Base Language - Feature #8138

ABLUnit configuration using environment variables

12/20/2023 04:08 AM - Marian Edu

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**vendor_id:**	GCD

Description

History

#1 - 12/20/2023 04:13 AM - Marian Edu

I'm extracting that from another unrelated issue just so it doesn't get lost.

Vladimir, we're running ABLUnit from Eclipse and you're right as it doesn't seems to be any option for pass any configuration parameters when the engine is initialized... what you could do maybe is to allow usage of an environment variable (fwd_config_file) and use that instead of the default if set in FWDTestEngine.

```
String configFile = configurationParameters.get(CFG_FWD_CONFIG_FILE).orElse(System.getenv(CFG_FWD_CONFIG_FILE)
);

if (initialize(configFile != null ? configFile : DEFAULT_FWD_CONFIG_FILE)) {
```

The reason behind this is the use of relative paths in PROPATH makes the execution dependent on the current working directory hence running it from client_gui/client_chui so it can pick-up the right abl_unit.xml file is causing issues on finding resources using relative path.

Thanks

#3 - 12/20/2023 06:02 AM - Vladimir Tsichevski

IMO this will add another layer of complexity and mixed technology. Besides, using OS environment variables with Java is not recommended.

I would prefer if the FWD client configuration was implemented with Java properties instead of command-line arguments, which **can** be used with Eclipse.

The command line may look like:

```
-Dnet.server.host=localhost -Dnet.connection.secure=false -Dnet.server.secure_port=4434 -Dnet.server.insecure_
port=4534 -Dclient.driver.type=gui_swing -Dclient.driver.theme=com.goldencode.p2j.ui.client.gui.theme.Windows1
0Theme
```

And standard Java property files can be used in place of XML files.

#4 - 12/20/2023 06:49 AM - Marian Edu

Vladimir Tsichevski wrote:

> IMO this will add another layer of complexity and mixed technology. Besides, using OS environment variables with Java is not recommended.
>
> I would prefer if the FWD client configuration was implemented with Java properties instead of command-line arguments, which **can** be used with Eclipse.
>
> The command line may look like:
>
> [...]
>
> And standard Java property files can be used in place of XML files.

I really have no preference here, whatever you think is appropriate, just not force the location of unit_test.xml file to be in current folder :)

We can certainly send command line arguments to the JUnit run-configuration, only I don't know exactly where you can extract those so environment variables was the easiest way for me, recommended or not those are there and can certainly be used :)

#5 - 12/20/2023 06:53 AM - Vladimir Tsichevski

Marian Edu wrote:

> Vladimir Tsichevski wrote:
>
> > IMO this will add another layer of complexity and mixed technology. Besides, using OS environment variables with Java is not recommended.
> >
> > I would prefer if the FWD client configuration was implemented with Java properties instead of command-line arguments, which **can** be used with Eclipse.
> >
> > The command line may look like:
> >
> > [...]
> >
> > And standard Java property files can be used in place of XML files.
>
> I really have no preference here, whatever you think is appropriate, just not force the location of unit_test.xml file to be in current folder :)
>
> We can certainly send command line arguments to the JUnit run-configuration, only I don't know exactly where you can extract those so environment variables was the easiest way for me, recommended or not those are there and can certainly be used :)

Using Java properties for client configuration will allow to run with no configuration files, so we will not have to bother about the client process working directory.

#6 - 12/20/2023 08:27 AM - Marian Edu

Vladimir Tsichevski wrote:

> Using Java properties for client configuration will allow to run with no configuration files, so we will not have to bother about the client process working directory.

I would rather not send all client properties that way, and I expect the use of XML configuration instead of plain properties files has something to do with the hierarchical structure that is needed - maybe not for the client though, I do not know, but any way it looks like it's just reusing the same bootstrap mechanism so really the easiest way for us at this point is to make one option to set the unit_test.xml (actually client configuration) when running from Eclipse. If not through environment variables I'm ok with a command argument but is not my call to make a design decision here :)

For time being we will just run with our local tweak based on environment variables and will switch to whatever needed when something else is available.

Thanks

#7 - 12/20/2023 01:21 PM - Greg Shah

I agree, we should add the ability to configure the bootstrap cfg file name (and only the bootstrap cfg file name) via Java property.

I don't want to add any other property support, as this would duplicate what we already have for the bootstrap cfg.