

## User Interface - Bug #7824

**Simplify idle connection settings and apply the default configuration for the web client debugging if the debug agent is set.**

09/22/2023 02:27 AM - Sergey Ivanovskiy

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Sergey Ivanovskiy	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>case_num:</b>	
<b>billable:</b>	No		
<b>vendor_id:</b>	GCD		
<b>Description</b>			

### History

#### #1 - 09/22/2023 02:39 AM - Sergey Ivanovskiy

These two timeout parameters of webClient

```
<node class="integer" name="maxIdleTime">
  <node-attribute name="value" value="300000"/>
</node>
<node class="integer" name="maxHttpIdleTimeout">
  <node-attribute name="value" value="300000"/>
</node>
```

define timeouts of inactivity (idle timeouts) in milliseconds for a websocket session and its http/https session. maxIdleTime is used by WebClientProtocol.onConnect by websocket session and maxHttpIdleTimeout is used by GuiWebDriver.init to set http configuration for the embedded web server that impacts http/https sessions.

#### #2 - 09/22/2023 06:47 AM - Sergey Ivanovskiy

- Status changed from New to WIP
- Assignee set to Sergey Ivanovskiy

I created task branch 7824a.

#### #3 - 09/22/2023 07:28 AM - Sergey Ivanovskiy

Also "maxIdleTimeout" is erroneously (or intentionally) used by WebServer to set its http/https session idle timeout.

#### #4 - 09/22/2023 08:02 AM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

Also "maxIdleTimeout" is erroneously (or intentionally) used by WebServer to set its http/https session idle timeout.

Does it make sense to set `getHttpConfiguration().setIdleTimeout(0)` for `WebServer`? It means infinite idle timeout for REST, SOAP and `WebHandler`?

**#5 - 09/22/2023 08:17 AM - Sergey Ivanovskiy**

`maxHttpIdleTimeout` is set to 0 (infinite timeout) if this setting is not defined in the directory under `webClient` container. But `maxIdleTimeout` is set to -1 (connection timeout) if this setting is not defined in the directory and it has consequences when debugging the large customer application.

**#6 - 09/22/2023 08:22 AM - Sergey Ivanovskiy**

Now I think that all these parameters should be present. It seems that it would be enough to fix the correct default value for `maxIdleTimeout` that should be 0 instead of -1. Are there another ideas?

**#7 - 09/22/2023 10:06 AM - Sergey Ivanovskiy**

There are the web server idle timeout and the embedded web server idle timeout. The first one is set for `WebServer` and the second one is set for `EmbeddedWebServerImpl` by `WebGuiDriver.init`. How can we simplify these settings? I found that if the web server idle timeout is set to connection timeout, then it can affect the embedded web server idle timeout for the large customer application during the web client debugging.

**#8 - 09/25/2023 08:49 AM - Sergey Ivanovskiy**

*- Subject changed from `There are two different connection timeout parameters that are effected by each other so they must be merged to Simplify idle connection settings and apply the default configuration for the web client debugging if the debug agent is set.`*

I changed the subject because these configuration parameters are needed to be different as they are used differently.

**#9 - 09/26/2023 07:32 PM - Sergey Ivanovskiy**

*- Status changed from `WIP` to `Review`  
- % Done changed from 0 to 100*

Committed rev 14754 (7824a). This version should simplify debugging.

**#11 - 12/07/2023 12:25 PM - Sergey Ivanovskiy**

Could this task be reviewed? I rebased 7824a.

**#12 - 12/07/2023 03:01 PM - Greg Shah**

Is there any documentation about the connection settings (the full set, not just the ones changed here)?

Hynek: Please review.

**#13 - 12/08/2023 02:13 AM - Sergey Ivanovskiy**

Hynek, please review rev 14867 (7824a) that removed `MAX_HTTP_IDLE_TIMEOUT` and added `CONNECTION_TIMEOUT` and `INFINITE_TIMEOUT`. Now I am collecting these settings in one document to post here and to add to the documentation after the review.

**#14 - 12/08/2023 02:16 AM - Sergey Ivanovskiy**

*- File `7824a.patch` added*

This is what was changed in this task in one patch.

## #15 - 12/08/2023 02:46 AM - Sergey Ivanovskiy

These web client settings (under webClient node)

```
"webSocketTimeout" = WebClientDebugConfig.MAX_IDLE_TIME
"watchdogTimeout" = WebClientDebugConfig.WATCH_DOG_TIMEOUT
"maxIdleTime" = WebClientDebugConfig.MAX_IDLE_TIME
"maxHttpIdleTimeout" = WebClientDebugConfig.INFINITE_TIMEOUT
"delayBetweenTriesToConnect" = WebClientDebugConfig.DELAY_BETWEEN_CONNECT_TRIES
"pingPongInterval" = WebClientDebugConfig.PING_PONG_INTERVAL
"maxLostPings" = WebClientDebugConfig.MAX_LOST_PINGS
"delayBetweenPingTries" = WebClientDebugConfig.DELAY_BETWEEN_PING_TRIES
"enableDebugLogging" = WebClientDebugConfig.ENABLE_DEBUG_LOGGING
```

are set automatically for the debug mode. The debug mode is detecting automatically if clientConfig/jvmArgs has debug settings as in this example

```
<node class="string" name="jvmArgs">
  <node-attribute name="value" value="-Xmx1024m -Djava.awt.headless=true -Djava.compiler=NONE -Xdebu
g -Xrunjdw:transport=dt_socket,server=y,suspend=n,address=9988"/>
</node>
```

These settings are default

```
"webServerMaxIdleTimeout" = HttpConfigurationConstants.CONNECTION_TIMEOUT
"embWebServerMaxIdleTimeout" = HttpConfigurationConstants.INFINITE_TIMEOUT
```

but for the debug mode it needs to change settings for "webServerMaxIdleTimeout" manually

```
"webServerMaxIdleTimeout" = HttpConfigurationConstants.INFINITE_TIMEOUT
```

because it could affect the debug mode and these setting is set at the server start time.  
I do not know how to improve this part of settings.

**#16 - 12/08/2023 05:04 AM - Hynek Cihlar**

Code review 7824a revisions 14862..14867.

I may be misinterpreting the changes, but I don't see how the debug mode is enabled when the client is directly given the `-Xrunjdpw jvmArgs` option in the directory. `applyDebugProfile` seems to be called only when the debug port is assigned implicitly by `createResourceQueue` from the range `minAgentPort,maxAgentPort`.

**#17 - 12/08/2023 05:05 AM - Hynek Cihlar**

Hynek Cihlar wrote:

Code review 7824a revisions 14862..14867.

I may be misinterpreting the changes, but I don't see how the debug mode is enabled when the client is directly given the `-Xrunjdpw jvmArgs` option in the directory. `applyDebugProfile` seems to be called only when the debug port is assigned implicitly by `createResourceQueue` from the range `minAgentPort,maxAgentPort`.

Also `applyDebugProfile` should use the added string constants.

**#18 - 12/08/2023 06:31 AM - Sergey Ivanovskiy**

Hynek Cihlar wrote:

Code review 7824a revisions 14862..14867.

I may be misinterpreting the changes, but I don't see how the debug mode is enabled when the client is directly given the `-Xrunjdpw jvmArgs` option in the directory. `applyDebugProfile` seems to be called only when the debug port is assigned implicitly by `createResourceQueue` from the range `minAgentPort,maxAgentPort`

The agent port is assigned by `WebClientManager` that reads `jvmArgs` option in the directory. But the agent range must be configured too.

**#19 - 12/08/2023 06:48 AM - Hynek Cihlar**

Sergey Ivanovskiy wrote:

The agent port is assigned by `WebClientManager` that reads `jvmArgs` option in the directory. But the agent range must be configured too.

So when the range is not configured, will the debug mode be enabled?

**#20 - 12/08/2023 07:06 AM - Sergey Ivanovskiy**

At this moment the debug mode is not enabled if the agent range is not configured. For the known application there are at least two clients spawned and if the `jvmArgs` setting were used directly, then it would block the web client.

**#21 - 12/08/2023 07:10 AM - Sergey Ivanovskiy**

The agent range is common for the debugger and `jmx` agents so `jvmArgs` just describes `jmx` and/or debug agents should be started with the web client.

**#22 - 12/08/2023 07:22 AM - Sergey Ivanovskiy**

We can change the code of `WebClientsManager` for the case when agent range is not configured but the `jvmArgs` is given so it builds the port range from the provided single port. I would do this change in order to make settings consistent if there are no objections.

**#23 - 12/08/2023 08:43 AM - Greg Shah**

Sergey Ivanovskiy wrote:

These web client settings (under `webClient` node)

[...]

are set automatically for the debug mode. The debug mode is detecting automatically if `clientConfig/jvmArgs` has debug settings as in this example

[...]

These settings are default

[...]

but for the debug mode it needs to change settings for `"webServerMaxIdleTimeout"` manually

[...]

because it could affect the debug mode and these setting is set at the server start time.

I do not know how to improve this part of settings.

Please write this up in [Web Clients](#) which is a part of the [Directory Configuration Reference](#). I expect this is not just about debugging. It should document all the connection settings for web clients, including the content discussed in #7149 and #7981.

**#24 - 12/08/2023 09:47 AM - Hynek Cihlar**

Sergey Ivanovskiy wrote:

At this moment the debug mode is not enabled if the agent range is not configured. For the known application there are at least two clients spawned and if the `jvmArgs` setting were used directly, then it would block the web client.

There may be several ways how to provide the Java debug options for the spawned clients. I usually provide the debug options in `jvmArgs` in the directory. I then take responsibility for making sure the network ports don't conflict. So I think it makes sense to enable the debug mode for whatever client that was started with Java debugger for the sake of the increased flexibility.

**#25 - 12/08/2023 09:49 AM - Hynek Cihlar**

Sergey Ivanovskiy wrote:

We can change the code of WebClientsManager for the case when agent range is not configured but the jvmArgs is given so it builds the port range from the provided single port. I would do this change in order to make settings consistent if there are no objections.

Yes, this would benefit all the other deployment/configuration cases.

**#26 - 12/11/2023 06:34 AM - Sergey Ivanovskiy**

Hynek Cihlar wrote:

Sergey Ivanovskiy wrote:

We can change the code of WebClientsManager for the case when agent range is not configured but the jvmArgs is given so it builds the port range from the provided single port. I would do this change in order to make settings consistent if there are no objections.

Yes, this would benefit all the other deployment/configuration cases.

Please review the committed revision 14868 (7824a).

**#27 - 12/11/2023 11:30 AM - Hynek Cihlar**

Sergey Ivanovskiy wrote:

Please review the committed revision 14868 (7824a).

Code review.

According to the javaoc parseJMXPort (and parseDebuggerPort) should return -1 if the port is not found.

In parseJMXPort (and parseDebuggerPort), instead of getting a fixed sub string it will be safer to use matching group of the added regular expression. Just make the decimal a group (\\d+) and retrieve it with the matcher, @String port = matcher.group(1).

Otherwise the changes look good.

**#28 - 12/18/2023 11:40 AM - Sergey Ivanovskiy**

Please review rev 14889 (7824a) that fixed refs: [#7824-17](#) and [#7824-27](#).

**#29 - 12/21/2023 04:00 AM - Sergey Ivanovskiy**

7824a has been rebased. It seems that it can be merged into the trunc. Are there any objections?

**#30 - 12/21/2023 09:46 AM - Hynek Cihlar**

- Status changed from Review to Merge Pending

Code review 7824a revision 14897. The changes are good, I have no more objections.

**#31 - 12/21/2023 10:00 AM - Greg Shah**

This can merge after 7889a.

**#32 - 12/21/2023 11:21 AM - Sergey Ivanovskiy**

Planning to merge within next 15 minutes after rebase.

**#33 - 12/21/2023 11:33 AM - Sergey Ivanovskiy**

7824a has been rebased, merged into the trunc as rev 14893 and archived.

**#34 - 12/21/2023 11:36 AM - Sergey Ivanovskiy**

- Status changed from Merge Pending to Test

**#35 - 02/08/2024 01:32 AM - Sergey Ivanovskiy**

- % Done changed from 100 to 90

- Status changed from Test to WIP

According to #7143-673 it needs to support this debugger option suspend=y. Created 7824b.

**#37 - 02/08/2024 07:00 AM - Sergey Ivanovskiy**

- % Done changed from 90 to 100

- Status changed from WIP to Review

Please review the committed rev 14976 (7824b).

**#38 - 02/08/2024 07:02 AM - Sergey Ivanovskiy**

- File 7824b\_14976.patch added

Added the corresponding patch for the trunc source.

**#39 - 02/08/2024 08:53 AM - Roger Borrello**

Sergey Ivanovskiy wrote:

Added the corresponding patch for the trunc source.

Thanks, Sergey. The patch looks good, I'll apply it so I can test.

#### #40 - 02/08/2024 05:50 PM - Roger Borrello

Sergey, I applied the patch, and it doesn't work properly. It actually has suspend as null when it tries to build the args. I tried initializing suspendOption to "n" and that works for allowing the spawner to work, but it doesn't honor the value in the directory.xml.

#### #41 - 02/08/2024 05:58 PM - Roger Borrello

Roger Borrello wrote:

Sergey, I applied the patch, and it doesn't work properly. It actually has suspend as null when it tries to build the args. I tried initializing suspendOption to "n" and that works for allowing the spawner to work, but it doesn't honor the value in the directory.xml.

Is an addition to WebClientManger.java required, as well? I see the javaDebuggerSinglePortNumber is handled there:

```
javaDebuggerSinglePortNumber = ClientBuilderOptions.parseDebuggerPort(jvmArgs);
```

#### #42 - 02/09/2024 06:00 AM - Sergey Ivanovskiy

This pattern should work. I tested

```
Pattern p2 = Pattern.compile("suspend=([ny])");
Matcher m2 = p2.matcher("-Duser.country=GB -Duser.language=en -Xmx512m -XX:MaxPermSize=64m -Djava.awt.headless=true -Xdebug -Xrunjdp:transport=dt_socket,server=y,suspend=y,address=9988,ttt");
if (m2.find())
{
    if (m2.group() != null)
    {
        System.out.println("0->" + m2.group());
    }
    if (m2.group(1) != null)
    {
        System.out.println("1->" + m2.group(1));
    }
}
```

The output should be

```
0->suspend=y
1->y
```



#### #43 - 02/09/2024 09:10 AM - Roger Borrello

Yes, the pattern works, but I had to add this to allow the server to start:

```
+  
+ /** Storage for debugger suspend option */  
+ private String suspendOption = "n";  
+
```

Because the value was "null" when substituted in. I protected that case with below, which probably isn't necessary after performing the initialization of suspendOption:

```
public void updatePortForJavaDebuggerAgent(int debuggerPort)  
{  
-     if (debuggerPort > 0)  
+     if ((debuggerPort > 0) && (suspendOption != null))
```

Still the "suspend=y" was not honored because the initialize method is never called prior to the spawn. It looks like it should be called somewhere before spawning, perhaps ClientSpawner.updateWebClientOptions? It was never changed from suspendOption = "n".

#### #44 - 02/09/2024 09:10 AM - Greg Shah

Sergey is busy with other tasks, so you will have to implement the fixes and Sergey can review your changes.

#### #45 - 02/09/2024 05:50 PM - Roger Borrello

Greg Shah wrote:

Sergey is busy with other tasks, so you will have to implement the fixes and Sergey can review your changes.

Updates are included in r14980, and ready for review.

#### #46 - 02/13/2024 08:02 AM - Sergey Ivanovskiy

Roger, your code changes helped me to find out what was missed. I think that WebClientCnfig should not have this suspend option as its field because suspend option is shared between all instances. It is enough to have

```

protected ClientBuilderOptions(ClientBuilderOptions copy, String[] accountIds)
{
    this.accountIds = accountIds;

    this.secure           = copy.secure;
    this.command          = copy.command;
    this.classPath        = copy.classPath;
    this.libPath          = copy.libPath;
    this.spawner          = copy.spawner;
    this.spawnerDebugEnabled = copy.spawnerDebugEnabled;
    this.workingDirectory = copy.workingDirectory;
    this.configFile       = copy.configFile;
    this.jvmArguments     = new LinkedList<String>(copy.jvmArguments);
    this.bootstrapConfig  = new LinkedList<String>(copy.bootstrapConfig);
    this.debuggerSuspendOption = copy.debuggerSuspendOption;
}

```

so this.debuggerSuspendOption is filled from WebClientBuilderOptions.cbo static field.

#### #47 - 02/13/2024 12:37 PM - Roger Borrello

Sergey Ivanovskiy wrote:

Roger, your code changes helped me to find out what was missed. I think that WebClientCnfig should not have this suspend option as its field because suspend option is shared between all instances. It is enough to have

[...]

so this.debuggerSuspendOption is filled from WebClientBuilderOptions.cbo static field.

Is that the same for debugAgentPort? I wanted to just follow how that was implements, since they work together for jvmArgs.

#### #48 - 02/14/2024 03:17 AM - Sergey Ivanovskiy

No, the debug port and the suspend option should be implemented differently. If the port range is not given, then the debug port is defined by the directory, otherwise it is defined by the port range

```

<node class="integer" name="minAgentPort">
  <node-attribute name="value" value="9988"/>

```

```
</node>
<node class="integer" name="maxAgentPort">
  <node-attribute name="value" value="9999"/>
</node>
```

Then each new client uses its own debug port from this range but the suspend option is the same for all debugged clients.

WebClientBuilderOptions.cbo is initialized when WebClientBuilderOptions.initialize is called at the server startup time. In my code this part was only missed

```
protected ClientBuilderOptions(ClientBuilderOptions copy, String[] accountIds)
{
  .....
  this.debuggerSuspendOption = copy.debuggerSuspendOption; // this line is only needed to add after renami
ng the suspend option
}
```

**#49 - 02/14/2024 08:56 AM - Roger Borrello**

Thank you for clarifying. This is ready for review as revision 14989. Branch has been rebased to trunk revision 14985.

**#50 - 02/14/2024 09:34 AM - Sergey Ivanovskiy**

I found minor issue that you missed javadoc for this field

```
=== modified file 'src/com/goldencode/p2j/main/ClientBuilderOptions.java'
.....
- /** Storage for debugger suspend option */
- private String debugSuspendOption;
-
+ private String suspendOption;
```

The changes are good and debugSuspendOption name can be here according to your taste.

**#51 - 02/14/2024 09:56 AM - Roger Borrello**

Oops... fixed and committed in revision 14990. I believe this is ready for merge to trunk.

**#52 - 02/27/2024 10:17 AM - Roger Borrello**

Branch 7824b was rebased to trunk revision 15011 and is now at revision 15017.

I believe this is ready for merge.

**#53 - 02/27/2024 10:21 AM - Sergey Ivanovskiy**

Yes, I agree with Roger.

**#54 - 02/27/2024 10:37 AM - Greg Shah**

Is there any risk associated with these changes?

**#55 - 02/27/2024 10:54 AM - Roger Borrello**

Greg Shah wrote:

Is there any risk associated with these changes?

No. They would only be used in debug scenarios.

Rebased to trunk revision 15012 and is at revision 15018.

**#56 - 02/27/2024 10:56 AM - Greg Shah**

You've tested non-debug scenarios to confirm that what "should" happen actually is the case?

**#57 - 02/27/2024 11:00 AM - Roger Borrello**

Greg Shah wrote:

You've tested non-debug scenarios to confirm that what "should" happen actually is the case?

Yes... I've been using this update in all my testing (debug and non-debug) for a couple of weeks now.

**#58 - 02/27/2024 11:01 AM - Greg Shah**

- Status changed from Review to Merge Pending

You can merge to trunk.

**#59 - 02/27/2024 12:24 PM - Roger Borrello**

Branch 7824b has been merged to trunk, and archived.

**#60 - 02/27/2024 01:07 PM - Greg Shah**

- Status changed from Merge Pending to Closed

**Files**

---

7824a.patch	25.1 KB	12/08/2023	Sergey Ivanovskiy
7824b_14976.patch	3.11 KB	02/08/2024	Sergey Ivanovskiy