Base Language - Bug #8116

SEARCH vs FILE-INFO

12/12/2023 04:19 AM - Marian Edu

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

Description

History

#1 - 12/12/2023 04:19 AM - Marian Edu

When using SEARCH using relative path in 4GL the file is looked-up in all PROPATH entries and if found the full-path is returned, pretty much the same functionality as when you set the FILE-NAME on FILE-INFO system handle and check for FULL-PATHNAME. In FWD the initial relative path is returned instead by SEARCH, this is true for both code and regular files.

#2 - 12/12/2023 08:16 AM - Marian Edu

Although there is this slight difference in FWD implementation my issue seems to come from invalid PROPATH settings in testcases project. The PROPATH there should be .,appsrv/api but when I want to display PROPATH in FWD client it only shows ., I see two entries for propath in deploy/server/directory.xml file (one for appsrv and one for runtime) and one in cfg/p2j.cfg.xml. The separator there seems to always be :, this seems to match the path-separator value but those entries doesn't look like being used anymore... the searchpath one is what actually sets the value. Even so, updating the searchpath to be .:apprsv/api: doesn't seem to change anything - searching for relative files in appsrv/api still doesn't find anything :(

I'm definitively missing something obviously here...

Anyway, using paths relative to the current working folder is causing issues when running the ABLUnit engine, sourcemapper helps in finding code to be executed like windows that we want to run, but other files like sikuli executable or script files can't be found when we start from deploy/client instead of the project root.

Is this a known issue, it is supposed to work but the setup is not correct, anything we can do to make this work?

#3 - 12/13/2023 08:51 AM - Greg Shah

- Assignee set to Eugenie Lyzenko

I'll make a couple of quick points and then Eugenie will take it from there.

1. Your use case should be supported already. I think this is just a cfg issue.

2. The PROPATH for conversion is set in p2j.cfg.xml (and optionally in .hints files). If I recall correctly, the path separator must match the legacy path separator in this case.

3. The PROPATH for runtime comes from the directory, from one of these locations (in this order of searching):

- /server/<serverID>/runtime/<account_or_group>/searchpath
- /server/<serverID>/runtime/default/searchpath
- /server/default/runtime/<account_or_group>/searchpath
- /server/default/runtime/default/searchpath

The code that reads it is the initialization code for EnvironmentOps.WorkArea which sets the propathOverride value. Then any call to EnvironmentOps.getSearchPath() uses this value to build the propath. The processing of search and file-info should take this into account and as you note, we have a complex multi-layer lookup that differentiates program names, jar-file resources and eventually falls back to searching the client side file system.

#4 - 12/13/2023 08:53 AM - Greg Shah

Vladimir can also help here regarding the specifics of our ABLUnit support. It is slightly different because of the way it is launched but the core 4GL support should still be there. Are you running the console launcher or via Eclipse? The capabilities are slightly different due to limitations in JUnit 5.

#5 - 12/13/2023 08:58 AM - Eugenie Lyzenko

From application point of view the PROPATH/SEARACHPATH should be , separated. But in a real application the separator can be OS dependent and is fixed at runtime according to path-separator value defined in config file or the default value for underlying OS at the time of the execution.

#6 - 12/13/2023 09:03 AM - Vladimir Tsichevski

Greg Shah wrote:

> Vladimir can also help here regarding the specifics of our ABLUnit support. It is slightly different because of the way it is launched but the core 4GL support should still be there. Are you running the console launcher or via Eclipse? The capabilities are slightly different due to limitations in JUnit 5.

The only difference I can recall is that the client configuration cannot be set from command line, so the only option is client config XML file.
The name of the file can be set if you are running unit tests with console launcher, and cannot be set from Eclipse.

#7 - 12/14/2023 06:46 AM - Marian Edu

Eugenie Lyzenko wrote:

> From application point of view the PROPATH/SEARACHPATH should be , separated. But in a real application the separator can be OS dependent and is fixed at runtime according to path-separator value defined in config file or the default value for underlying OS at the time of the execution.

OK, still a bit confused about configuration node names here but using searchpath along with path-separator seems to render the expected result. I guess the directory.xml template in testcases project is using propath instead of searchpath, propath seems to simply be ignored. Not sure how this works for application server (standard/appservers/app_server) where searchpath and path-separator seems not to be valid configuration options and propath is set with : as separator... luckily we're not using app_server in FWD just yet ;)

The issue with relative vs. absolute path reported by search remains however, just in case this causes any other issues.

#8 - 12/14/2023 07:13 AM - Marian Edu

Vladimir Tsichevski wrote:

> The only difference I can recall is that the client configuration cannot be set from command line, so the only option is client config XML file. The name of the file can be set if you are running unit tests with console launcher, and cannot be set from Eclipse.

Thanks for that Vladimir, we're indeed running it from Eclipse and 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)) {
```

#9 - 12/14/2023 09:29 AM - Greg Shah

I like the idea.