. I will give the rest the option to cancel or bear the pain of going through the loop with the application.sleep option for canceling (I don't want them to get into the habit of getting everything at once).

Thanks! JC

Files

notes_js.pdf	10.5 KB	03/30/2012	Juan-Carlos Sanchez