

Database - Feature #4514

run ProDataSet test suite in FWD and resolve any issues such that it works the same way as in the 4GL

01/22/2020 05:40 PM - Greg Shah

| | |
|--|----------------------------------|
| Status: WIP | Start date: |
| Priority: Normal | Due date: |
| Assignee: Ovidiu Maxiniuc | % Done: 10% |
| Category: | Estimated time: 0.00 hour |
| Target version: | vendor_id: GCD |
| billable: No | |
| Description | |
| Related issues: | |
| Related to Database - Feature #3809: ProDataSet support | Closed |
| Related to Database - Bug #6475: double-check and fix any read/write-xml/json... | New |

History

#1 - 01/22/2020 05:42 PM - Greg Shah

As part of the work to add ProDataSet support in [#3809](#), a test suite was written to explore the behavior of the PDS feature set, error handling, transaction handling, parameter passing and so forth.

This task is meant to convert and run that test suite in FWD, resolving any issues so that FWD is fully compatible with the 4GL (at least in regard to this set of tests).

#2 - 01/22/2020 05:42 PM - Greg Shah

- Related to Feature #3809: ProDataSet support added

#3 - 04/14/2023 07:18 AM - Greg Shah

- Assignee set to Maciej Zieniuk

#4 - 04/14/2023 07:21 AM - Greg Shah

The tests are in [Testcases](#) under the tests/dataset/ directory (for the ones already migrated to ABLUnit) and under dataset/ for some tests that are not yet migrated.

Marian: Is there any advice needed on how to run these tests?

#5 - 04/17/2023 10:31 AM - Maciej Zieniuk

- Status changed from New to WIP

- Start date set to 04/17/2023

I have issues with conversion on tests/dataset/ and dataset/ - latest trunk on testcases and p2j repos.

The file-cvt-list.txt content:

```
tests/dataset/parameter/inout/TestInputOutputDynamicDsInDynamicProc.cls
```

I have the skeleton/ repo checked out, p2j latest trunk linked as p2j directory, and .propath file.

I can't seem to locate what is the issue with the test.AssertExt, any idea ?

The error:

```
[java] Code Conversion Annotations
[java] -----
[java]
[java] Optional rule set [customer_specific_annotations_prep] not found.
[java] ./tests/dataset/parameter/inout/TestInputOutputDynamicDsInDynamicProc.cls
[java] WARNING: Null annotation (full-java-class) for test.AssertExt [CLASS_NAME] @60:9 (107374182928)
[java] WARNING: Null annotation (simple-java-class) for test.AssertExt [CLASS_NAME] @60:9 (107374182928)
[java] WARNING: Null annotation (containing-package) for test.AssertExt [CLASS_NAME] @60:9 (107374182928)
[java] WARNING: Null annotation (found-in-full-java-class) for NotErrorNotWarning [OO_METH_VOID] @60:24 (
107374182929)
[java] WARNING: Null annotation (full-java-class) for test.AssertExt [CLASS_NAME] @75:9 (107374183087)
[java] WARNING: Null annotation (simple-java-class) for test.AssertExt [CLASS_NAME] @75:9 (107374183087)
[java] WARNING: Null annotation (containing-package) for test.AssertExt [CLASS_NAME] @75:9 (107374183087)
[java] WARNING: Null annotation (found-in-full-java-class) for NotErrorNotWarning [OO_METH_VOID] @75:24 (
107374183088)
[java] WARNING: Null annotation (full-java-class) for test.AssertExt [CLASS_NAME] @92:9 (107374183240)
[java] WARNING: Null annotation (simple-java-class) for test.AssertExt [CLASS_NAME] @92:9 (107374183240)
[java] WARNING: Null annotation (containing-package) for test.AssertExt [CLASS_NAME] @92:9 (107374183240)
[java] WARNING: Null annotation (found-in-full-java-class) for NotErrorNotWarning [OO_METH_VOID] @92:24 (
107374183241)
[java] WARNING: Null annotation (full-java-class) for test.AssertExt [CLASS_NAME] @105:9 (107374183392)
[java] WARNING: Null annotation (simple-java-class) for test.AssertExt [CLASS_NAME] @105:9 (107374183392)
[java] WARNING: Null annotation (containing-package) for test.AssertExt [CLASS_NAME] @105:9 (107374183392
)
[java] WARNING: Null annotation (found-in-full-java-class) for NotErrorNotWarning [OO_METH_VOID] @105:24
(107374183393)
[java] WARNING: Null annotation (full-java-class) for test.AssertExt [CLASS_NAME] @118:9 (107374183526)
[java] WARNING: Null annotation (simple-java-class) for test.AssertExt [CLASS_NAME] @118:9 (107374183526)
[java] WARNING: Null annotation (containing-package) for test.AssertExt [CLASS_NAME] @118:9 (107374183526
)
[java] WARNING: Null annotation (found-in-full-java-class) for NotErrorNotWarning [OO_METH_VOID] @118:24
(107374183527)
[java] WARNING: Null annotation (full-java-class) for test.AssertExt [CLASS_NAME] @126:9 (107374183604)
[java] WARNING: Null annotation (simple-java-class) for test.AssertExt [CLASS_NAME] @126:9 (107374183604)
[java] WARNING: Null annotation (containing-package) for test.AssertExt [CLASS_NAME] @126:9 (107374183604
)
[java] WARNING: Null annotation (found-in-full-java-class) for NotErrorNotWarning [OO_METH_VOID] @126:24
(107374183605)
[java] WARNING: Null annotation (full-java-class) for test.AssertExt [CLASS_NAME] @138:9 (107374183745)
[java] WARNING: Null annotation (simple-java-class) for test.AssertExt [CLASS_NAME] @138:9 (107374183745)
[java] WARNING: Null annotation (containing-package) for test.AssertExt [CLASS_NAME] @138:9 (107374183745
)
[java] WARNING: Null annotation (found-in-full-java-class) for NotErrorNotWarning [OO_METH_VOID] @138:24
(107374183746)
[java] WARNING: Null annotation (full-java-class) for test.AssertExt [CLASS_NAME] @149:9 (107374183841)
[java] WARNING: Null annotation (simple-java-class) for test.AssertExt [CLASS_NAME] @149:9 (107374183841)
[java] WARNING: Null annotation (containing-package) for test.AssertExt [CLASS_NAME] @149:9 (107374183841
)
[java] WARNING: Null annotation (found-in-full-java-class) for Error [OO_METH_VOID] @149:24 (107374183842
)
[java] WARNING: Null annotation (full-java-class) for test.AssertExt [CLASS_NAME] @162:9 (107374183985)
[java] WARNING: Null annotation (simple-java-class) for test.AssertExt [CLASS_NAME] @162:9 (107374183985)
[java] WARNING: Null annotation (containing-package) for test.AssertExt [CLASS_NAME] @162:9 (107374183985
)
[java] WARNING: Null annotation (found-in-full-java-class) for NotErrorNotWarning [OO_METH_VOID] @162:24
(107374183986)
[java] WARNING: Null annotation (full-java-class) for test.AssertExt [CLASS_NAME] @176:9 (107374184128)
[java] WARNING: Null annotation (simple-java-class) for test.AssertExt [CLASS_NAME] @176:9 (107374184128)
[java] WARNING: Null annotation (containing-package) for test.AssertExt [CLASS_NAME] @176:9 (107374184128
)
[java] WARNING: Null annotation (found-in-full-java-class) for Error [OO_METH_VOID] @176:24 (107374184129
)
[java] WARNING: Null annotation (full-java-class) for test.AssertExt [CLASS_NAME] @180:9 (107374184190)
[java] WARNING: Null annotation (simple-java-class) for test.AssertExt [CLASS_NAME] @180:9 (107374184190)
[java] WARNING: Null annotation (containing-package) for test.AssertExt [CLASS_NAME] @180:9 (107374184190
)
[java] WARNING: Null annotation (found-in-full-java-class) for Error [OO_METH_VOID] @180:24 (107374184191
)
[java] Elapsed job time: 00:00:01.296
[java] EXPRESSION EXECUTION ERROR:
```

```

[java] -----
[java] persist()
[java] ^ { null value for annotation 'full-java-class':test.AssertExt [CLASS_NAME]:107374182928 @60:9
[java] }
[java] -----
[java] ERROR:
[java] com.goldencode.p2j.pattern.TreeWalkException: ERROR! Active Rule:
[java] -----
[java]          RULE REPORT
[java] -----
[java] Rule Type :    POST
[java] Source AST:  [ block ] BLOCK/ @0:0 {107374182401}
[java] Copy AST   :  [ block ] BLOCK/ @0:0 {107374182401}
[java] Condition  :  persist()
[java] Loop       :  false
[java] --- END RULE REPORT ---
[java]
[java]
[java]
[java] at com.goldencode.p2j.pattern.PatternEngine.run(PatternEngine.java:1093)
[java] at com.goldencode.p2j.convert.TransformDriver.processTrees(TransformDriver.java:586)
[java] at com.goldencode.p2j.convert.ConversionDriver.back(ConversionDriver.java:563)
[java] at com.goldencode.p2j.convert.TransformDriver.executeJob(TransformDriver.java:998)
[java] at com.goldencode.p2j.convert.ConversionDriver.main(ConversionDriver.java:1284)
[java] Caused by: com.goldencode.expr.ExpressionException: Expression execution error @1:1
[java] at com.goldencode.expr.Expression.execute(Expression.java:484)
[java] at com.goldencode.p2j.pattern.Rule.apply(Rule.java:500)
[java] at com.goldencode.p2j.pattern.RuleContainer.apply(RuleContainer.java:590)
[java] at com.goldencode.p2j.pattern.RuleSet.apply(RuleSet.java:98)
[java] at com.goldencode.p2j.pattern.PatternEngine.apply(PatternEngine.java:1704)
[java] at com.goldencode.p2j.pattern.PatternEngine.processAst(PatternEngine.java:1571)
[java] at com.goldencode.p2j.pattern.PatternEngine.processAst(PatternEngine.java:1504)
[java] at com.goldencode.p2j.pattern.PatternEngine.run(PatternEngine.java:1056)
[java] ... 4 more
[java] Caused by: java.lang.NullPointerException: null value for annotation 'full-java-class':test.Assert
Ext [CLASS_NAME]:107374182928 @60:9
[java]
[java] at com.goldencode.ast.XmlFilePlugin.writeSingleAnnotation(XmlFilePlugin.java:1154)
[java] at com.goldencode.ast.XmlFilePlugin.writeAnnotations(XmlFilePlugin.java:1115)
[java] at com.goldencode.ast.XmlFilePlugin.writeAst(XmlFilePlugin.java:1074)
[java] at com.goldencode.ast.XmlFilePlugin.writeAst(XmlFilePlugin.java:1080)
[java] at com.goldencode.ast.XmlFilePlugin.writeAst(XmlFilePlugin.java:1080)
[java] at com.goldencode.ast.XmlFilePlugin.writeAst(XmlFilePlugin.java:1080)
[java] at com.goldencode.ast.XmlFilePlugin.writeAst(XmlFilePlugin.java:1080)
[java] at com.goldencode.ast.XmlFilePlugin.writeAst(XmlFilePlugin.java:1080)
[java] at com.goldencode.ast.XmlFilePlugin.writeAst(XmlFilePlugin.java:1080)
[java] at com.goldencode.ast.XmlFilePlugin.writeAst(XmlFilePlugin.java:1080)
[java] at com.goldencode.ast.XmlFilePlugin.writeAst(XmlFilePlugin.java:1080)
[java] at com.goldencode.ast.XmlFilePlugin.writeAst(XmlFilePlugin.java:1080)
[java] at com.goldencode.ast.XmlFilePlugin.writeAst(XmlFilePlugin.java:1080)
[java] at com.goldencode.ast.XmlFilePlugin.writeAst(XmlFilePlugin.java:1080)
[java] at com.goldencode.ast.XmlFilePlugin.writeAst(XmlFilePlugin.java:1080)
[java] at com.goldencode.ast.XmlFilePlugin.saveTree(XmlFilePlugin.java:518)
[java] at com.goldencode.ast.AstManager.saveTree(AstManager.java:360)
[java] at com.goldencode.p2j.pattern.CommonAstSupport$Library.persist(CommonAstSupport.java:2968)
[java] at com.goldencode.expr.CE9357.execute(Unknown Source)
[java] at com.goldencode.expr.Expression.execute(Expression.java:391)
[java] ... 11 more

```

BUILD FAILED

#6 - 04/17/2023 10:43 AM - Greg Shah

At this time we have a limitation in our conversion approach, such that we must list the entire referenced object graph in the conversion list. We don't (yet) calculate the OO dependencies and automatically add them. See [#6082](#).

#7 - 04/18/2023 04:43 PM - Maciej Zieniuk

- % Done changed from 0 to 10

I have few tests passing, but there are few which will definitively not run.

As an example, the tests/dataset/TestAfterFill.cls have a dependency on the run dataset/events/handler/eventsLib.p, but such file no longer exists in support/ nor tests/. I was thinking about checking against the bzz log for the commits, # refs that correlates to the deletion of the eventsLib.p, but i am not sure what to expect yet - any idea what it was used for or how was it replaced with something else ?

#8 - 04/19/2023 05:12 AM - Marian Edu

Maciej Zieniuk wrote:

I have few tests passing, but there are few which will definitively not run.

As an example, the tests/dataset/TestAfterFill.cls have a dependency on the run dataset/events/handler/eventsLib.p, but such file no longer exists in support/ nor tests/. I was thinking about checking against the bzz log for the commits, # refs that correlates to the deletion of the eventsLib.p, but i am not sure what to expect yet - any idea what it was used for or how was it replaced with something else ?

That one was moved in a different location under support folder only forgot to push the changes in bazaar, will do the sync in a few moments.

#9 - 04/19/2023 06:27 AM - Marian Edu

Marian Edu wrote:

That one was moved in a different location under support folder only forgot to push the changes in bazaar, will do the sync in a few moments.

Should be fixed in rev #1378 of testcases.

#10 - 04/19/2023 03:27 PM - Maciej Zieniuk

Thanks Marian.

But I think i am doing something wrong.

The tests/dataset/TestXmlNodeType.cls have a line during the setup, which runs some other procedure run dataset/create/dsMaster.p. I experimented with different options, read the documentation and could not find anything where i could get it working. It's converting, but the tests during runtime immediately fails, that the dataset/create/dsMaster.p is nowhere to be found. The only way to fix this is to add the full path (support/) to the run run support/dataset/create/dsMaster.p.

Any ideas on this ?

On the other failures, there are plenty of tests that depends on the dataset/stdFiles or dataset/outFiles files, but they are nowhere to be found.

Example dataset/outFiles/dynamic/xml/memptr/dsMaster_dsMaster_element_no_no_UTF-8_default_no_no_no_no.xml in file tests/dataset/TestReadXmlFileNoPkReadGenerated.cls.

#11 - 04/20/2023 04:35 AM - Marian Edu

Maciej Zieniuk wrote:

The tests/dataset/TestXmlNodeType.cls have a line during the setup, which runs some other procedure run dataset/create/dsMaster.p. I experimented with different options, read the documentation and could not find anything where i could get it working. It's converting, but the tests during runtime immediately fails, that the dataset/create/dsMaster.p is nowhere to be found. The only way to fix this is to add the full path (support/) to the run run support/dataset/create/dsMaster.p.
Any ideas on this ?

Yes, for whatever reason we had 'support' and 'tests' folders added to the PROPATH and that is probably not the case for FWD. I think we should better fix that one right now so we only keep the project's ROOT in PROPATH, we will 'refactor' current tests for this and let you know when available on testcases.

On the other failures, there are plenty of tests that depends on the dataset/stdFiles or dataset/outFiles files, but they are nowhere to be found. Example dataset/outFiles/dynamic/xml/memptr/dsMaster_dsMaster_element_no_no_UTF-8_default_no_no_no_no.xml in file tests/dataset/TestReadXmlFileNoPkReadGenerated.cls.

Yes, those are files we use for comparing results - we normally generate them but it looks like the 'generators' were also missing, anyway I will add those back in new location tests\dataset.

#12 - 04/21/2023 06:47 AM - Maciej Zieniuk

The TestNamespaceUri.cls method TestSetInvalid is outdated and needs to be corrected.

Below procedure result in error in 4GL:

```
session:suppress-warnings = false.  
  
define temp-table ttCustomer no-undo  
    field name as character.  
  
define dataset dsMaster  
    for ttCustomer.  
  
dataset dsMaster:namespace-uri = ?.
```

but it works without errors in FWD code.

The converted procedure results in code `silent() -> dsMaster.namespaceURI(new character())`, where the inside of the `DataSet.java` is:

```
public void namespaceURI(character uri)
{
    namespaceURI(uri == null || uri.isUnknown() ? "" : uri.toJavaType());
}

@Override
public void namespaceURI(String uri)
{
    if (uri == null)
    {
        unableToAssignUnknown("NAMESPACE-URI", "DATASET widget");
        return;
    }

    nsUri = uri;
}
```

I am thinking about making a change to the FWD code, to something like:

```
public void namespaceURI(character uri)
{
    namespaceURI(uri == null || uri.isUnknown() ? null : uri.toJavaType());
}
```

Thoughts ?

#13 - 04/21/2023 07:23 AM - Marian Edu

I've pushed a bunch of changes in testcases, we took out tests, support from PROPATH and added missing 'standard files' we use for comparing

write-xml/json output and of course updated references for include files/procedures we call. I've also added a couple of TestSuites that you might want to try, however there is a known issue with the length of characters in test suite's classes (<https://community.progress.com/s/article/TestSuites-fail-with-a-large-number-characters-in-the-TestSuite-classes-string>) which for some cases prevents us to use only one test suite and having more than one kinda defeats the purpose of having an option to run all tests at once :(

Running all tests in a suite seems to make some of those tests to fail even if ran individually they pass... currently under investigation.

There are some tests that doesn't compile in 4GL, those were generated for attributes/methods that needs to be tested only we did not had any tests for those before so remains to be implemented but thought to keep those there for reference, we will exclude those from the test suites.

#14 - 04/21/2023 07:55 AM - Greg Shah

I am thinking about making a change to the FWD code, to something like:

That seems reasonable if the unableToAssignUnknown() replicates the 4GL behavior.

Ovidiu: Any objections?

#15 - 04/21/2023 08:00 AM - Marian Edu

Greg Shah wrote:

I am thinking about making a change to the FWD code, to something like:

That seems reasonable if the unableToAssignUnknown() replicates the 4GL behavior.

That seems to be the general behaviour of 4GL for any handle object that doesn't accept unknown, same as for read-only attributes - always same generic error.

Ovidiu: Any objections?

#16 - 04/21/2023 04:28 PM - Ovidiu Maxiniuc

Greg Shah wrote:

I am thinking about making a change to the FWD code, to something like:

That seems reasonable if the `unableToAssignUnknown()` replicates the 4GL behavior.
Ovidiu: Any objections?

No, no objections. That is the right thing to do (null is correct parameter there). Sorry I missed this in my review of r14483.1.238.

#17 - 04/22/2023 02:18 AM - Maciej Zieniuk

I have pushed that change to `~/secure/code/p2j_repo/p2j/active/4514a`. If i find anything i will push more to the branch, I will let you know.

#18 - 04/24/2023 05:47 AM - Greg Shah

- *Start date deleted (04/17/2023)*

I have pushed that change to `~/secure/code/p2j_repo/p2j/active/4514a`. If i find anything i will push more to the branch, I will let you know.

You used the word "push" so I just want to clarify something. Are you using `bzr push --overwrite` or are you using `bzr commit` in a "bound" branch? The latter is what you should be doing.

#19 - 04/25/2023 02:51 AM - Maciej Zieniuk

Yes by "push" i meant bound branch to a remote repo with `bzr commit`.

I will have to rebase too once in a while, to get the most recent FWD code, which i believe involves `bzr push --overwrite` anyway, no ?

#20 - 04/25/2023 07:41 AM - Greg Shah

Maciej Zieniuk wrote:

Yes by "push" i meant bound branch to a remote repo with `bzr commit`.

I will have to rebase too once in a while, to get the most recent FWD code, which i believe involves `bzr push --overwrite` anyway, no ?

Yes, it does. We also, in rare cases where someone has corrupted a task branch master, can use `bzr push --overwrite` to replace a branch with a fixed/clean version. I just want to ensure that it isn't used in place of a normal commit. Using push too much can break our normal dev process which relies upon `bzr commit/bzr update`.

#21 - 04/25/2023 10:45 AM - Maciej Zieniuk

Running all tests in a suite seems to make some of those tests to fail even if ran individually they pass... currently under investigation.

Marian, is this why after approx 100th test i get "Legacy error" failures ?

Even if they are not part of the test suite, so they run one by one (but in one JUnit vm run), they fail the same.

Either way it is not easy to get a good estimate on the progress when a test fails with a message, which involves an dialog box / alert, which halts the execution of other tests (you have to close the dialog) on the `gui_swing` client driver type. The only way where i could run it without interruptions is with `chui_batch` driver - which i doubt is intended. Any idea on how to overcome this issue ?

Another thing is that when a test fails due to `assert`, it does not exactly specify where it failed. If it would fail otherwise, i.e. due to NPE, the `stacktrace` is present which points to specific line in the generated `.java` class, which is easier to debug, as you can compare the 4GL test code with the generated code.

You could technically argue, that the test should only do one assertion, but i think we should have that information, at least the JVM stack which includes the `.java` source code line.

I also have a problem with this test `tests/dataset/parameter/input/TestDatasetAppend.cls`, where you can find `Assert.IsTrue(numCust = 3 and numItem = 5)`. assertions. On failure you just get `Expected: TRUE but was: no error`, not very helpful - i will split those into separate equality assertions, i think it make sense.

#22 - 04/25/2023 12:34 PM - Vladimir Tsichevski

Maciej Zieniuk wrote:

Running all tests in a suite seems to make some of those tests to fail even if ran individually they pass... currently under investigation.

Marian, is this why after approx 100th test i get "Legacy error" failures ?

Even if they are not part of the test suite, so they run one by one (but in one JUnit vm run), they fail the same.

Either way it is not easy to get a good estimate on the progress when a test fails with a message, which involves an dialog box / alert, which halts the execution of other tests (you have to close the dialog) on the `gui_swing` client driver type. The only way where i could run it without interruptions is with `chui_batch` driver - which i doubt is intended. Any idea on how to overcome this issue ?

Running with `chui` is possible (for example, to run test remotely using an ssh connection with no gui) and was even tested a bit.

But it will not help much in closing multiple dialogs, since you will be asked to press keyboard keys anyway.

To solve your problem, I would recommend you to create temporary runtime configurations to select specific packages, classes, or event test methods to investigate why some tests fail.

Another thing is that when a test fails due to assert, it does not exactly specify where it failed. If it would fail otherwise, i.e. due to NPE, the stacktrace is present which points to specific line in the generated .java class, which is easier to debug, as you can compare the 4GL test code with the generated code.

You could technically argue, that the test should only do one assertion, but I think we should have that information, at least the JVM stack which includes the .java source code line.

I think, this is a good idea. In addition, instead of raw Java stack (it is a bit messy) we could try to extract the 4gl stack and try to display it somehow in the reports and/or GUI.

I also have a problem with this test tests/dataset/parameter/input/TestDatasetAppend.cls, where you can find Assert:IsTrue(numCust = 3 and numItem = 5). assertions. On failure you just get Expected: TRUE but was: no error, not very helpful - I will split those into separate equality assertions, I think it makes sense.

This sounds reasonable for me either.

#23 - 04/25/2023 06:26 PM - Maciej Zieniuk

- File 4gl_results_20230426_1.xml added

- File 4gl_results_20230426_3.xml added

- File 4gl_results_20230426_2.xml added

I wanted to make sure that the dataset testcases actually work fine in 4GL ABLUnit.

With 4GL 11.6 version I was able to get to 423 running ABLUnit tests total, out of which there were 19 assertion failures and 12 runtime errors. See attached for result failures if you are interested.

I also had to disable a few test cases completely, as they were crashing the test run (disrupting other tests):

```
tests.dataset.TestDefault,  
tests.dataset.TestRelationsActive,  
tests.dataset.TestRowCreateUpdate,  
tests.dataset.TestRowDelete,  
tests.dataset.TestXmlNodeType,
```

With FWD I had to split it to 4 Test Suites runs (otherwise results in "legacy error" type of issues), which resulted in 127 assertion failures and 13

runtime errors out of 423 tests.

Compared to 4GL and excluding the disabled tests, we are still off by 108 assertion failures and 1 runtime error - investigating on those issues.

#24 - 04/25/2023 06:36 PM - Maciej Zieniuk

Oh i almost forgot, i have this weird issue, that i cannot run ant dist on the testcases repo, because it fails with error that tests/copy_lob/support/testcases_lob_data.jar file does not exist.

Every time i run ant dist and bzd status i notice that the file is gone (removed) and have to bring it back with bzd revert, but the ant dist fails anyway - i just disabled the <copy file="./tests/copy_lob/support/testcases_lob_data.jar" todir="\${deploy.home}/lib"/> line in the build.xml on my local env. Any idea why that happens ?

#25 - 04/26/2023 02:09 AM - Marian Edu

Maciej Zieniuk wrote:

Oh i almost forgot, i have this weird issue, that i cannot run ant dist on the testcases repo, because it fails with error that tests/copy_lob/support/testcases_lob_data.jar file does not exist.

Every time i run ant dist and bzd status i notice that the file is gone (removed) and have to bring it back with bzd revert, but the ant dist fails anyway - i just disabled the <copy file="./tests/copy_lob/support/testcases_lob_data.jar" todir="\${deploy.home}/lib"/> line in the build.xml on my local env.

Any idea why that happens ?

That file was moved in tests/copy_lob/support/data folder so the path needs to be updated in build.xml, unfortunately I can not push anything back in testcases repo so not much I can do to help for time being :(

#26 - 04/26/2023 05:04 PM - Maciej Zieniuk

Thanks Marian, that helped.

Few findings:

- tests.dataset.parameter.inout.TestInputOutputDynamicDsInDynamicProc method passInvalidDatasetHandle - i have added the handling of the 12314, invalid DATASET-HANDLE parameter given error. I placed it in the handle constructor of the DataSetParameter.java class:

```
public DataSetParameter(handle dsHandle, EnumSet<ParameterOption> options, boolean input, boolean output)
{
    BufferManager.registerScopeable(null);

    if (dsHandle.isUnknown() ||
        !(dsHandle.getResource() instanceof DataSet))
    {
        ErrorManager.recordOrShowError(12314);
        this.dataset = TypeFactory.handle();
    }
    else
    {
        this.originalDataSet = (DataSet)dsHandle.getResource();
        this.dataset = dsHandle;
    }

    this.options = options; // TODO: check default value for mode if ParameterOption.NONE is passed in
    this.input = input;
    this.output = output;
```

```
}

```

I am not exactly sure whether we have to handle the unknown or whether we have to recordOrShowError or recordOrThrowError. The reason for the if/else if to satisfy the finals.

- tests.dataset.parameter.inout.TestInputOutputDynamicDsInDynamicProc method passDatasetByReferenceAndEraseDatasetInCalledProcedure - the logic to handle the 12327 error is there in the DataSet.allowDelete class, method, but for whatever reason. because it happens in a different procedure, the error does not seem to be propagated. I tried to debug this, but i did not get to any other conclusion, except that the block scope / error set inside the called in procedure (from the test) is simply not being propagated, set back ? Not sure, i will have to dig in deeper into this.

- tests.dataset.parameter.inout.TestInputOutputDynamicDsInDynamicProc method passDiferentDatasetHandle - here we are trying to extract the unique-id of the handle that is unknown, which in FWD code results in error. This is because we try to resolve (unwrap) the handle into the UniqueID interface and then access the unique-id through the getUniqueID method. The issue here is that resolution/unwrapping, we get the handle.InvalidAttributeAccess class as a proxy, as the handle is unknown, so we end up calling handle.invalidAttribute, from which we get the 3135 and 3140 errors, instead of nothing.

I am hesitant on touching this code. I would at best make some special handling for the Uniqueid, but that does not feel right. I feel like I would need to understand the intention of the 3134, 3140 errors for this scenario, maybe they do not fit to all the cases when handle is unknown and we try to access something ?

- tests.dataset.parameter.inout.TestInputOutputInDatasetHandleProcedure and runWatchdogProcedure tests - i feel like the intention of this tests was for it to run after the other tests (like callProcedureWDeleteDatasetInProcedure), so probably in order as written in the .cls file ? I will investigate whether that assumption is correct by merging them together and checking the results (they also fail on the 4GL ABLUnit test run).

- tests.dataset.parameter.input.TestDatasetAppend - here is a specific order in which the tests have to run. They seem to be dependent on each other, which is obviously wrong. I will fix that by having a "cleanup" phase after each test and additional prepare in the tests to match the expectancy. (Same for TestDatasetAppendOo)

I would appreciate the feedback on any of this - also see my changes in relation to DataSetParameter change in the 4514a branch in p2j repo.

#27 - 04/27/2023 07:49 AM - Maciej Zieniuk

Marian, i don't think the support/test/HandleFactory.cls is converting in newest testcases revision, because of the support/test/handle_factory_helper.p procedure dependency, which contain issues.

I have both of these classes in file-cvt-list.txt file.

```
[java] support/test/handle_factory_helper.p
[java] Failure in file 'support/test/handle_factory_helper.p':
[java] com.goldencode.ast.AstException: Error processing ./support/test/handle_factory_helper.p
[java]   at com.goldencode.p2j.uast.AstGenerator.processFile (AstGenerator.java:1025)
[java]   at com.goldencode.p2j.uast.ScanDriver.lambda$scan$0 (ScanDriver.java:398)
[java]   at com.goldencode.p2j.uast.ScanDriver.scan (ScanDriver.java:434)
[java]   at com.goldencode.p2j.uast.ScanDriver.scan (ScanDriver.java:263)
[java]   at com.goldencode.p2j.convert.TransformDriver.runScanDriver (TransformDriver.java:390)
[java]   at com.goldencode.p2j.convert.TransformDriver.front (TransformDriver.java:255)
[java]   at com.goldencode.p2j.convert.TransformDriver.executeJob (TransformDriver.java:986)
```

```

[java] at com.goldencode.p2j.convert.ConversionDriver.main(ConversionDriver.java:1284)
[java] Caused by: java.lang.RuntimeException: Parser encountered 23 errors
[java] at com.goldencode.p2j.uast.AstGenerator.parse(AstGenerator.java:1659)
[java] at com.goldencode.p2j.uast.AstGenerator.processFile(AstGenerator.java:1013)
[java] ... 7 more
[java] support/test/HandleFactory.cls
[java] Lvl01 parse: ./support/test/HandleFactory.cls
[java] ./support/test/handle_factory_helper.p:16:31: expecting KW_FOR, found 'no-error'
[java] at antlr.Parser.match(Parser.java:211)
[java] at com.goldencode.p2j.uast.ProgressParser.for_table_clause(ProgressParser.java:53192)
[java] at com.goldencode.p2j.uast.ProgressParser.create_buffer_stmt(ProgressParser.java:29542)
[java] at com.goldencode.p2j.uast.ProgressParser.stmt_list(ProgressParser.java:27873)
[java] at com.goldencode.p2j.uast.ProgressParser.statement(ProgressParser.java:9819)
[java] at com.goldencode.p2j.uast.ProgressParser.single_block(ProgressParser.java:8375)
[java] at com.goldencode.p2j.uast.ProgressParser.then_clause(ProgressParser.java:39637)
[java] at com.goldencode.p2j.uast.ProgressParser.when_clause(ProgressParser.java:39761)
[java] at com.goldencode.p2j.uast.ProgressParser.case_stmt(ProgressParser.java:28689)
[java] at com.goldencode.p2j.uast.ProgressParser.stmt_list(ProgressParser.java:27403)
[java] at com.goldencode.p2j.uast.ProgressParser.statement(ProgressParser.java:9819)
[java] at com.goldencode.p2j.uast.ProgressParser.single_block(ProgressParser.java:8375)
[java] at com.goldencode.p2j.uast.ProgressParser.block(ProgressParser.java:7482)
[java] at com.goldencode.p2j.uast.ProgressParser.procedure(ProgressParser.java:8517)
[java] at com.goldencode.p2j.uast.ProgressParser.single_block(ProgressParser.java:8325)
[java] at com.goldencode.p2j.uast.ProgressParser.block(ProgressParser.java:7482)
[java] at com.goldencode.p2j.uast.ProgressParser.external_proc(ProgressParser.java:7409)
[java] at com.goldencode.p2j.uast.AstGenerator.parse(AstGenerator.java:1592)
[java] at com.goldencode.p2j.uast.AstGenerator.processFile(AstGenerator.java:1013)
[java] at com.goldencode.p2j.uast.ScanDriver.lambda$scan$0(ScanDriver.java:398)
[java] at com.goldencode.p2j.uast.ScanDriver.scan(ScanDriver.java:434)
[java] at com.goldencode.p2j.uast.ScanDriver.scan(ScanDriver.java:263)
[java] at com.goldencode.p2j.uast.TransformDriver.runScanDriver(TransformDriver.java:390)
[java] at com.goldencode.p2j.convert.TransformDriver.front(TransformDriver.java:255)
[java] at com.goldencode.p2j.convert.TransformDriver.executeJob(TransformDriver.java:986)
[java] at com.goldencode.p2j.convert.ConversionDriver.main(ConversionDriver.java:1284)
[java] ./support/test/handle_factory_helper.p:18:32: unexpected token: no-error
[java] at com.goldencode.p2j.uast.ProgressParser.create_dataset_stmt(ProgressParser.java:29884)
[java] at com.goldencode.p2j.uast.ProgressParser.stmt_list(ProgressParser.java:27889)
[java] at com.goldencode.p2j.uast.ProgressParser.statement(ProgressParser.java:9819)
[java] at com.goldencode.p2j.uast.ProgressParser.single_block(ProgressParser.java:8375)
[java] at com.goldencode.p2j.uast.ProgressParser.then_clause(ProgressParser.java:39637)
[java] at com.goldencode.p2j.uast.ProgressParser.when_clause(ProgressParser.java:39761)
[java] at com.goldencode.p2j.uast.ProgressParser.case_stmt(ProgressParser.java:28689)
[java] at com.goldencode.p2j.uast.ProgressParser.stmt_list(ProgressParser.java:27403)
[java] at com.goldencode.p2j.uast.ProgressParser.statement(ProgressParser.java:9819)
[java] at com.goldencode.p2j.uast.ProgressParser.single_block(ProgressParser.java:8375)
[java] at com.goldencode.p2j.uast.ProgressParser.block(ProgressParser.java:7482)
[java] at com.goldencode.p2j.uast.ProgressParser.procedure(ProgressParser.java:8517)
[java] at com.goldencode.p2j.uast.ProgressParser.single_block(ProgressParser.java:8325)
[java] at com.goldencode.p2j.uast.ProgressParser.block(ProgressParser.java:7482)
[java] at com.goldencode.p2j.uast.ProgressParser.external_proc(ProgressParser.java:7409)
[java] at com.goldencode.p2j.uast.AstGenerator.parse(AstGenerator.java:1592)
[java] at com.goldencode.p2j.uast.AstGenerator.processFile(AstGenerator.java:1013)
[java] at com.goldencode.p2j.uast.ScanDriver.lambda$scan$0(ScanDriver.java:398)
[java] at com.goldencode.p2j.uast.ScanDriver.scan(ScanDriver.java:434)
[java] at com.goldencode.p2j.uast.ScanDriver.scan(ScanDriver.java:263)
[java] at com.goldencode.p2j.uast.TransformDriver.runScanDriver(TransformDriver.java:390)
[java] at com.goldencode.p2j.convert.TransformDriver.front(TransformDriver.java:255)
[java] at com.goldencode.p2j.convert.TransformDriver.executeJob(TransformDriver.java:986)
[java] at com.goldencode.p2j.convert.ConversionDriver.main(ConversionDriver.java:1284)
[java] ./support/test/handle_factory_helper.p:19:5: unexpected token: when

```

Any idea ? I would probably need to revert to 1387, as the 1388 revision started heavily using it.

#28 - 04/27/2023 09:19 AM - Marian Edu

Maciej Zieniuk wrote:

Marian, i don't think the support/test/HandleFactory.cls is converting in newest testcases revision, because of the support/test/handle_factory_helper.p procedure dependency, which contain issues.
Any idea ? I would probably need to revert to 1387, as the 1388 revision started heavily using it.

Looks like create buffer was used there without for table, no syntax error on 4GL though but an error at runtime (probably a deviation that needs to be fixed). Anyway, fixed that and tried convert but when branches without do block in case statement seems not to work very well so added do block to all of them... still nothings gets converted for create dataset and create data-source statements and we do use those for testing dynamic objects (dataset/data-source).

Truth been told we do not even bother to convert the code right now but this looks like it will be a good test for conversion ;)

Pushed the changes in rev #1390.

#29 - 04/27/2023 09:30 AM - Greg Shah

Maciej: Please do fix all of these conversion issues. I don't want to leave behind any latent problems.

So we need to ensure that we have some examples of these failures in real test code that will be checked over time.

#30 - 04/28/2023 01:49 AM - Marian Edu

Maciej Zieniuk wrote:

- tests.dataset.parameter.inout.TestInputOutputDynamicDsInDynamicProc method passDatasetByReferenceAndEraseDatasetInCalledProcedure - the logic to handle the 12327 error is there in the DataSet.allowDelete class, method, but for whatever reason. because it happens in a different procedure, the error does not seem to be propagated. I tried to debug this, but i did not get to any other conclusion, except that the block scope / error set inside the called in procedure (from the test) is simply not being propagated, set back ? Not sure, i will have to dig in deeper into this.

That has something to do with the error handling from the sound of it, Constantin was fixing those kind of issues before so he probably knows best how to help here.

- tests.dataset.parameter.inout.TestInputOutputDynamicDsInDynamicProc method passDiferentDatasetHandle - here we are trying to extract the unique-id of the handle that is unknown, which in FWD code results in error. This is because we try to resolve (unwrap) the handle into the UniqueID interface and then access the unique-id through the getUniqueID method. The issue here is that resolution/unwrapping, we get the handle.InvalidAttributeAccess class as a proxy, as the handle is unknown, so we end up calling handle.invalidAttribute, from which we get the 3135 and 3140 errors, instead of nothing.

This sounds good to me, but then why the handle is invalid? I would check the converted code, from what we've seen with the 'factory helper' the create dataset statement is completely ignored so create dataset dsInIt. probably is not to be found in converted code hence accessing dsInIt:unique-id afterwards gives you those errors (which are expected if handle is invalid, this is not the issue).

I am hesitant on touching this code. I would at best make some special handling for the UniqueId, but that does not feel right. I feel like I would

need to understand the intention of the 3134, 3140 errors for this scenario, maybe they do not fit to all the cases when handle is unknown and we try to access something ?

The handle shouldn't be unknown, I will add an assert right after create dataset... sort of expect the unexpected, sorry for missing that :(

- tests.dataset.parameter.inout.TestInputOutputInDatasetHandleProcedure and runWatchdogProcedure tests - i feel like the intention of this tests was for it to run after the other tests (like callProcedureWDeleteDatasetInProcedure), so probably in order as written in the .cls file ? I will investigate whether that assumption is correct by merging them together and checking the results (they also fail on the 4GL ABLUnit test run).

Yes, the 'translation' of a single liniar test we had previously is not correct there... let me try to fix that and I will have the guy that worked on those review the rest of them.

- tests.dataset.parameter.input.TestDatasetAppend - here is a specific order in which the tests have to run. They seem to be dependent on each other, which is obviously wrong. I will fix that by having a "cleanup" phase after each test and additional prepare in the tests to match the expectancy. (Same for TestDatasetAppendOo)

Same as before, the test was not break down correctly, we will fix that.

#31 - 04/28/2023 02:27 AM - Maciej Zieniuk

This sounds good to me, but then why the handle is invalid? I would check the converted code, from what we've seen with the 'factory helper' the create dataset statement is completely ignored so create dataset dslnit. probably is not to be found in converted code hence accessing dslnit:unique-id afterwards gives you those errors (which are expected if handle is invalid, this is not the issue).

You are right. I initially though this is about dslnit being unknown, not initialized, but again, the reason why it's passing in 4GL is because of different methods execution order, which creates the dataset, while the FWD does not because of different order. I wonder how many other tests are broken simply because of this assumption (let's face it, i would say it's just badly written test).

we will fix that.

I can just continue the investigation on the other failures. I think it make sense, considering i still do not have the knowledge and full picture of how things work.

#32 - 04/28/2023 04:06 AM - Marian Edu

Maciej Zieniuk wrote:

You are right. I initially thought this is about dslNit being unknown, not initialized, but again, the reason why it's passing in 4GL is because of different methods execution order, which creates the dataset, while the FWD does not because of different order.

In this case it has nothing to do with the execution order of test methods, the dslNit was created just there in the test method. Looking at the converted code the object looks like is being created and not sure what is causing that error - either in the called procedure where unique-id is accessed or in the test after the procedure returns (maybe after the call the handle is not valid anymore, no idea).

I wonder how many other tests are broken simply because of this assumption (let's face it, i would say it's just badly written test).

Well, I guess I've just admitted the test was not correctly refactored for ABLUnit - probably the easiest would have been to convert previous tests in a single method test class :)

Those were not meant to run in suite and be completely independent of each other, in some cases this is simply impossible because of session wide system handles that can't always be reset... we're trying to make them better though.

#33 - 04/28/2023 04:18 AM - Marian Edu

Pushed a few changes, the conversion issue with create dataset/data-source was due to the fact that no-error option is not supported so I've took that out for the time being.

Aka, this doesn't convert because of the use of no-error:

```
def var hobj as handle.  
  
create dataset hobj no-error.  
create data-source hobj no-error.
```

Fixed one dataset test to completely clean-up all state between test methods, we will do the same with all other tests and fix those that have state dependency.

#34 - 04/28/2023 08:04 AM - Greg Shah

Aka, this doesn't convert because of the use of no-error:

What is the error? Is it during parsing or during later conversion?

#35 - 04/28/2023 09:00 AM - Marian Edu

Greg Shah wrote:

Aka, this doesn't convert because of the use of no-error:

What is the error? Is it during parsing or during later conversion?

```
./convert/data.p:3:19: unexpected token: no-error
  at com.goldencode.p2j.uast.ProgressParser.create_dataset_stmt (ProgressParser.java:29854)
  at com.goldencode.p2j.uast.ProgressParser.stmt_list (ProgressParser.java:27859)
  at com.goldencode.p2j.uast.ProgressParser.statement (ProgressParser.java:9789)
  at com.goldencode.p2j.uast.ProgressParser.single_block (ProgressParser.java:8345)
  at com.goldencode.p2j.uast.ProgressParser.block (ProgressParser.java:7452)
  at com.goldencode.p2j.uast.ProgressParser.external_proc (ProgressParser.java:7379)
  at com.goldencode.p2j.uast.AstGenerator.parse (AstGenerator.java:1592)
  at com.goldencode.p2j.uast.AstGenerator.processFile (AstGenerator.java:1013)
  at com.goldencode.p2j.uast.ScanDriver.lambda$0 (ScanDriver.java:398)
  at com.goldencode.p2j.uast.ScanDriver.scan (ScanDriver.java:434)
  at com.goldencode.p2j.uast.ScanDriver.scan (ScanDriver.java:263)
  at com.goldencode.p2j.convert.TransformDriver.runScanDriver (TransformDriver.java:390)
  at com.goldencode.p2j.convert.TransformDriver.front (TransformDriver.java:255)
  at com.goldencode.p2j.convert.TransformDriver.executeJob (TransformDriver.java:986)
  at com.goldencode.p2j.convert.ConversionDriver.main (ConversionDriver.java:1284)
  at testcases.Convert.main (Convert.java:22)
./convert/data.p:4:23: unexpected token: no-error
  at com.goldencode.p2j.uast.ProgressParser.create_datasource_stmt (ProgressParser.java:29911)
```

Not supported in antlr grammar, both create_dataset_stmt and create_datasource_stmt are missing the NO-ERROR option ((KW_NO_ERROR)?).

#36 - 04/28/2023 09:09 AM - Greg Shah

Correct.

```
=== modified file 'src/com/goldencode/p2j/uast/progress.g'  
--- old/src/com/goldencode/p2j/uast/progress.g 2023-03-27 09:55:57 +0000  
+++ new/src/com/goldencode/p2j/uast/progress.g 2023-04-28 13:07:24 +0000  
@@ -25535,6 +25535,7 @@  
     ds:KW_DATASET! { hide(ds); }  
     handle  
     (in_widget_pool_clause)?  
+     (KW_NO_ERROR)?  
     stmt_term  
     ;  
  
@@ -25550,6 +25551,7 @@  
     ds:KW_DATA_SRC! { hide(ds); }  
     handle  
     (in_widget_pool_clause)?  
+     (KW_NO_ERROR)?  
     stmt_term  
     ;
```

Maciej: Please include this change in your branch.

#37 - 04/28/2023 04:15 PM - Maciej Zieniuk

I am not sure what to do with the tests.dataset.parameter.input.TestDatasetBind class test referenceOnlyNotValid. Here, it is failing on the line:

```
run isOrderBind in hproc (output retVal).  
Assert:.IsTrue(not retVal).
```

where the isOrderBind is located in support/dataset/help/io_static_parameters.p:

```
procedure isOrderBind:  
    define output parameter retVal as logical no-undo.  
  
    for each ttOrderCustomer:  
        end.  
  
    retVal = true.  
  
    catch e as Progress.Lang.Error :  
        retVal = false.  
    end catch.
```

end procedure.

The expectation is that the for each ttOrderCustomer: will fail with Progress.Lang.Error exception LegacyErrorException type, but we get QueryOffEndException during the AdaptiveQuery.next.

The dataset dsOrder and temp-tables ttOrderCustomer are defined in the io_static_parameters.p as reference-only, so until the dataset BIND or pass by reference to the procedure is called, then technically the ttOrderCustomer have unknown handle ? So i would understood, this that doing for each on an unknown / unbound temp table / dataset is invalid ?

I will try to figure out what kind of exception, error 4GL is supposed to be thrown here, maybe we should be more explicit when validating, rather than just a "catch-all".

#38 - 04/28/2023 09:56 PM - Ovidiu Maxiniuc

Maciej Zieniuk wrote:

The expectation is that the for each ttOrderCustomer: will fail with Progress.Lang.Error exception LegacyErrorException type, but we get QueryOffEndException during the AdaptiveQuery.next.

Exactly. The reference-only / bind was added recently to FWD and we do not have a full understanding of what they imply. This is one of the cases. The situation where the buffer about to be used is reference-only and unbound must be detected and an error (Progress.Lang.Error) must be raised. I put to test this isolated testcase and FWD kicked the client because the SQL table was not created. If you got the QueryOffEndException then the buffer was already bound probably.

#39 - 04/29/2023 03:38 AM - Marian Edu

Maciej Zieniuk wrote:

The expectation is that the for each ttOrderCustomer: will fail with Progress.Lang.Error exception LegacyErrorException type, but we get QueryOffEndException during the AdaptiveQuery.next.

I don't know anything about QueryOffEndException but any catch block in 4GL does expect a kind of Progress.Lang.Error :)

The dataset dsOrder and temp-tables ttOrderCustomer are defined in the io_static_parameters.p as reference-only, so until the dataset BIND or pass by reference to the procedure is called, then technically the ttOrderCustomer have unknown handle ?

Yes.

So i would understood, this that doing for each on an unknown / unbound temp table / dataset is invalid ?

Yes.

I will try to figure out what kind of exception, error 4GL is supposed to be thrown here, maybe we should be more explicit when validating, rather than just a "catch-all".

That is not the purpose of the test - aka to find out the exact error numbers - but to test the bind/reference-only options when passing datasets as parameters.

If that is to be done I would say the right place is in temp-table suite not dataset, we have some tests there but also mostly for parameter passing and almost all other tests for methods/attributes are simply boiler plate that we generated and added some simple tests here and there. Some doesn't even compile in 4GL cause we've stoped extending them as there were not part of previous tests, not everything under dataset and temptable was tested and that includes using a reference-only table that was not bound.

I did a quick extension of NUM-REFERENCES test in temp-table (tests/table/TestNumReferences.cls), basically accessing any method/attribute on such a temptable should raise an error - same holds true for when used in a for-each/find/open-query but hey, we do not have tests for all those cases, this doesn't mean we need to assert any possible errors on each tests as you suggest here but feel free to extend them the way you see fit.

#40 - 05/01/2023 03:43 PM - Vladimir Tsichevski

Maciej, can you provide me and other with the file-cvt-list.txt you use to convert tests for this issue, please. Thanks.

#41 - 05/02/2023 03:00 AM - Maciej Zieniuk

- File file-cvt-list.txt added

Attached file-cvt-list.txt

#42 - 05/02/2023 04:28 PM - Maciej Zieniuk

I just pushed a potential fix to 4514a p2j branch, for the tests.dataset.parameter.input.TestInputDynamicDsInDynamicProc class method notUsedForPersitentProcedureWithDynamicDatasetDefined. Basically i have added handling of BIND when dynamic dataset is needed. Not sure whether i did it correctly - i deduced it's probably a problem with the FWD code, rather than conversion, again, might be wrong.

On the same class, there is the passDatasetByReferenceAndEraseDatasetInCalledProcedure method, which was supposed to throw the 12327 error. Again, like in previous cases, the error is thrown, but does not propagate further to the test because of the no-error (or silent in FWD). But then after the no-error/silent block, the error disappears and the assert fails, that error was expected but nothing is there. Again, this seems like an issue with the error propagation, but i don't fully understand where is the issue.

Any help on this? :)

#43 - 05/03/2023 11:44 AM - Greg Shah

Vladimir has some error propagation changes in task branch 3827b. Can you try those to see if the issue is resolved?

#44 - 05/03/2023 02:48 PM - Maciej Zieniuk

This test which had error propagation issues is working with the 3827b branch :)

I will rebase my changes on top of 3827b and see how it goes.

There is one more issue which i tried to debug. The tests.dataset.parameter.output.TestOutputDynamicDsInDynamicProc test passWrongDefinition method, when that runs (with error at the moment), no other test will run correctly.

The next test after this one for me is passDatasetByReferenceAddRecord, which will fail during setup on below assertion:

```
@Setup.  
method public void setUpBeforeClass( ):  
  
//define dataset as dynamic with no pk and get ttCustomer  
run support/dataset/create/dsMaster.p(false, false, false, output dsMaster).  
Assert.IsTrue(valid-handle (dsMaster)).
```

Note: For ABLUnit, the setUpBeforeClass is actually a before each test type.

I identified the issue, thhat the run support/dataset/create/dsMaster.p does not propagate the assigned handle output dsMaster out of the calling procedure - it get's assigned when i debug inside of the procedure, but as soon as i exit the stack, to back to the setUpBeforeClass block / scope, the dsMaster is unassigned / unknown.

I might be wrong, but from what i observed, in the BlockManager.topLevelBlock, the OutputParameterAssigner opa = wa.tm.deregisterOutputParameterAssigner(); returns null but probably should not ? It might have something to do with the blocks not being cleaned up completely after the test execution ? (guessing that one)

#45 - 05/03/2023 03:01 PM - Greg Shah

I identified the issue, thhat the run support/dataset/create/dsMaster.p does not propagate the assigned handle output dsMaster out of the calling procedure - it get's assigned when i debug inside of the procedure, but as soon as i exit the stack, to back to the setUpBeforeClass block / scope, the dsMaster is unassigned / unknown.

Vladimir: Please help Maciej with this issue.

#46 - 05/03/2023 04:30 PM - Vladimir Tsichevski

Greg Shah wrote:

I identified the issue, thhat the run support/dataset/create/dsMaster.p does not propagate the assigned handle output dsMaster out of the calling procedure - it get's assigned when i debug inside of the procedure, but as soon as i exit the stack, to back to the setUpBeforeClass block / scope, the dsMaster is unassigned / unknown.

Vladimir: Please help Maciej with this issue.

OK

#47 - 05/03/2023 09:30 PM - Ovidiu Maxiniuc

Maciej Zieniuk wrote:

I identified the issue, thhat the run support/dataset/create/dsMaster.p does not propagate the assigned handle output dsMaster out of the calling procedure - it get's assigned when i debug inside of the procedure, but as soon as i exit the stack, to back to the setUpBeforeClass block / scope, the dsMaster is unassigned / unknown.

I understand from the context that dsMaster is an output handle variable/parameter. It should be assigned by the OutputParameterAssigner.processAssignments() at the end of BlockManager.topLevelBlock().

Note that for dataset parameters (ie not a handle), in case the variable is bound, the value printed in debugger (usually toString()) might not reflect its actual internal state due the proxy delegation.

#48 - 05/04/2023 05:16 PM - Vladimir Tsichevski

I have an issue compiling the TestHandle.cls test: statements like

```
dsMasterDynamic:handle no-error.
```

are valid in OE, but incorrectly converted by FWD as:

```
silent(() -> dsMasterDynamic);  
dsMasterDynamic;
```

I used 6237b to convert, I think, the 4514a will have this error either, since I have not seen any conversion-related changes in this branch.

#49 - 05/04/2023 05:25 PM - Maciej Zieniuk

In my testcases branch i changed this code to assign the handle to some other handle, as it did not compile for me.

I generally don't understand what this testGet method is testing, since, the dsMasterDynamic is already a handle and we check for the validity during the setup, so what we are exactly testing here ? I would expect something like handle -> DataSet type of get would make sense here ?

#50 - 05/04/2023 05:41 PM - Vladimir Tsichevski

Maciej Zieniuk wrote:

In my testcases branch i changed this code to assign the handle to some other handle, as it did not compile for me.

I generally don't understand what this testGet method is testing, since, the dsMasterDynamic is already a handle and we check for the validity during the setup, so what we are exactly testing here ? I would expect something like handle -> DataSet type of get would make sense here ?

I gather, this code checks the handle is valid. For example, the same code with no no-error generates the Invalid handle... for an unknown value.

So, we have probably to create a task for this problem? Greg, what do you think?

Until this fixed, we can replace the statement by an assignment, as you did.

#51 - 05/04/2023 06:11 PM - Greg Shah

We might as well fix it here, now.

It is weird syntax. Normally you would call valid-handle(dsMasterDynamic) to check validity. Still, if it is valid in OE then let's fix it.

#52 - 05/05/2023 07:26 AM - Marian Edu

Greg Shah wrote:

We might as well fix it here, now.

It is weird syntax.

It just looks like that, however HANDLE is a valid attribute for all handle based objects [https://documentation.progress.com/output/ua/OpenEdge_latest/pdsoe/PLUGINS_ROOT/com.openedge.pdt.langref.help/rfi1424920272514.html#rfi1424920272514]. It's true normally it is used for static objects to get the handle and for dynamic will kinda return self - unless of course the object is invalid in which case it will throw error as for when trying to access any attribute on an invalid object.

Normally you would call valid-handle(dsMasterDynamic) to check validity.

We're not trying to check if the object is valid or not, just testing if accessing the HANDLE attribute (what is being tested there) succeed or fails.

Still, if it is valid in OE then let's fix it.

Accessing any object attribute on dynamic handle is valid 4GL syntax (even if the attribute does not exist for the particular object), this applies to HANDLE attribute as well :)

#53 - 05/05/2023 07:33 AM - Marian Edu

Did pushed a new revision on testcases #1405 that includes some changes in the dataset tests, mainly clean-up after each test method to make sure there are no other dependencies on the execution order.

#54 - 05/05/2023 08:50 AM - Vladimir Tsichevski

Another conversion issue: dataset.TestAcceptChanges legacy class converts to:

```
public void __tests_dataset_TestAcceptChanges_execute__()
{
    ...
    externalProcedure(TestAcceptChanges.class, TestAcceptChanges.this, new Block((Body) () ->
    {
        RecordBuffer.openScope(customer, item, ttcustomer, bttCustomer, ttitem, bttItem);
        onBlockLevel(Condition.ERROR, Action.THROW);
        {
;
        }
    }));
}
```

Note the extra **semicolon** in the block body. Eclipse considers this a problem, but more important, I suspect there should be some real code here.

#55 - 05/05/2023 04:38 PM - Vladimir Tsichevski

Maciej Zieniuk wrote:

There is one more issue which i tried to debug. The tests.dataset.parameter.output.TestOutputDynamicDsInDynamicProc test passWrongDefinition method, when that runs (with error at the moment), no other test will run correctly.

The next test after this one for me is passDatasetByReferenceAddRecord, which will fail during setup on below assertion:

[...]

Note: For ABLUnit, the setUpBeforeClass is actually a before each test type.

I identified the issue, that the run support/dataset/create/dsMaster.p does not propagate the assigned handle output dsMaster out of the calling procedure - it get's assigned when i debug inside of the procedure, but as soon as i exit the stack, to back to the setUpBeforeClass block /

scope, the dsMaster is unassigned / unknown.

I ran this test with the latest 3827b (rebased to the latest trunk). The setUpBeforeClass runs with no issues, both dsMaster and dsOrder were valid handles.

#56 - 05/06/2023 04:42 AM - Maciej Zieniuk

Vladimir, the setUpBeforeClass runs fine on it's own. But let the passWrongDefinition test run and then any other test method that runs setUpBeforeClass will have this problem and invalid handles.

#57 - 05/06/2023 10:30 AM - Maciej Zieniuk

I found this exception from the server's logs, that happens inside the passWrongDefinition. I am guessing they are the cause of the errors i am describing.

```
[05/06/2023 16:31:39 CEST] (TransactionManager.registerScopeable:WARNING) {00000001:0000000C:bogus} <depth = 4
; trans_level = -1; trans_label = null; rollback_scope = -1; rollback_label = null; rollback_pending = false;
in_quit = false; retry_scope = -1; retry_label = null; ignore_err = false> [label = methodScope; type = INTERN
AL_PROC; full = false; trans_level = SUB_TRANSACTION; external = false; top_level = true; loop = false; loop_p
rotection = true; had_pause = false; endkey_retry = false; next_or_leave = leave; is_retry = false; needs_retr
y = false; FOR (aggressive) flushing = false; ilp_count = -1; pending_break = false; database_trigger = false;
properties = 'ERROR, ENDKEY'; finally = none] Illegal attempt to register Scopeable
java.lang.Throwable
  at com.goldencode.p2j.util.TransactionManager.registerScopeable (TransactionManager.java:3658)
  at com.goldencode.p2j.util.TransactionManager.registerBlockScopeable (TransactionManager.java:3605)
  at com.goldencode.p2j.util.TransactionManager.access$5700 (TransactionManager.java:691)
  at com.goldencode.p2j.util.TransactionManager$TransactionHelper.registerBlockScopeable (TransactionMana
ger.java:8042)
  at com.goldencode.p2j.util.ObjectOps.pendingInitialize (ObjectOps.java:2180)
  at com.goldencode.p2j.util.ObjectOps.newInstanceInternal (ObjectOps.java:1302)
  at com.goldencode.p2j.util.ObjectOps.newInstanceInternal (ObjectOps.java:1221)
  at com.goldencode.p2j.util.ObjectOps.newInstance (ObjectOps.java:1148)
  at com.goldencode.p2j.oo.lang.SysError.newInstance (SysError.java:159)
  at com.goldencode.p2j.util.ErrorManager.recordForLegacyThrow (ErrorManager.java:2734)
  at com.goldencode.p2j.util.ErrorManager$ServerDataAccess.recordForLegacyThrow (ErrorManager.java:4568)
  at com.goldencode.p2j.util.ErrorManager.recordOrShowError (ErrorManager.java:1208)
  at com.goldencode.p2j.util.ErrorManager.recordOrShowError (ErrorManager.java:1140)
  at com.goldencode.p2j.util.ErrorManager.recordOrShowError (ErrorManager.java:1099)
  at com.goldencode.p2j.util.ErrorManager.recordOrShowError (ErrorManager.java:1346)
  at com.goldencode.p2j.persist.TemporaryBuffer.handleTablesDoNotMatch (TemporaryBuffer.java:5145)
  at com.goldencode.p2j.persist.TemporaryBuffer.createPropsMap (TemporaryBuffer.java:4449)
  at com.goldencode.p2j.persist.TemporaryBuffer.copyAllRows (TemporaryBuffer.java:3972)
  at com.goldencode.p2j.persist.AbstractTempTable.copyTempTable (AbstractTempTable.java:540)
  at com.goldencode.p2j.persist.BufferImpl.copyTempTable (BufferImpl.java:6466)
  at com.goldencode.p2j.persist.DataSet.copyDatasetImpl (DataSet.java:6089)
  at com.goldencode.p2j.persist.OutputDataSetHandleCopier.finishedImpl (OutputDataSetHandleCopier.java:22
8)
  at com.goldencode.p2j.persist.OutputDataSetHandleCopier.lambda$finished$0 (OutputDataSetHandleCopier.ja
va:136)
  at com.goldencode.p2j.jmx.NanoTimer.timer (NanoTimer.java:131)
  at com.goldencode.p2j.persist.OutputDataSetHandleCopier.finished (OutputDataSetHandleCopier.java:136)
  at com.goldencode.p2j.util.ProcedureManager$WorkArea.scopeFinished (ProcedureManager.java:5217)
  at com.goldencode.p2j.util.TransactionManager.processScopeNotifications (TransactionManager.java:7655)
  at com.goldencode.p2j.util.TransactionManager.popScope (TransactionManager.java:4748)
  at com.goldencode.p2j.util.TransactionManager.access$7200 (TransactionManager.java:691)
  at com.goldencode.p2j.util.TransactionManager$TransactionHelper.popScope (TransactionManager.java:8826)
  at com.goldencode.p2j.util.BlockManager.topLevelBlock (BlockManager.java:8586)
  at com.goldencode.p2j.util.BlockManager.internalProcedure (BlockManager.java:715)
  at com.goldencode.p2j.util.BlockManager.internalProcedure (BlockManager.java:692)
  at com.goldencode.testcases.support.dataset.help.IoDynamicParameters.dsAsOutput (IoDynamicParameters.ja
va:127)
  at com.goldencode.testcases.support.dataset.help.IoDynamicParametersMethodAccess.invoke (Unknown Source
)
```

```

at com.goldencode.p2j.util.ControlFlowOps$InternalEntryCaller.invokeImpl (ControlFlowOps.java:9376)
at com.goldencode.p2j.util.ControlFlowOps$InternalEntryCaller.invoke (ControlFlowOps.java:9332)
at com.goldencode.p2j.util.ControlFlowOps.lambda$invokeImpl$10 (ControlFlowOps.java:7295)
at com.goldencode.p2j.util.ControlFlowOps.invokeImpl (ControlFlowOps.java:7310)
at com.goldencode.p2j.util.ControlFlowOps.invoke (ControlFlowOps.java:4446)
at com.goldencode.p2j.util.ControlFlowOps.invokeImpl (ControlFlowOps.java:6876)
at com.goldencode.p2j.util.ControlFlowOps.invokeImpl (ControlFlowOps.java:6779)
at com.goldencode.p2j.util.ControlFlowOps.invokeInImpl (ControlFlowOps.java:6735)
at com.goldencode.p2j.util.ControlFlowOps.invokeInWithMode (ControlFlowOps.java:1879)
at com.goldencode.p2j.util.ControlFlowOps.invoke (ControlFlowOps.java:1305)
at com.goldencode.p2j.util.InvokeConfig.run (InvokeConfig.java:425)
at com.goldencode.testcases.tests.dataset.parameter.output.TestOutputDynamicDsInDynamicProc.lambda$null$11 (TestOutputDynamicDsInDynamicProc.java:219)
at com.goldencode.p2j.util.ErrorManager.silentWorker (ErrorManager.java:3873)
at com.goldencode.p2j.util.ErrorManager.silent (ErrorManager.java:681)
at com.goldencode.testcases.tests.dataset.parameter.output.TestOutputDynamicDsInDynamicProc.lambda$passWrongDefinition$12 (TestOutputDynamicDsInDynamicProc.java:219)
at com.goldencode.p2j.util.Block.body (Block.java:636)
at com.goldencode.p2j.util.BlockManager.processBody (BlockManager.java:8838)
at com.goldencode.p2j.util.BlockManager.topLevelBlock (BlockManager.java:8504)
at com.goldencode.p2j.util.BlockManager.internalProcedure (BlockManager.java:766)
at com.goldencode.p2j.util.BlockManager.internalProcedure (BlockManager.java:739)
at com.goldencode.testcases.tests.dataset.parameter.output.TestOutputDynamicDsInDynamicProc.passWrongDefinition (TestOutputDynamicDsInDynamicProc.java:212)
at com.goldencode.testcases.tests.dataset.parameter.output.TestOutputDynamicDsInDynamicProcMethodAccess.invoke (Unknown Source)
at com.goldencode.p2j.util.ControlFlowOps$InternalEntryCaller.invokeImpl (ControlFlowOps.java:9376)
at com.goldencode.p2j.util.ControlFlowOps$InternalEntryCaller.invoke (ControlFlowOps.java:9332)
at com.goldencode.p2j.util.ControlFlowOps.invokeLegacyMethod (ControlFlowOps.java:5103)
at com.goldencode.p2j.util.ControlFlowOps.invokeLegacyMethod (ControlFlowOps.java:5024)
at com.goldencode.p2j.util.ObjectOps.invoke (ObjectOps.java:3028)
at com.goldencode.p2j.util.ObjectOps.invokeStandalone (ObjectOps.java:912)
at com.goldencode.p2j.testengine.TestExecutionSupport.lambda$callClassMethod$0 (TestExecutionSupport.java:201)
at com.goldencode.p2j.testengine.TestExecutionSupport.lambda$doInBlock$4 (TestExecutionSupport.java:271)
)
at com.goldencode.p2j.util.Block.body (Block.java:636)
at com.goldencode.p2j.util.BlockManager.processBody (BlockManager.java:8838)
at com.goldencode.p2j.util.BlockManager.doBlockWorker (BlockManager.java:9993)
at com.goldencode.p2j.util.BlockManager.doBlock (BlockManager.java:1489)
at com.goldencode.p2j.testengine.TestExecutionSupport.doInBlock (TestExecutionSupport.java:263)
at com.goldencode.p2j.testengine.TestExecutionSupport.callClassMethod (TestExecutionSupport.java:170)
at com.goldencode.p2j.testengine.AbstractFWDTestDescriptor.callMethodImpl (AbstractFWDTestDescriptor.java:500)
at com.goldencode.p2j.testengine.AbstractFWDTestDescriptor.callMethodImpl (AbstractFWDTestDescriptor.java:471)
at com.goldencode.p2j.testengine.AbstractMethodTestDescriptor.executeTestMethod (AbstractMethodTestDescriptor.java:412)
at com.goldencode.p2j.testengine.AbstractMethodTestDescriptor.executeImpl (AbstractMethodTestDescriptor.java:273)
at com.goldencode.p2j.testengine.UnitTestServer.lambda$execute$8 (UnitTestServer.java:369)
at com.goldencode.p2j.testengine.UnitTestServer.convertError (UnitTestServer.java:448)
at com.goldencode.p2j.testengine.UnitTestServer.execute (UnitTestServer.java:369)
at com.goldencode.p2j.testengine.UnitTestServerTestMethodAccess.invoke (Unknown Source)
at com.goldencode.p2j.util.MethodInvoker.invoke (MethodInvoker.java:156)
at com.goldencode.p2j.net.Dispatcher.processInbound (Dispatcher.java:786)
at com.goldencode.p2j.net.Conversation.block (Conversation.java:422)
at com.goldencode.p2j.net.Conversation.run (Conversation.java:232)
at java.lang.Thread.run (Thread.java:750)
[05/06/2023 16:31:39 CEST] (TransactionManager.registerPendingScopeable:WARNING) {00000001:0000000C:bogus} <depth = 4; trans_level = -1; trans_label = null; rollback_scope = -1; rollback_label = null; rollback_pending = false; in_quit = false; retry_scope = -1; retry_label = null; ignore_err = false> [label = methodScope; type = INTERNAL_PROC; full = false; trans_level = SUB_TRANSACTION; external = false; top_level = true; loop = false; loop_protection = true; had_pause = false; endkey_retry = false; next_or_leave = leave; is_retry = false; need_ds_retry = false; FOR (aggressive) flushing = false; ilp_count = -1; pending_break = false; database_trigger = false; properties = 'ERROR, ENDKEY'; finally = none] Illegal attempt to register Scopeable
java.lang.Throwable
at com.goldencode.p2j.util.TransactionManager.registerPendingScopeable (TransactionManager.java:3622)
at com.goldencode.p2j.util.TransactionManager.access$5800 (TransactionManager.java:691)
at com.goldencode.p2j.util.TransactionManager$TransactionHelper.registerPendingScopeable (TransactionManager.java:8053)
at com.goldencode.p2j.util.ObjectOps.registerPendingScopeable (ObjectOps.java:2115)
at com.goldencode.p2j.util.BlockManager.topLevelBlock (BlockManager.java:8441)
at com.goldencode.p2j.util.BlockManager.externalProcedure (BlockManager.java:555)
at com.goldencode.p2j.util.BlockManager.externalProcedure (BlockManager.java:528)

```

```

at com.goldencode.p2j.oo.lang.SysError.___lang_SysError_execute___(SysError.java:92)
at com.goldencode.p2j.oo.lang.SysErrorMethodAccess.invoke(Unknown Source)
at com.goldencode.p2j.util.ControlFlowOps$InternalEntryCaller.invokeImpl(ControlFlowOps.java:9376)
at com.goldencode.p2j.util.ControlFlowOps$InternalEntryCaller.invoke(ControlFlowOps.java:9332)
at com.goldencode.p2j.util.ControlFlowOps.initializeLegacyObject(ControlFlowOps.java:4874)
at com.goldencode.p2j.util.ObjectOps.newInstanceInternal(ObjectOps.java:1308)
at com.goldencode.p2j.util.ObjectOps.newInstanceInternal(ObjectOps.java:1221)
at com.goldencode.p2j.util.ObjectOps.newInstance(ObjectOps.java:1148)
at com.goldencode.p2j.oo.lang.SysError.newInstance(SysError.java:159)
at com.goldencode.p2j.util.ErrorManager.recordForLegacyThrow(ErrorManager.java:2734)
at com.goldencode.p2j.util.ErrorManager$ServerDataAccess.recordForLegacyThrow(ErrorManager.java:4568)
at com.goldencode.p2j.util.ErrorManager.recordOrShowError(ErrorManager.java:1208)
at com.goldencode.p2j.util.ErrorManager.recordOrShowError(ErrorManager.java:1140)
at com.goldencode.p2j.util.ErrorManager.recordOrShowError(ErrorManager.java:1099)
at com.goldencode.p2j.util.ErrorManager.recordOrShowError(ErrorManager.java:1346)
at com.goldencode.p2j.persist.TemporaryBuffer.handleTablesDoNotMatch(TemporaryBuffer.java:5145)
at com.goldencode.p2j.persist.TemporaryBuffer.createPropsMap(TemporaryBuffer.java:4449)
at com.goldencode.p2j.persist.TemporaryBuffer.copyAllRows(TemporaryBuffer.java:3972)
at com.goldencode.p2j.persist.AbstractTempTable.copyTempTable(AbstractTempTable.java:540)
at com.goldencode.p2j.persist.BufferImpl.copyTempTable(BufferImpl.java:6466)
at com.goldencode.p2j.persist.DataSet.copyDatasetImpl(DataSet.java:6089)
at com.goldencode.p2j.persist.OutputDataSetHandleCopier.finishedImpl(OutputDataSetHandleCopier.java:22
8)
at com.goldencode.p2j.persist.OutputDataSetHandleCopier.lambda$finished$0(OutputDataSetHandleCopier.ja
va:136)
at com.goldencode.p2j.jmx.NanoTimer.timer(NanoTimer.java:131)
at com.goldencode.p2j.persist.OutputDataSetHandleCopier.finished(OutputDataSetHandleCopier.java:136)
at com.goldencode.p2j.util.ProcedureManager$WorkArea.scopeFinished(ProcedureManager.java:5217)
at com.goldencode.p2j.util.TransactionManager.processScopeNotifications(TransactionManager.java:7655)
at com.goldencode.p2j.util.TransactionManager.popScope(TransactionManager.java:4748)
at com.goldencode.p2j.util.TransactionManager.access$7200(TransactionManager.java:691)
at com.goldencode.p2j.util.TransactionManager$TransactionHelper.popScope(TransactionManager.java:8826)
at com.goldencode.p2j.util.BlockManager.topLevelBlock(BlockManager.java:8586)
at com.goldencode.p2j.util.BlockManager.internalProcedure(BlockManager.java:715)
at com.goldencode.p2j.util.BlockManager.internalProcedure(BlockManager.java:692)
at com.goldencode.testcases.support.dataset.help.IoDynamicParameters.dsAsOutput(IoDynamicParameters.ja
va:127)
at com.goldencode.testcases.support.dataset.help.IoDynamicParametersMethodAccess.invoke(Unknown Source
)
at com.goldencode.p2j.util.ControlFlowOps$InternalEntryCaller.invokeImpl(ControlFlowOps.java:9376)
at com.goldencode.p2j.util.ControlFlowOps$InternalEntryCaller.invoke(ControlFlowOps.java:9332)
at com.goldencode.p2j.util.ControlFlowOps.lambda$invokeImpl$10(ControlFlowOps.java:7295)
at com.goldencode.p2j.util.ControlFlowOps.invokeImpl(ControlFlowOps.java:7310)
at com.goldencode.p2j.util.ControlFlowOps.invoke(ControlFlowOps.java:4446)
at com.goldencode.p2j.util.ControlFlowOps.invokeImpl(ControlFlowOps.java:6876)
at com.goldencode.p2j.util.ControlFlowOps.invokeImpl(ControlFlowOps.java:6779)
at com.goldencode.p2j.util.ControlFlowOps.invokeInImpl(ControlFlowOps.java:6735)
at com.goldencode.p2j.util.ControlFlowOps.invokeInWithMode(ControlFlowOps.java:1879)
at com.goldencode.p2j.util.ControlFlowOps.invoke(ControlFlowOps.java:1305)
at com.goldencode.p2j.util.InvokeConfig.run(InvokeConfig.java:425)
at com.goldencode.testcases.tests.dataset.parameter.output.TestOutputDynamicDsInDynamicProc.lambda$nul
l$11(TestOutputDynamicDsInDynamicProc.java:219)
at com.goldencode.p2j.util.ErrorManager.silentWorker(ErrorManager.java:3873)
at com.goldencode.p2j.util.ErrorManager.silent(ErrorManager.java:681)
at com.goldencode.testcases.tests.dataset.parameter.output.TestOutputDynamicDsInDynamicProc.lambda$pas
sWrongDefinition$12(TestOutputDynamicDsInDynamicProc.java:219)
at com.goldencode.p2j.util.Block.body(Block.java:636)
at com.goldencode.p2j.util.BlockManager.processBody(BlockManager.java:8838)
at com.goldencode.p2j.util.BlockManager.topLevelBlock(BlockManager.java:8504)
at com.goldencode.p2j.util.BlockManager.internalProcedure(BlockManager.java:766)
at com.goldencode.p2j.util.BlockManager.internalProcedure(BlockManager.java:739)
at com.goldencode.testcases.tests.dataset.parameter.output.TestOutputDynamicDsInDynamicProc.passWrongD
efinition(TestOutputDynamicDsInDynamicProc.java:212)
at com.goldencode.testcases.tests.dataset.parameter.output.TestOutputDynamicDsInDynamicProcMethodAcces
s.invoke(Unknown Source)
at com.goldencode.p2j.util.ControlFlowOps$InternalEntryCaller.invokeImpl(ControlFlowOps.java:9376)
at com.goldencode.p2j.util.ControlFlowOps$InternalEntryCaller.invoke(ControlFlowOps.java:9332)
at com.goldencode.p2j.util.ControlFlowOps.invokeLegacyMethod(ControlFlowOps.java:5103)
at com.goldencode.p2j.util.ControlFlowOps.invokeLegacyMethod(ControlFlowOps.java:5024)
at com.goldencode.p2j.util.ObjectOps.invoke(ObjectOps.java:3028)
at com.goldencode.p2j.util.ObjectOps.invokeStandalone(ObjectOps.java:912)
at com.goldencode.p2j.testengine.TestExecutionSupport.lambda$callClassMethod$0(TestExecutionSupport.ja
va:201)
at com.goldencode.p2j.testengine.TestExecutionSupport.lambda$doInBlock$4(TestExecutionSupport.java:271
)

```

```
at com.goldencode.p2j.util.Block.body(Block.java:636)
at com.goldencode.p2j.util.BlockManager.processBody(BlockManager.java:8838)
at com.goldencode.p2j.util.BlockManager.doBlockWorker(BlockManager.java:9993)
at com.goldencode.p2j.util.BlockManager.doBlock(BlockManager.java:1489)
at com.goldencode.p2j.testengine.TestExecutionSupport.doInBlock(TestExecutionSupport.java:263)
at com.goldencode.p2j.testengine.TestExecutionSupport.callClassMethod(TestExecutionSupport.java:170)
at com.goldencode.p2j.testengine.AbstractFWDTestDescriptor.callMethodImpl(AbstractFWDTestDescriptor.java:500)
at com.goldencode.p2j.testengine.AbstractFWDTestDescriptor.callMethodImpl(AbstractFWDTestDescriptor.java:471)
at com.goldencode.p2j.testengine.AbstractMethodTestDescriptor.executeTestMethod(AbstractMethodTestDescriptor.java:412)
at com.goldencode.p2j.testengine.AbstractMethodTestDescriptor.executeImpl(AbstractMethodTestDescriptor.java:273)
at com.goldencode.p2j.testengine.UnitTestServer.lambda$execute$8(UnitTestServer.java:369)
at com.goldencode.p2j.testengine.UnitTestServer.convertError(UnitTestServer.java:448)
at com.goldencode.p2j.testengine.UnitTestServer.execute(UnitTestServer.java:369)
at com.goldencode.p2j.testengine.UnitTestServerMethodAccess.invoke(Unknown Source)
at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:156)
at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:786)
at com.goldencode.p2j.net.Conversation.block(Conversation.java:422)
at com.goldencode.p2j.net.Conversation.run(Conversation.java:232)
at java.lang.Thread.run(Thread.java:750)
```

#58 - 05/06/2023 12:15 PM - Vladimir Tsichevski

Maciej Zieniuk wrote:

I found this exception from the server's logs, that happens inside the passWrongDefinition. I am guessing they are the cause of the errors i am describing.
[...]

I see similar error. The offending code is:

```
private static void registerScopeable(WorkArea wa, BlockDefinition block, Scopeable target, boolean start)
{
    int id = target.getScopeId().ordinal();

    if (wa.nowIterating != null && wa.nowIterating == block.scopeList && block.scopeList[id] == null)
    {
        if (LOG.isLoggable(Level.WARNING))
        {
            log(Level.WARNING,
                "registerScopeable",
                "Illegal attempt to register Scopeable",
                new Throwable());
        }
    }
}
```

```
return;
}
```

#59 - 05/06/2023 12:17 PM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

Maciej Zieniuk wrote:

I found this exception from the server's logs, that happens inside the passWrongDefinition. I am guessing they are the cause of the errors i am describing.
[...]

I see similar error. The offending code is:

[...]

I wonder if this is an error at all. The code emits warning and continues.

#60 - 05/06/2023 12:58 PM - Maciej Zieniuk

I don't think it's a coincidence that just after this test and exception, other tests are unable to run correctly.

On the tests.dataset.TestAddRelation test testValidTablesValidFieldsType, it fails because the qryHdl = relHdl:query no-error. was supposed to raise an error but it does not. From what i can see it's trying to parse this query's where condition where ttItem.itemName=ttCustomer.customerNum where the two columns are different types (character and numeric) and it should raise 223 error, but does not.

I found this line in annotatons.xml rules file early type-checking, line no 467, which says: NOTE: not 100% compatible with P4GL. In P4GL: the character type fields cause error 223, which also prints to stdout.log:

```
Additional info: ch1type=text ch2type=numbertype
= [EQUALS]:12884901933 @3:62
  decimal [FUNC_DEC]:12884901945 @0:0
    (oldtype=1250, builtin=true, returnsunknown=true)
  ttItem.itemName [FIELD_CHAR]:12884901934 @3:47
    (oldtype=6, schemaname=ttitem_tt_dtt2.itemname, bufname=ttitem, dbname=, recordtype=14, name=ttItem.itemName, type=411)
  ttCustomer.customerNum [FIELD_INT]:12884901935 @3:63
    (oldtype=6, schemaname=ttcustomer_tt_dttl.customernum, bufname=ttcustomer, dbname=, recordtype=14, name=ttCustomer.customerNum, type=419)
```

I wonder whether that's intentional incompatibility or just something, that maybe should not be enabled for anything database / query related ?

#61 - 05/06/2023 01:25 PM - Vladimir Tsichevski

Maciej Zieniuk wrote:

I don't think it's a coincidence that just after this test and exception, other tests are unable to run correctly.

Is there an ordinary 4gl procedure containing the same test? I want to know if whether the results in FWD test engine differ.

#62 - 05/06/2023 01:28 PM - Maciej Zieniuk

I did run TestOutputDynamicDsInDynamicProc against 4GL 11.6 and it was green (all test methods passing)

#63 - 05/06/2023 01:34 PM - Maciej Zieniuk

Also just pushed a commit to 4514a p2j branch that fixes the TestDataSourceModified test - error 4083 as warning:

```
=== modified file 'src/com/goldencode/p2j/persist/DataSet.java'
--- old/src/com/goldencode/p2j/persist/DataSet.java      2023-05-02 20:12:31 +0000
+++ new/src/com/goldencode/p2j/persist/DataSet.java      2023-05-06 17:26:04 +0000
@@ -2151,7 +2151,7 @@
     {
         if (mod == null || mod.isUnknown())
         {
-            unableToAssignUnknown("DATA-SOURCE-MODIFIED", "DATASET widget");
+            ErrorManager.recordOrShowError(4083, false, "DATA-SOURCE-MODIFIED", "DATASET widget");
             return;
         }
     }
 }
```

#64 - 05/06/2023 01:40 PM - Vladimir Tsichevski

Maciej Zieniuk wrote:

I did run TestOutputDynamicDsInDynamicProc against 4GL 11.6 and it was green (all test methods passing)

I meant the same unit test code under test implemented as ordinary 4gl procedure, so we could convert and run both variants in FWD and see whether there is a difference resulted in [#4514-57](#).

#65 - 05/07/2023 06:00 AM - Maciej Zieniuk

I can create it. So essentially, we take out the content of the passWrongDefinition test as a standalone procedure .p file with all of the things in the setup, cleanup and variables, correct ?

Another error i observed is during tests.dataset.TestFill tests execution, the error: SEVERE: {main} ** "resources/logo.png" was not found. (293). The file exists and it's referenced in support/temptable/populate/ttCustomer.p line copy-lob from file 'resources/logo.png' to bufferCustomer::logo no-error.

#66 - 05/07/2023 08:08 AM - Maciej Zieniuk

Another commit i have added to 4514a p2j branch. That fixes the TestFill test.

```
=== modified file 'src/com/goldencode/p2j/persist/DataSource.java'
--- old/src/com/goldencode/p2j/persist/DataSource.java 2023-04-06 02:34:19 +0000
+++ new/src/com/goldencode/p2j/persist/DataSource.java 2023-05-07 12:06:06 +0000
@@ -525,6 +525,18 @@
     @Override
     public void setFillWhereString(String pref)
     {
+       if (datasetBuffer == null
+         && (pref != null && !pref.trim().equalsIgnoreCase("where")))
+       {
+         EntityManager.recordOrShowError(12786);
+         // FILL-WHERE-STRING cannot be assigned unless the DATA-SOURCE is attached to some dataset buffer
+         // by a prior ATTACH-DATA-SOURCE. (12786)
+
+         EntityManager.recordOrThrowError(3131, "FILL-WHERE-STRING", "", "DATA-SOURCE");
+         // Unable to set attribute FILL-WHERE-STRING in widget of type DATA-SOURCE (3131)
+         return;
+       }
+
+       // NOTE: IN P4GL, the syntax of the string is NOT checked (at least at this moment)!
+       this._fillWhereString = pref;
    }
@@ -542,7 +554,7 @@
    {
      if (pref == null || pref.isUnknown())
      {
-       unableToAssignUnknown("FILL-WHERE-STRING", "DATA-SOURCE widget");
+       EntityManager.recordOrShowError(4083, false, "FILL-WHERE-STRING", "DATA-SOURCE widget");
        return;
      }
    }
  }
```

As in previous commit, i am unsure on whether the unableToAssignUnknown should in general result in warning, instead of error. Do you think i should change it everywhere (within the unableToAssignUnknown) to result in warning instead ?

#67 - 05/08/2023 09:41 AM - Maciej Zieniuk

I don't fully understand this test `tests.dataset.TestFillMode` method with `PkCorrectBufferAssign`, specifically this part:

```
// with Pk throw error
define variable errNums3 as integer extent 3 no-undo.
define variable errMsg3 as character extent 3 no-undo.

ttOrderCust:default-buffer-handle:fill-mode = 'append'.
dsOrder:fill no-error.
assign
  errNums3[1] = 132
  errNums3[2] = 132
  errNums3[3] = 11878.

assign
  errMsg3[1] = '~*~* ttOrderCustomer already exists with 1. (132)'
  errMsg3[2] = '~*~* ttOrderCustomer already exists with 1. (132)'
  errMsg3[3] = 'FILL got error -1 creating record ttOrderCustomer. (11878)'.
support.test.AssertExt:Errors(errNums3, errMsg3, true).
```

Why would it fail twice with `ttOrderCustomer` already exists with 1. (132) ?

#68 - 05/08/2023 02:16 PM - Ovidiu Maxiniuc

Vladimir Tsichevski wrote:

Maciej Zieniuk wrote:

I found this exception from the server's logs, that happens inside the `passWrongDefinition`. I am guessing they are the cause of the errors I am describing.
[...]

I see similar error. The offending code is:
[...]

The problem is, most likely, not in the stack. From my educated guess, the problem occurred before that and the internal data at the moment when `registerScopeable()` (or `registerPendingScopeable()`) is invoked is already broken. Are there any (error/warning) messages logged before `TransactionManager.registerScopeable:WARNING` ?

#69 - 05/08/2023 03:33 PM - Ovidiu Maxiniuc

Vladimir Tsichevski wrote:

Another conversion issue: dataset.TestAcceptChanges legacy class converts to:
[...]

Note the extra **semicolon** in the block body. Eclipse considers this a problem, but more important, I suspect there should be some real code here.

You can investigate the ast/jast files to trace back to original ABL code. I suspect there is a do/end block with empty content OR maybe a 4GL feature not fully supported in FWD (the lexer/parser identify it but actually not handled during the rest of the conversion).

#70 - 05/08/2023 03:42 PM - Vladimir Tsichevski

- File *TestOutputDynamicDsInDynamicProc.p* added

Ovidiu Maxiniuc wrote:

Vladimir Tsichevski wrote:

Maciej Zieniuk wrote:

I found this exception from the server's logs, that happens inside the passWrongDefinition. I am guessing they are the cause of the errors i am describing.
[...]

I see similar error. The offending code is:
[...]

The problem is, most likely, not in the stack. From my educated guess, the problem occurred before that and the internal data at the moment when registerScopeable() (or registerPendingScopeable()) is invoked is already broken. Are there any (error/warning) messages logged before TransactionManager.registerScopeable:WARNING ?

I've implemented the PassWrongDefinition test as an ordinary procedure (see the TestOutputDynamicDsInDynamicProc.p attached), which I ran as an ordinary FWD client.

The behavior is similar to that of the corresponding ABLUnit test in that in both cases two errors are displayed in message boxes, which are not present in ErrorManager, so the test fails in both cases due to the same reason.

Is that the expected behavior?

#71 - 05/08/2023 03:59 PM - Ovidiu Maxiniuc

Maciej Zieniuk wrote:

I don't think it's a coincidence that just after this test and exception, other tests are unable to run correctly.

On the tests.dataset.TestAddRelation test testValidTablesValidFieldsType, it fails because the qryHdl = relHdl:query no-error. was supposed to raise an error but it does not. From what i can see it's trying to parse this query's where condition where ttItem.itemName=ttCustomer.customerNum where the two columns are different types (character and numeric) and it should raise 223 error, but does not.

I found this line in annotations.xml rules file early type-checking, line no 467, which says: NOTE: not 100% compatible with P4GL. In P4GL: the character type fields cause error 223, which also prints to stdout.log:
[...]

I wonder whether that's intentional incompatibility or just something, that maybe should not be enabled for anything database / query related ?

Until the runtime (dynamic) conversion was implemented in FWD, the static conversion assumed the input 4GL code was correct. As result the syntactic errors were reported directly to console, allowing a quick feedback.

When the runtime conversion was implemented, this error (and a few others) was recognised and a new TRPL method (compileError()) was added to report this event to calling method as an exception (CompileException). This exception carries the information necessary for composing an error message as similar as possible as in 4GL. Please see the rules/annotations/annotations.xml:542. This is the place the event is reported only for runtime conversion.

But, there are always exceptions so unmatching data type is not always reported. At that time we discovered that the dynamic conversion in 4GL allows a bit of implicit conversion. In that case, as described in comment you sound, it was about 'concatenation' of a numeric data with a character. The log was added intentionally for detecting other similar cases in (future) customer code. It worked, and the cases at hand are two different numeric types. They just need treated as for previous case, using wrapping injection.

#72 - 05/08/2023 04:11 PM - Ovidiu Maxiniuc

Maciej Zieniuk wrote:

Another commit i have added to 4514a p2j branch. That fixes the TestFill test.

[...]

As in previous commit, i am unsure on whether the unableToAssignUnknown should in general result in warning, instead of error. Do you think i should change it everywhere (within the unableToAssignUnknown) to result in warning instead ?

This must be tested.

Generally speaking, yes, all methods will return a value (yes for success) to let the caller know if the respective operation was successful instead of raising a true error condition. It seems normal that, in the these cases to use ErrorManager.recordOrShowError(n, false, .. instead of

unableToAssignUnknown() which declares the event as an error.

#73 - 05/08/2023 04:16 PM - Ovidiu Maxiniuc

Maciej Zieniuk wrote:

I don't fully understand this test tests.dataset.TestFillMode method withPkCorrectBufferAssign, specifically this part:
[...]

Why would it fail twice with ttOrderCustomer already exists with 1. (132) ?

IIRC, this is how 4GL works. Probably there are two values in the source which would break the unique index. Another less probable cause may be two different indices which are broken by appending the same record to the table.
Is FWD passing this testcase?

#74 - 05/08/2023 05:59 PM - Vladimir Tsichevski

- File *persistprocwitherror.p* added

- File *testerrorinpersistentproc.p* added

There is something wrong with error propagation in FWD when a persistent procedure is concerned.

The procedure *persistprocwitherror.p* is executed persistent from *testerrorinpersistentproc.p*, then an internal procedure is called as no-error. The internal procedure raises an error, which sets the error-status:ERROR to true.

In FWD the error-status:ERROR remains false instead.

#75 - 05/09/2023 07:25 AM - Maciej Zieniuk

Vladimir Tsichevski wrote:

There is something wrong with error propagation in FWD when a persistent procedure is concerned.

The procedure *persistprocwitherror.p* is executed persistent from *testerrorinpersistentproc.p*, then an internal procedure is called as no-error. The internal procedure raises an error, which sets the error-status:ERROR to true.

In FWD the error-status:ERROR remains false instead.

It works fine on the 3827b and 4514a (based on top of 3827b) FWD branches, where I think the aim of the 3827b is to fix that issue - so it's expected it does not work in trunk.

Ovidiu Maxiniuc wrote:

Maciej Zieniuk wrote:

I don't think it's a coincidence that just after this test and exception, other tests are unable to run correctly.

On the tests.dataset.TestAddRelation test testValidTablesValidFieldsType, it fails because the qryHdl = relHdl:query no-error. was supposed to raise an error but it does not. From what i can see it's trying to parse this query's where condition where ttlItem.itemName=ttCustomer.customerNum where the two columns are different types (character and numeric) and it should raise 223 error, but does not.

I found this line in annotations.xml rules file early type-checking, line no 467, which says: NOTE: not 100% compatible with P4GL. In P4GL: the character type fields cause error 223, which also prints to stdout.log:
[...]

I wonder whether that's intentional incompatibility or just something, that maybe should not be enabled for anything database / query related ?

Until the runtime (dynamic) conversion was implemented in FWD, the static conversion assumed the input 4GL code was correct. As result the syntactic errors were reported directly to console, allowing a quick feedback.

When the runtime conversion was implemented, this error (and a few others) was recognised and a new TRPL method (compileError()) was added to report this event to calling method as an exception (CompileException). This exception carries the information necessary for composing an error message as similar as possible as in 4GL. Please see the rules/annotations/annotations.xml:542. This is the place the event is reported only for runtime conversion.

But, there are always exceptions so unmatching data type is not always reported. At that time we discovered that the dynamic conversion in 4GL allows a bit of implicit conversion. In that case, as described in comment you sound, it was about 'concatenation' of a numeric data with a character. The log was added intentionally for detecting other similar cases in (future) customer code. It worked, and the cases at hand are two different numeric types. They just need treated as for previous case, using wrapping injection.

The problem is that there is the expectancy of the 223 error, but it does not happen, because of that rule in annotations.xml, which basically converts the character/text to the numeric/long type. The rule to action as 223 error is only when wrapperFn==null, which here results in false, because of the rule ch1type=="text" and ch2type=="numbertype" which action is to provide the wrapperFn, so it's not null.

If we want that text -> number logic to stay as it is, then that test will stay broken. I don't have any other idea on how to fix it, other than put some special handling during the runtime conversion of the where condition for the query to be strict about the conversion of the types.

#77 - 05/09/2023 09:02 AM - Marian Edu

Ovidiu Maxiniuc wrote:

IIRC, this is how 4GL works. Probably there are two values in the source which would break the unique index. Another less probable cause may be two different indices which are broken by appending the same record to the table.

This has nothing to do with number of records in that temp-table nor unique constraints being violated, it is in fact because of the NO-UNDO option set in the temp-table. Basically what happens the records is created and assignment fail because unique index violation but given the tem-table is defined as NO-UNDO the record remains in the buffer and another 132 error might be thrown in different situations - for instance when the buffer goes out of scope and another save on the buffer is attempted at that point.

Some details on those KB entries: [[<https://community.progress.com/s/article/P185553>]], [[<https://community.progress.com/s/article/P103161>]]

#78 - 05/09/2023 10:16 AM - Maciej Zieniuk

Vladimir Tsichevski wrote:

Ovidiu Maxiniuc wrote:

Vladimir Tsichevski wrote:

Maciej Zieniuk wrote:

I found this exception from the server's logs, that happens inside the passWrongDefinition. I am guessing they are the cause of the errors i am describing.
[...]

I see similar error. The offending code is:
[...]

The problem is, most likely, not in the stack. From my educated guess, the problem occurred before that and the internal data at the moment when registerScopeable() (or registerPendingScopeable()) is invoked is already broken. Are there any (error/warning) messages logged before TransactionManager.registerScopeable:WARNING ?

I've implemented the PassWrongDefinition test as an ordinary procedure (see the TestOutputDynamicDsInDynamicProc.p attached), which I ran as an ordinary FWD client.

The behavior is similar to that of the corresponding ABLUnit test in that in both cases two errors are displayed in message boxes, which are not present in ErrorManager, so the test fails in both cases due to the same reason.

Is that the expected behavior?

It's not about PassWrongDefinition failing, but about PassWrongDefinition affecting the entire runtime of FWD, affecting other tests. Basically, run PassWrongDefinition in front of other tests versus don't run the PassWrongDefinition at all and you end up with different results for the tests run. It should not happen.

That error i pointed out, must be affecting the scope, essentially breaking the entire FWD runtime to some degree.

Consider attached variation of the procedure.

The expectancy is that you can create valid dataset by calling run support/dataset/create/dsMaster.p(false, false, false, output dsMaster). and assessing it with valid-handle (dsMaster), as it is the case for TestOutputDynamicDsInDynamicProc.cls in setUp method.

In my variation, the create the dataset is called once more at the end, in which case the handle is invalid, but should be to my understanding. When running tests with FWD Test Engine (ABLUnit), this failure persists across different test classes, methods after the PassWrongDefinition failure.

#79 - 05/09/2023 10:19 AM - Maciej Zieniuk

- File TestOutputDynamicDsInDynamicProcInvalidHandle.p added

attaching the procedure i mentioned :)

#80 - 05/09/2023 06:46 PM - Ovidiu Maxiniuc

Marian Edu wrote:

Ovidiu Maxiniuc wrote:

IIRC, this is how 4GL works. Probably there are two values in the source which would break the unique index. Another less probable cause may be two different indices which are broken by appending the same record to the table.

This has nothing to do with number of records in that temp-table nor unique constraints being violated, it is in fact because of the NO-UNDO option set in the temp-table. Basically what happens the records is created and assignment fail because unique index violation but given the tem-table is defined as NO-UNDO the record remains in the buffer and another 132 error might be thrown in different situations - for instance when the buffer goes out of scope and another save on the buffer is attempted at that point.

Some details on those KB entries: [[<https://community.progress.com/s/article/P185553>]], [[<https://community.progress.com/s/article/P103161>]]

Thank you. Seems logical. I was not remembering correctly :(.

I was thinking of some other behaviour with similar result (also printing multiple identical errors, also related to datasets).

#81 - 05/10/2023 02:59 AM - Marian Edu

Ovidiu Maxiniuc wrote:

Thank you. Seems logical. I was not remembering correctly :(.

I was thinking of some other behaviour with similar result (also printing multiple identical errors, also related to datasets).

I've added a test for this in testcases under tests/table/crud/create/TestUniqueKeyViolationError.cls just so you don't forget about it ;)

However, imho the dataset is probably the most complex object in 4GL and testing this one first looks a bit odd. There are a bunch of other

components used in dataset tests: table, buffer, data relation, data source, query... we're not testing those components inside the dataset tests, we just expect them to work like in 4GL. So when a test fails because setUp didn't work or because one of those components are failing the issue is not related to the dataset functionality per-se.

Anyway, just my X cents (do we adjust that with inflation???).

#82 - 05/12/2023 05:52 PM - Vladimir Tsichevski

Maciej Zieniuk wrote:

attaching the procedure i mentioned :)

This procedure (the assertion in the end of it) fails when being run with 3827b. I suppose this is expected?

If so, which patch should I apply to make it work?

Thanks,
Vladimir

#83 - 05/15/2023 05:58 AM - Maciej Zieniuk

Vladimir Tsichevski wrote:

Maciej Zieniuk wrote:

attaching the procedure i mentioned :)

This procedure (the assertion in the end of it) fails when being run with 3827b. I suppose this is expected?

If so, which patch should I apply to make it work?

Thanks,
Vladimir

Let's clarify end of this procedure:

```
//@Test.  
// method public void PassWrongDefinition( ):  
  
//pass wrong definition dataset-handle by-value  
create temp-table ttInit.  
ttInit:add-new-field ("num", "integer").  
ttInit:temp-table-prepare ('init').  
create dataset dsInit.  
dsInit:add-buffer (ttInit).  
run dsAsOutput in hproc( output dataset-handle dsInit by-value, input 3, output dsReturnId, output ttCustomerI  
temsReturned ) no-error.  
assign
```

```

errNums2[1] = 5363
errNums2[2] = 9030.

assign
    errMsg2[1] = 'The caller~'s temp-table parameter init does not match to the target temp-table ttCustomer.
(5363)'
    errMsg2[2] = 'try switching temp-table method add-fields-from for create-like. (9030)'.
// Lets ignore the errors
// support.test.AssertExt:Errors(errNums2, errMsg2, true).

// @Teardown.
HandleFactory:deleteDataObjects().

// This should result in valid handle, but does not
run support/dataset/create/dsMaster.p(false, false, false, output dsMaster2).
MESSAGE valid-handle(dsMaster2).

```

The commented out line `// support.test.AssertExt:Errors(errNums2, errMsg2, true)`. actually fails on that assertion. I commented it for this reason. The MESSAGE `valid-handle(dsMaster2)`. should show true but shows false.

Regardless of the failures, those two lines at the end should show true. It is independent of everything happening in the test and it should not mater where you run it.

```

run support/dataset/create/dsMaster.p(false, false, false, output dsMaster2).
MESSAGE valid-handle(dsMaster2).

```

It shows false and it is to do with the scoping, assignment of the value returned inside the `dsMaster.p()` to the `dsMaster2`, which does not happen. Ovidiu suggested this, but i was not able to validate it, nor confirm any other errors or warnings

The problem is, most likely, not in the stack. From my educated guess, the problem occurred before that and the internal data at the moment when `registerScopeable()` (or `registerPendingScopeable()`) is invoked is already broken. Are there any (error/warning) messages logged before `TransactionManager.registerScopeable:WARNING` ?

#84 - 05/15/2023 12:29 PM - Maciej Zieniuk

Marian Edu wrote:

Maciej Zieniuk wrote:

The tests/dataset/TestXmlNodeType.cls have a line during the setup, which runs some other procedure run dataset/create/dsMaster.p. I experimented with different options, read the documentation and could not find anything where i could get it working. It's converting, but the tests during runtime immediately fails, that the dataset/create/dsMaster.p is nowhere to be found. The only way to fix this is to add the full path (support/) to the run run support/dataset/create/dsMaster.p.
Any ideas on this ?

Yes, for whatever reason we had 'support' and 'tests' folders added to the PROPATH and that is probably not the case for FWD. I think we should better fix that one right now so we only keep the project's ROOT in PROPATH, we will 'refactor' current tests for this and let you know when available on testcases.

On the other failures, there are plenty of tests that depends on the dataset/stdFiles or dataset/outFiles files, but they are nowhere to be found. Example dataset/outFiles/dynamic/xml/memptr/dsMaster_dsMaster_element_no_no_UTF-8_default_no_no_no_no_no.xml in file tests/dataset/TestReadXmlFileNoPkReadGenerated.cls.

Yes, those are files we use for comparing results - we normally generate them but it looks like the 'generators' were also missing, anyway I will add those back in new location tests\dataset.

Am i supposed to copy the tests/dataset/support files into deploy/client to be able to run any JSON or XML dependent tests, like TestReadJsonFile.cls ?

From what i understood from https://proj.goldencode.com/projects/p2j/wiki/4GL_Unit_Testing, we run the tests from deploy/client, so it's not like it's expected for those files to be in there ?

#85 - 05/16/2023 10:42 PM - Ovidiu Maxiniuc

When running any converted 4GL application, the resource files can be either in jar, but usually, like in this case, the file path is computed starting from the client level. Even if the application actually runs on server (the server application is started from other location or even from other PC) the streams and memptrs are, in fact bound to client (and the data is transferred from client to server back and forth).

So the answer is yes. Assuming you launch the client from deploy/client and the paths of the files to be opened start with tests/dataset/support, then you need to copy (or link) the support resources to deploy/client folder.

#86 - 05/17/2023 06:50 AM - Greg Shah

We should change the code to make it operate more naturally. I prefer not to copy lots of files around when running the test code.

#87 - 05/17/2023 07:18 AM - Marian Edu

Greg Shah wrote:

We should change the code to make it operate more naturally. I prefer not to copy lots of files around when running the test code.

I guess we can use SEARCH if PROPATH resolution works for those files instead of using relative path to testcases root folder, is that something that can help?

#88 - 05/17/2023 07:36 AM - Greg Shah

SEARCH can work. We also could provide some kind of configuration value that could be the "root" path for relative filename access.

#89 - 05/17/2023 07:43 AM - Marian Edu

Greg Shah wrote:

SEARCH can work. We also could provide some kind of configuration value that could be the "root" path for relative filename access.

OK, we will change the code for the tests that needs to load 'static' resources to use SEARCH as needed.

#90 - 05/17/2023 08:29 AM - Maciej Zieniuk

I have analyzed the failure in the tests/dataset/TestReadXmlFileNoPkReadGenerated.cls test createFilesForDsmaster method. It fails during the schema load from xml file, when preparing the temp-table, specifically tests/dataset/support/stdFiles/dynamic/xml/file/dsMaster_dsMaster_element_no_no_UTF-8_default_yes_no_no_no.xml file. When the end of </schema> is reached for ttCustomer temp-table, temp-table-prepare (TempTableBuilder.java file tempTablePrepare method), it checks whether the temp-table is in prepared state and it fails because it detects that it is, while the expectancy is that it should not.

The 4GL 11.7 docs says that temp-table is in prepared state only after successfully execution of the temp-table-prepare. Digging into the TempTableBuilder implementation (which to my understanding is the dynamic temp-table implementation in FWD), the prepared boolean check is currently implemented as name is not null, which is wrong.

Further reading the documentation, the name is a parameter what can be set or changed at any time and it have nothing to do with the prepared state.

I think we should implement it correctly, basically handle the clear, unprepared or prepared change of states in the implementation correctly for the dynamic temp-tables (might also apply to static, not sure yet). We probably want it as a separate # issue ? I can work on it, That's assuming it understand the problem correctly, any additional comments on this ?

#91 - 05/17/2023 08:32 AM - Maciej Zieniuk

Adding to the name, it looks like during the xml file load, the name is set on the temp-table when it encounters this xml element: `<xsd:element name="ttCustomer" minOccurs="0" maxOccurs="unbounded">`, which is the reason why it have the name set. This does not change the fact, that name have nothing to do with the prepared state to my understanding.

#92 - 05/17/2023 08:52 AM - Greg Shah

the name is a parameter what can be set or changed at any time and it have nothing to do with the prepared state

I agree, this seems like the wrong implementation. Please add a separate flag for it. Put the change in your 4514a branch.

If Eric or Ovidiu disagree, they can say so.

#93 - 05/17/2023 11:28 AM - Maciej Zieniuk

The prepared flag did not fix the test, mostly because the temp-tables are prepared already when run `support/dataset/create/dsMaster.p` is called (temp-table-prepare is called)

I will need to do further testing on 4GL to understand how it does not fail like in FWD.

#94 - 05/17/2023 08:51 PM - Ovidiu Maxiniuc

Maciej Zieniuk wrote:

I have analyzed the failure in the `tests/dataset/TestReadXmlFileNoPkReadGenerated.cls` test `createFilesForDsMaster` method. [...]

What was the revision you used for testing? Recently (r14566 in trunk), Igor committed a rather big update to JSON/XML/XMLSchema (de)serialization.

The 4GL 11.7 docs says that temp-table is in prepared state only after successfully execution of the temp-table-prepare. Digging into the TempTableBuilder implementation (which to my understanding is the dynamic temp-table implementation in FWD), the prepared boolean check is currently implemented as name is not null, which is wrong.

Two comments: we encountered often enough cases where the reference manual of OpenEdge 4GL is not always accurate and we created a habit of testing to confirm the issues. Secondly, I think r14566 adds some changes in this area. Please confirm you used a FWD revision which includes this commit.

Further reading the documentation, the name is a parameter what can be set or changed at any time and it have nothing to do with the prepared state.

I think we should implement it correctly, basically handle the clear, unprepared or prepared change of states in the implementation correctly for the dynamic temp-tables (might also apply to static, not sure yet). We probably want it as a separate # issue ? I can work on it, That's assuming it understand the problem correctly, any additional comments on this ?

I am pretty sure the name of static temp-tables cannot be set/changed dynamically.

In the case of dynamic temp-tables, there is error 9040 which will notify you that "TEMP-TABLE must be PREPARED before referencing NAME...". After the table was prepared, the name can be changed. I live with the sensation that this works correctly in FWD. Please create specific testcases, run them against FWD and report which ones are behaving differently from 4GL.

#95 - 05/18/2023 05:47 AM - Maciej Zieniuk

I just rebased 4514a on top of newest 3827b (which includes some fixes for ABLUnit that i need), which is based on top of the 14566 revision from trunk, so including the JSON/XML (de)serialization fixes.
It is still failing in the same way as before.

#96 - 05/18/2023 06:24 AM - Marian Edu

Ovidiu Maxiniuc wrote:

I am pretty sure the name of static temp-tables cannot be set/changed dynamically.

In the case of dynamic temp-tables, there is error 9040 which will notify you that "TEMP-TABLE must be PREPARED before referencing NAME...". After the table was prepared, the name can be changed. I live with the sensation that this works correctly in FWD. Please create specific testcases, run them against FWD and report which ones are behaving differently from 4GL.

I can confirm that, for static there is always an error, for dynamic it can only be set/get after it was prepared - temp-table-prepare sets the initial value for name.

#97 - 05/18/2023 08:16 AM - Maciej Zieniuk

Yeah i agree, set name for dynamic temp-table is equivalent to prepared state - i will revert my change from the branch.

I have found a different issue with the FWD code, that does not match 4GL (both in documentation and testing). Basically you can call READ-XML on a DataSet or temp-table with a schema multiple times with rows, even if the temp-table contains records or is in prepared state already. The important piece is the last parameter of the read-xml, which is schema validation. By default 'LOOSE', but can also be 'STRICT' or 'IGNORE'. In all of the cases, READ-XML will not modify or clear already prepared temp-table in any way, schema wise. For FWD, during the XML file reading, we end up calling tempTablePrepare, which i believe is incorrect (XMLImport.java method processDsBuffers method), as 4GL does not behave in that way. In 4GL it will only prepare and load the XML schema if the temp-table is in CLEAR state and then PREPARE it, otherwise it needs to be in PREPARED state and falls down to validation of the schema between the XML schema and temp-table schema (the result is dependent on the read-xml schema validation parameter), the docs says:

```
If the ProDataSet or temp-table object does not have a schema (that is, the object is dynamic and in the CLEAR state), the AVM creates the schema from either the XML Schema file specified in schema-location, or the XML Schema defined or referenced in the XML document. If a dynamic temp-table is not in the PREPARED or CLEAR state, the method generates an error and returns FALSE.
```

```
If the ProDataSet or temp-table object already has a schema (that is, the object is static, or the temp-tables are in the PREPARED state), the AVM verifies any XML Schema specified by schema-location, or defined or referenced in the XML document, against the object's schema, unless the verify-schema-mode is "IGNORE".
```

I think the fix here is not to call tempTablePrepare during the schema load from the XML file (in XMLImport), but only if the temp-table is in the prepared state - so something like

```
if (!ttb._prepared())
{
    ttb.tempTablePrepare(e.getKey());
}
```

The schema validation code is already in place after the processDsBuffers method in XMLImport class.

This change fixed 4 tests and i think it make sense, any comments on this ?

#98 - 05/18/2023 08:28 AM - Greg Shah

This change fixed 4 tests and i think it make sense, any comments on this ?

Igor: I think this is for you.

#99 - 05/18/2023 09:08 AM - Maciej Zieniuk

Just wondering that maybe for better clarity, we only prepare temp-table, when the temp-table is in the CLEAR state ? You can't execute READ-XML- when temp-table is in UNPREPARED state anyway:

```
if (ttb._clear())
{
    ttb.tempTablePrepare(e.getKey());
}
```

#100 - 05/18/2023 09:12 AM - Igor Skornyakov

Maciej Zieniuk wrote:

Just wondering that maybe for better clarity, we only prepare temp-table, when the temp-table is in the CLEAR state ? You can't execute READ-XML- when temp-table is in UNPREPARED state anyway:
[...]

Actually, we can. In this case the table structure is inferred from the XML file. I'm, working on this. See [#7247](#).

#101 - 05/18/2023 09:36 AM - Maciej Zieniuk

There is something else, that caught my attention. In the tests/dataset/TestReadXmlFileNoPkReadGenerated.cls test, during the setup, a warning/error is thrown BUFFER-FIELD customHandle was not found in buffer ttCustomer. (7351), that fails in the run support/temptable/populate/ttCustomer.p(table-handle ttOrderCust by-reference). line.
That field is not there, because when the run support/dataset/create/dsMaster.p (false, false, false, output dsMasterRef). and then run support/temptable/create/ttCustomer.p("", noDefaults, output tableHdl). is called with noDefaults being false.

That error propagates to the test, the same for FWD and 4GL.
Shouldn't we introduce the noDefaults in the populate procedures ?
Or maybe we did something wrong with the create procedures and the noDefaults should mean something else ?

Please help, i can't run the tests when stuff breaks in the setup phase.

#102 - 05/26/2023 02:11 PM - Eric Faulhaber

- Related to Bug #6475: double-check and fix any read/write-xml/json issues at table and dataset added

#103 - 05/29/2023 01:35 PM - Maciej Zieniuk

The tests/dataset/TestReadXmlFile.cls method invalidFileIncompleteTags should fail with 13064 instead of 13055 on the 2nd error. When i create a scenario in 4GL with missing XML tags during the READ-XML, regardless of whether i am reading into the dataset or temp-table, the errors are always the same, first error 13035, second error 13064.

Looking into the FWD latest trunk, the XmlImport.java method importTable exception handling of XMLStreamException differs from the importDataset -> ... -> readRecord2 exception handling.

This does not look right, as looking into the readRecord2 method, it feels like regardless of the error recorded in the no-error mode (silent) during the xml record reading (in the try or catch), we end up with 13055 error. This also disrupts the differentDataType method in the same test file, in which we expect 13052 as 2nd error (although it's not thrown), but get 13055 regardless (so basically 3rd one)

#104 - 05/29/2023 01:59 PM - Maciej Zieniuk

For the Unable to convert XML to native data type for field '<field>' in table '<table>'. (13052) i am thinking about putting it inside of the setField in the XmlImport catch block, that if 76 error type is detected (or similar?), then it throws the 13052 ? Opinions ?

For context, in this scenario the ttCustomer temp-table field customerNum is expected to be integer / numeric type, but in the tests/dataset/support/stdFiles/dynamic/xml/dsMaster_no_pk_no_schema_diffrent_data_type.xml it's defined as string.

#105 - 05/29/2023 02:31 PM - Ovidiu Maxiniuc

Maciej,
IIRC, when I added dataset and schema reading support I handled the errors as they occurred from my testcases. Later, Marian added other testcases and I noticed some errors were reversed. Apparently the order in which the read data is validated was incorrect. I tried to fix that accordingly and I had the surprise that my initial tests were failing now. I was not able to get the reason or the cause for some specific error numbers. I guess these are those. However, the code evolved a bit and now we might have a better understanding of the data flow while reading the XML. You should work with Igor on this aspects as he added the last big improvements in this area.

#106 - 05/30/2023 07:36 AM - Maciej Zieniuk

Fine, but that does not change the fact that the xfer testcases fail in FWD and does not in 4GL 11.6. I will confirm the behavior for 11.7 next. The verifySchemaMode method in the TestReadXmlFile.cls test is failing in FWD during the strict mode read-xml. We expect two errors, 13091 and 13033 but none of them happens.

Again, in FWD code XmlImport.java we have three separate implementations for readTableSchema, one for read-xml for temp-table and one for dataset (not sure when we use the 3rd one). In case of the dataset readTableSchema implementation, those two errors are not there, while according to testcases (and 4GL test run) they should happen, so clearly to me there are edge cases testcases that did not get covered.

Igor, what's your take on this ?

#107 - 05/30/2023 07:42 AM - Igor Skornyakov

Maciej Zieniuk wrote:

Fine, but that does not change the fact that the xfer testcases fail in FWD and does not in 4GL 11.6. I will confirm the behavior for 11.7 next. The verifySchemaMode method in the TestReadXmlFile.cls test is failing in FWD during the strict mode read-xml. We expect two errors, 13091 and 13033 but none of them happens.

Again, in FWD code XmlImport.java we have three separate implementations for readTableSchema, one for read-xml for temp-table and one for dataset (not sure when we use the 3rd one). In case of the dataset readTableSchema implementation, those two errors are not there, while according to testcases (and 4GL test run) they should happen, so clearly to me there are edge cases testcases that did not get covered.

Igor, what's your take on this ?

Maciej,

So far my tests do not check this functionality. I will take a look at this after I'll finish with [#7247](#)

#108 - 05/30/2023 07:56 AM - Maciej Zieniuk

Noted. I will leave the TestReadXml* tests for now and continue with the rest of failures.

#109 - 05/30/2023 10:20 AM - Maciej Zieniuk

I have just committed a change to 4514a FWD branch, which fixes the tests/dataset/TestSerializeName.cls failure.

```
=== modified file 'src/com/goldencode/p2j/persist/DataSet.java'
--- old/src/com/goldencode/p2j/persist/DataSet.java      2023-05-26 11:55:47 +0000
+++ new/src/com/goldencode/p2j/persist/DataSet.java      2023-05-30 14:07:55 +0000
@@ -2570,11 +2570,7 @@
     @Override
     public void setSerializeName(String sName)
     {
-         if (sName != null)
-         {
-             serializeName = sName;
-         }
-         // no error is raised otherwise
+         serializeName = sName;
     }

     /**
```

#110 - 05/30/2023 11:24 AM - Ovidiu Maxiniuc

Not sure that is the right thing to do. I remember writing that intentionally. Here is the testcase:

```
define temp-table ttl
```

```
    field f1 as int
    field f2 as int.
buffer ttl:buffer-field("f1"):serialize-name="abc".
buffer ttl:buffer-field("f2"):serialize-name=?.
message buffer ttl:buffer-field("f1"):serialize-name
    buffer ttl:buffer-field("f2"):serialize-name.
```

It will print abc f2 because when the ? assignment is a no-op so the old serialize-name attribute is kept.

#111 - 05/30/2023 11:28 AM - Ovidiu Maxiniuc

Sorry, I used the fields, but this is also valid for datasets. Consider the following lines appended to my previous test:

```
define dataset ds1 for ttl.

message dataset ds1:serialize-name.
dataset ds1:serialize-name = ?.
message dataset ds1:serialize-name.
```

It will print ds1 twice.

#112 - 05/31/2023 03:17 AM - Maciej Zieniuk

override the NAME attribute and you will see that it differs. I did this testing in 4GL and FWD code in the getter of the SERIALIZE-NAME confirms that it defaults to NAME when it's not set (unknown). The 4GL docs says If you do not set either attribute, the AVM sets both to the ABL object name. (In context of SERIALIZE-NAME setter and XML-NODE-NAME attribute.) also confirms that it's override able to NAME when you pass unknown value.

On the other hand the docs suggests that it "sets", which would suggest that if you set an ? unknown value, it overrides the SERIALIZE-NAME, but from NAME attribute ?

I think this is a separate investigation to do, like whether it's a soft (current implementaton of FWD getter) or hard reference (by value) to the NAME attribute's value and so on.

#113 - 06/01/2023 05:03 AM - Maciej Zieniuk

For the tests/dataset/TestSetCallback.cls test wrongRoutineNameClass and wrightRoutineNameButWrongMethodDefinitionAsInputParamsClass method failures.

In both tests, we expect 13452 error, but we firstly get the 14457 and then 13452. According to 4GL, the 14457 should not be recorded, thrown at all. The 14457 is error is cached in DataSet.java method invokeCallback try-catch block and then 13452 is raised. I cannot find any reference anywhere regarding the 14457 error, so i guess it came from testing ? On the other hand, if in this test it does not occur, then we probably do not want to record the 14457 error at all ?

Opinions ?

#114 - 06/01/2023 06:12 PM - Ovidiu Maxiniuc

What do you mean by "we probably do not want to record the 14457 error at all?".

This is a normal error condition when 4GL found a method with the name specified (for callback, in this case) but cannot invoke it because the signature is not the one it expects for properly passing the required parameters.

The dataset 'blindly' invokes the method and hopes for the best, allowing the natural routine to be resolved by ControlFlowOps. Alternatively, we could expose the resolving feature of the CFO and:

- if the method with matching signature is detected, *invoke it*
- otherwise report error 13452 directly.

Although this is probably doable, I think both the resolve process and *invoke it* are deeply embedded in CFO logic and might not come with low risk.

Maciej, is this what you mean?

Constantin, can we split the invocation in CFO in two stages as described above?

#115 - 06/02/2023 04:21 AM - Maciej Zieniuk

Hi Ovidiu, yes that will work. Thanks

#116 - 06/07/2023 08:24 AM - Maciej Zieniuk

The test failure tests/dataset/TestTopNavQuery.cls method TestSetInvalid, throws 3 errors instead of 2.

The unexpected error is 7361, which in FWD is thrown in DataSet.java method _setTopBufferQuery, when the QueryWrapper.bufferHandleNative method is called.

The QueryWrapper.java method bufferHandleNative is simply calling getBufferByIndex(1).

I did some testing in 4GL (11.6 so far) and the 7361 is thrown during the get-buffer-handle only when the query handle have any buffers (i.e. via set-buffer or add-buffer).

Also the 7361 is thrown as error, but it is a warning at the moment (the TestGetBufferHandle.cls confirm this behaviour and manual 4GL testing):

Therefore i propose a fix in FWD code in QueryWrapper.java, to be as follows:

```
=== modified file 'src/com/goldencode/p2j/persist/QueryWrapper.java'
--- old/src/com/goldencode/p2j/persist/QueryWrapper.java      2023-05-12 10:05:12 +0000
+++ new/src/com/goldencode/p2j/persist/QueryWrapper.java      2023-06-07 10:49:32 +0000
@@ -3689,6 +3689,11 @@
     */
     public handle bufferHandle(character bufferName)
     {
+       if (buffers.isEmpty())
+       {
+         return new handle();
+       }
+
       for (int i = 0; i < buffers.size(); i++)
       {
         Buffer buf = buffers.get(i);
@@ -3704,7 +3709,7 @@
         errMsg = String.format(errMsg, numBuffers().intValue(), qname);

-       ErrorManager.recordOrShowError(7361, errMsg, false, false, false);
+       ErrorManager.recordOrShowError(7361, errMsg, false, false);

       return new handle();
     }
@@ -3756,6 +3761,11 @@
     */
```

```

    public Buffer getBufferByIndex(int bufferSequenceNumber)
    {
+       if (buffers.isEmpty())
+       {
+           return null;
+       }
+
        if (bufferSequenceNumber >= 1 && bufferSequenceNumber <= buffers.size())
        {
            return buffers.get(bufferSequenceNumber - 1);
@@ -3765,7 +3775,7 @@
            errMsg = String.format(errMsg, numBuffers().intValue(), qname);

-       ErrorManager.recordOrShowError(7361, errMsg, false, false, false);
+       ErrorManager.recordOrShowError(7361, errMsg, false, false);

        return null;
    }

```

#117 - 06/09/2023 06:16 AM - Maciej Zieniuk

I have hard time implementing the 302 and 463 errors in FWD, as expected by 4GL in case of tests/dataset/TestWriteXmlFile.cls failures. Those two errors happens when the WRITE-XML('file', '<LOCATION>') is called, provided an location this is a directory, either existing or not (in case of non-existing, the location must end with a path separator like / or \ for windows).

In the FWD code, the location needs to be accessible by the client, so when a test is executed on the server, the XmlExport.write is called, then the Stream is opened with target.getStream() which at the moment throws ErrorConditionException for 302 or 463 errors. On the client side, StreamDaemon.openFileStream is executed, then new FileStream is constructed, which fails with FileNotFoundException exception, but goes into the ErrorManager.recordOrThrowError path and re-throws DeferredLegacyErrorException and returns with absolutely no information to the server about the cause, as the error code nor the original exception is not recorded anywhere to be available from the server point of view.

Is this intended that server does not see any information about the error when the error is thrown in the client ?

Also I am struggling to guess what can be done in this case, without changing the behavior of the FileStream. Or maybe it should be changed to add the 302 and 463 errors handling ?

I also though maybe i put some code before the target.getStream() is called, but that would require to have similar implementation as the getStream, to distinguish between remote vs local execution and would probably include additional method or class to get information about the path, file.

Opinions ?

#118 - 06/09/2023 10:42 AM - Greg Shah

Is this intended that server does not see any information about the error when the error is thrown in the client ?

Generally, no. Also we very much want to avoid throwing Java exceptions instead of the 4GL-compatible condition exceptions or OO 4GL error classes.

Also I am struggling to guess what can be done in this case, without changing the behavior of the FileStream. Or maybe it should be changed to add the 302 and 463 errors handling ?

This is mostly likely the correct approach.

#119 - 06/12/2023 10:25 AM - Maciej Zieniuk

I am not exactly sure how to handle the DeferredLegacyErrorException exception, when the client throws 302 or 463.

Even if i catch it, let's say in XmlExport.write catch block, what do i do with it ?

Do i get the exception details with LegacyErrorException lex = ErrorManager.getDeferredLegacyError() and manually extract the error codes and then re-record them with ErrorManager.recordOrShowError one by one ?

This feels wrong, especially in that place.

To me it feels like the DeferredLegacyErrorException should be caught in the server right after the remote call from the client, which is either in the place when it's called, StreamFactory.openFileStream, when the wa.rsb.openFileStream(filename, write, append); is executed or directly in the RemoteObject\$RemoteAccess.invokeCore.

#120 - 06/12/2023 10:36 AM - Greg Shah

We already catch DeferredLegacyErrorException in BlockManager.processBody() and we call BlockManager.processLegacyError() which tries to do this processing you are talking about. Please look at that code and determine why it is not working in this case. Perhaps we are intercepting this exception too early and need to let it "bubble up" to the BlockManager?

#121 - 06/12/2023 10:56 AM - Constantin Asofiei

DeferredLegacyErrorException is missing the case for when is thrown from the FWD client. Originally this was meant for the FWD server to save it in ErrorManager\$WorkArea.legacyError, and process it later in processLegacyError.

I think we need for ThinClient.getChanges to transfer ErrorManager\$WorkArea.legacyError to the FWD server, and also let DeferredLegacyErrorException propagate to the FWD server.

#122 - 06/12/2023 11:03 AM - Greg Shah

Rather than processing the error manager logging on the client, why not add the needed handling to `DeferredLegacyErrorException` and let it be thrown and caught naturally? It seems to be a more straightforward solution.

#123 - 06/12/2023 11:27 AM - Constantin Asofiei

Greg Shah wrote:

Rather than processing the error manager logging on the client, why not add the needed handling to `DeferredLegacyErrorException` and let it be thrown and caught naturally? It seems to be a more straightforward solution.

The problem here I think was related to legacy error serialization. We can add the error instance directly to `DeferredLegacyErrorException` to not rely on `ErrorManager$WorkArea.legacyError`, and when the client throws this, the server will be able to have all the info.

#124 - 06/12/2023 11:34 AM - Greg Shah

Good.

Maciej: Please move ahead on this basis.

#125 - 06/13/2023 11:16 AM - Maciej Zieniuk

I have implemented the changes, under rev 14621 on 4514a branch. There are still some other errors that the `TestWriteXmlFile` is expecting but they are not there, so investigating that next.

#126 - 06/13/2023 01:04 PM - Constantin Asofiei

Having this:

```
if (target.isDirectory())
{
    throw new ErrorConditionException(463, "\"" + filename + "\" is a directory");
}
else if (filename.endsWith("/"))
{
    throw new ErrorConditionException(302, target.getPath() + " is not a directory");
}
```

in `FileStream` constructor doesn't seem right. Try a output to path/to/folder and you will see that error 463 is not thrown.

Also, the way you changed `DeferredLegacyErrorException` was not what I really meant. I'll get back to you on this.

#127 - 06/14/2023 09:33 AM - Maciej Zieniuk

I have problems understanding the other failures in the TestWriteXmlFile.cls method writeFileInvalid.

This portion of the test, checks that Progress.Lang.Object cannot be serialized into the xml during the WRITE-XML and expects a failure:

```
// P.L.O fields can't be serialized
// write empty but with schema
dsTest:write-xml('file', outXmlFilePath + "test_xml.xml", false, ?, ?, true) no-error.
```

```
assign
    extent(errNums) = ?
    extent(errNums) = 3
    errNums[1]      = 13078
    errNums[2]      = 13066
    errNums[3]      = 13092.
```

```
assign
    extent(errMsg) = ?
    extent(errMsg) = 3
    errMsg[1]      = 'Unsupported data type for xml serialization: progress.lang.object. (13078)'
    errMsg[2]      = 'Unable to write XML Schema for temp-table ttCustomer. (13066)'
    errMsg[3]      = 'Write in-line schema failed for WRITE-XML. (13092)'.
support.test.AssertExt:Errors(errNums, errMsg, true).
```

The FWD does not fail, because check it checks for wrong types, it detects that Progress.Lang.Object is serializable, so it can see no problem with that - which from what i understood from the documentation, it is serializable, just depends on the class fields.

I can make it unserializable for write-xml or in overall, but that is probably going to break other things.

#128 - 06/14/2023 02:21 PM - Greg Shah

I have problems understanding the other failures in the TestWriteXmlFile.cls method writeFileInvalid.

This portion of the test, checks that Progress.Lang.Object cannot be serialized into the xml during the WRITE-XML and expects a failure:

Marian: Any idea why this reports as unserializable when the class itself is marked serializable?

I can make it unserializable for write-xml or in overall, but that is probably going to break other things.

Correct, this doesn't seem right. We need to address the root cause difference.

#129 - 06/14/2023 02:44 PM - Constantin Asofiei

progress.lang.object fields are not serializable. This test:

```
def temp-table ttl field f1 as progress.lang.object.  
create ttl.  
ttl.f1 = new progress.lang.object().  
  
output to a.txt.  
temp-table ttl:write-xml("file", "a.xml").  
output close.
```

writes these errors:

```
Unsupported data type for XML Serialization: progress.lang.object. (13078)  
Write temp-table data failed for WRITE-XML. (13093)
```

We need to fix this at the WRITE-XML FWD implementation.

#130 - 06/15/2023 03:03 AM - Marian Edu

Greg Shah wrote:

Marian: Any idea why this reports as unserializable when the class itself is marked serializable?

'Serializable' for XML has a different meaning than that of OO serializable, the later is used to pass the objects between client/server but this doesn't mean the object can be serialized in XML, nor JSON as a matter of fact. There were discussions at the time wether or not the erialization format should be public or not, most peoples said we do not care how they do it as long as we can pass along objects from client/server and back :)

I can make it unserializable for write-xml or in overall, but that is probably going to break other things.

Correct, this doesn't seem right. We need to address the root cause difference.

As it is now temp-table (and as a consequence dataset) data can't be serialised if there is any P.L.O field in it, it doesn't matter if the objects there are serialisable or not. When there will be an option to specify the actual class for a temp-table field that might offer the opportunity to support XML/JSON serialisation... until then I would say only fix the xml-json serialisation for temp-table.

#131 - 06/16/2023 04:08 AM - Maciej Zieniuk

I have a potential fix for the remaining issues of the TestWriteXmlFile.cls, but are probably wrong:

- BaseDataType.java the OBJECT data-type to be fully qualified as PROGRESS.LANG.OBJECT. I did not find any direct references of the OBJECT and i could not find any other use cases that could confirm that this is correct in different cases (For some reason calling DATA-TYPE directly on the buffer / temp-table field did not worked for me in 4GL)

- com/goldencode/p2j/persist/serial/Util.java the object not to be supported for serialization. This class seems to be only used for XML or JSON serialization. Also when the serialize-hidden is true, the serialization should be skipped for that field (no validation or errors). Not sure whether i added it in the correct place.

```
=== modified file 'src/com/goldencode/p2j/persist/serial/Util.java'
--- old/src/com/goldencode/p2j/persist/serial/Util.java 2023-05-25 08:21:36 +0000
+++ new/src/com/goldencode/p2j/persist/serial/Util.java 2023-06-15 13:25:16 +0000
@@ -129,7 +129,7 @@
     mutableDefMap.put (recid.class,      "?");
     mutableDefMap.put (rowid.class,     "?");
     mutableDefMap.put (handle.class,    "?"); /* widget-handle */
-    mutableDefMap.put (object.class,   "?"); /* ??? */
+//    mutableDefMap.put (object.class,  "?"); /* ??? */
     defaultValuesMap = Collections.unmodifiableMap(mutableDefMap);

     Map<Class<? extends BaseDataType>, String> mutableTypeMap = new IdentityHashMap<>();
@@ -148,7 +148,7 @@
     mutableTypeMap.put (recid.class,    "xsd:long");
     mutableTypeMap.put (rowid.class,    "xsd:base64Binary");
     mutableTypeMap.put (handle.class,   "xsd:long"); /* widget-handle */
-    mutableTypeMap.put (object.class,   "xsd:long"); /* ??? */
+//    mutableTypeMap.put (object.class,  "xsd:long"); /* ??? */
     xsdMap = Collections.unmodifiableMap(mutableTypeMap);

     Map<Class<? extends BaseDataType>, String> mutableSubTypeMap = new IdentityHashMap<>();
@@ -159,7 +159,7 @@
     mutableSubTypeMap.put (recid.class,  "prodata:recid");
     mutableSubTypeMap.put (rowid.class,  "prodata:rowid");
     mutableSubTypeMap.put (handle.class, "prodata:handle"); /* widget-handle */
-    mutableSubTypeMap.put (object.class, "prodata:Progress.Lang.Object"); /* ??? */
+//    mutableSubTypeMap.put (object.class, "prodata:Progress.Lang.Object"); /* ??? */
     xsdProdataMap = Collections.unmodifiableMap(mutableSubTypeMap);

     // longchar not supported
@@ -230,6 +230,10 @@
     {
         for (TempTableSchema.Column column : schema.columns())
         {
+            if (column.isSerializeHidden())
+            {
+                continue;
+            }
             Class<? extends BaseDataType> bdtClass = column.getType();
             if (!xsdMap.containsKey(bdtClass))
             {
@@ -250,6 +254,10 @@
             */
             static boolean checkSupportedType(TempTableSchema.Column column)
             {
+                if (column.isSerializeHidden())
+                {
+                    return true;
+                }
                 Class<? extends BaseDataType> bdtClass = column.getType();
                 return xsdMap.containsKey(bdtClass);
             }
         }
     }
 }
```

```

=== modified file 'src/com/goldencode/p2j/persist/serial/XmlExport.java'
--- old/src/com/goldencode/p2j/persist/serial/XmlExport.java 2023-06-13 15:10:01 +0000
+++ new/src/com/goldencode/p2j/persist/serial/XmlExport.java 2023-06-15 13:33:41 +0000
@@ -210,11 +210,11 @@
     /** Number of DataSet non PARENT-ID-RELATION relations */
     private int prodataRelationCount ;

-// /**
-//  * Flags a {@code WRITE-XMLSCHEMA} method in progress (for both {@code DataSet} and
-//  * {@code TempTable}). When a {@code WRITE-XML} is processed the value is {@code false}.
-//  */
-// private boolean isSchemaOnly = false;
+ /**
+  * Flags a {@code WRITE-XMLSCHEMA} method in progress (for both {@code DataSet} and
+  * {@code TempTable}). When a {@code WRITE-XML} is processed the value is {@code false}.
+  */
+ private boolean isSchemaOnly = false;

    /** The list of indexes, in order of their buffers. */
    private final Map<Integer, Pair<String, TableMapper.LegacyIndexInfo>> indexes = new TreeMap<>();
@@ -685,8 +685,16 @@
        // Unsupported data type for XML Serialization: <type>.
        EntityManager.recordOrShowError(13066, schema.getTable_name());
        // Unable to write XML Schema for temp-table <table-name>.
        EntityManager.recordOrShowError(13097);
        // Write schema failed for WRITE-XMLSCHEMA.
+        if (isSchemaOnly)
+        {
+            EntityManager.recordOrShowError(13097);
+            // Write schema failed for WRITE-XMLSCHEMA.
+        }
+        else
+        {
+            EntityManager.recordOrShowError(13092);
+            // Write in-line schema failed for WRITE-XML.
+        }
        return false;
    }
}
@@ -785,7 +793,7 @@
{
    return false;
}
-// this.isSchemaOnly = true;
+ this.isSchemaOnly = true;

    try
    {
@@ -793,7 +801,7 @@
    }
    catch (ErrorConditionException e2)
    {
-        EntityManager.recordOrShowError(4065, "WRITE-XML", "DATASET widget");
+        EntityManager.recordOrShowError(4065, "WRITE-XMLSCHEMA", "DATASET widget");
        // **The <attribute> attribute on the <widget id> has invalid arguments. (4065)

        return false;
@@ -887,7 +895,7 @@
    return false;
}

-// this.isSchemaOnly = true;
+ this.isSchemaOnly = true;
return write() -> writeTableSchema(
    (BufferImpl) buffer.getDMOProxy(),
    new TempTableSchema(buffer, false),
@@ -2413,8 +2421,16 @@
    // Unsupported data type for XML Serialization: <type>.
    EntityManager.recordOrShowError(13066, schema.getTable_name());
    // Unable to write XML Schema for temp-table <table-name>.
    EntityManager.recordOrShowError(13097);
    // Write schema failed for WRITE-XMLSCHEMA.
+    if (isSchemaOnly)
+    {
+        EntityManager.recordOrShowError(13097);

```



```

+         // Write schema failed for WRITE-XMLSHEMA.
+     }
+     else
+     {
+         ErrorManager.recordOrShowError(13092);
+         // Write in-line schema failed for WRITE-XML.
+     }
+     return false;
+ }
+ String addType = Util.xsdProdataMap.get(columnType);
@@ -2668,10 +2684,10 @@
+ String unsupportedType = Util.checkSupportedTypes(schema);
+ if (unsupportedType != null)
+ {
+     ErrorManager.recordOrShowError(13078, unsupportedType);
+     // Unsupported data type for XML Serialization: <type>.
+     ErrorManager.recordOrShowError(13093);
+     // Write temp-table data failed for WRITE-XML.
-     ErrorManager.recordOrThrowError(13078, unsupportedType, "");
-     // Unsupported data type for XML Serialization: <type>.
+     return -1;
+ }

```

=== modified file 'src/com/goldencode/p2j/util/BaseDataType.java'

```

--- old/src/com/goldencode/p2j/util/BaseDataType.java 2023-05-25 08:21:36 +0000
+++ new/src/com/goldencode/p2j/util/BaseDataType.java 2023-06-15 12:04:15 +0000
@@ -960,7 +960,7 @@

```

```

+     LOGICAL("LOGICAL", logical.class),
+     LONGCHAR("LONGCHAR", longchar.class),
+     MEMPTR("MEMPTR", memptr.class),
-     OBJECT("OBJECT", object.class),
+     OBJECT("PROGRESS.LANG.OBJECT", object.class),
+     RAW("RAW", raw.class),
+     RECID("RECID", recid.class),
+     ROWID("ROWID", rowid.class),

```

=== modified file 'src/com/goldencode/p2j/util/ErrorMessageManager.java'

```

--- old/src/com/goldencode/p2j/util/ErrorMessageManager.java 2023-06-13 15:10:01 +0000
+++ new/src/com/goldencode/p2j/util/ErrorMessageManager.java 2023-06-15 12:10:40 +0000
@@ -3463,6 +3463,7 @@

```

```

+     case 13089: msg = "XML Schema definition for field '%1' in table '%2' does not match Temp-Table field definition"; break;
+     case 13090: msg = "XML Schema definition for field '%1' in table '%2' not found"; break;
+     case 13091: msg = "XML Schema definition for table '%1' contains extra field '%2'"; break;
+     case 13092: msg = "Write in-line schema failed for WRITE-XML"; break;
+     case 13093: msg = "Write temp-table data failed for WRITE-XML"; break;
+     case 13095: msg = "WRITE-XML target-type must be 'FILE' when buffers of a dataset have a different N
AMESPACE-URI"; break;
+     case 13097: msg = "Write schema failed for WRITE-XMLSHEMA"; break;

```

#132 - 07/19/2023 03:43 PM - Maciej Zieniuk

- File tests_results_4514a.log added

- File tests_results_trunk.log added

I lot have changed in fwd trunk and testcases repos since i last worked on this, some of my comments may be inaccurate anymore. Here I will try to summarize the progress and findings for each of the dataset tests failures.

You can find the fwd 4514a code on devsrv01 in /opt/secure/code/p2j_repo/p2j/active/4514a, with revisions 14664 to 14679. The branch is rebased on today's trunk.

The testcases 4514a repo is in /home/mjz/testcases/4514a, with revisions 1454 to 1456, rebased on top of trunk today.

The testcases 4514a branch with the tests/dataset/TestSuiteClass1.cls, TestSuiteClass2.cls and TestSuiteClass3.cls contains 421 tests, from which there are 134 failures on top of fwd trunk branch and 107 failures on top of fwd 4514a branch.

Running those tests on 4GL 11.7 results in 19 failures. If we exclude those from FWD tests, then we have few less failures to look at.

See the results of 11.6 run in already attached 4gl_results_20230426_1.xml to _3.xml (I confirmed most of those failures still occurs in 11.7)

There are also tests that i did not run on 11.7, because they disrupts the entire tests suite:

```
tests.dataset.TestDefault
tests.dataset.TestRelationsActive
tests.dataset.TestRowCreateUpdate
tests.dataset.TestRowDelete
tests.dataset.TestXmlNodeType
```

I never really looked into why those fails in 4GL, so if we exclude those from FWD tests, then we have 4 less failures to look at.

See tests_results_trunk.log and tests_results_4514a.log for failures in trunk and 4514a fwd branches.

Detailed analysis on the failures:

- TestAcceptChanges.cls is now failing because of Cannot turn on TRACKING-CHANGES for a static temp-table unless DEFINED with a BEFORE-TABLE phrase. (12344) - that was not the case just few weeks ago, it must be regression on the trunk branch. Please note, that this tests had 3 fails in 4GL 11.7, but for different reason.
- TestAddRelation.cls see <https://proj.goldencode.com/issues/4514#note-60>
- TestCopyDataset.cls there is something wrong with the assert "Equals". Before this test got merged into one test, it contained 1 failure of 13.
- TestFillMode.cls method withPkCorrectBufferAssign see <https://proj.goldencode.com/issues/4514#note-67> and <https://proj.goldencode.com/issues/4514#note-77>. Apparently this is expected to work again in 4GL 12.0
- TestReadJsonFile.cls method modifyTableNameMaster is a regression in trunk, it was not failing just few weeks ago.
- TestReadXml* failures see <https://proj.goldencode.com/issues/4514#note-103> to <https://proj.goldencode.com/issues/4514#note-108>. I left it out, as it was planned to be fixed later.
- TestSetCallback.cls failures see <https://proj.goldencode.com/issues/4514#note-113> and <https://proj.goldencode.com/issues/4514#note-114>. I left it out, as it was planned to be fixed later.
- parameter/inout/TestInputOutputDynamicDsInDynamicProc.cls method passInvalidDatasetHandle see <https://proj.goldencode.com/issues/4514#note-26>. I think i fixed it, not sure why it fails again.
- parameter/input/estDatasetAppend.cls and TestDatasetAppendOo.cls see <https://proj.goldencode.com/issues/4514#note-30>. I was told it will get fixed.
- parameter/input/TestDatasetBind.cls method referenceOnlyNotValid see <https://proj.goldencode.com/issues/4514#note-37> to <https://proj.goldencode.com/issues/4514#note-39>. But i don't remember it failing with NPE on other methods, possible regression.
- parameter/output/TestOutputDynamicDsInDynamicProc.cls see <https://proj.goldencode.com/issues/4514#note-44>, <https://proj.goldencode.com/issues/4514#note-57>, <https://proj.goldencode.com/issues/4514#note-58>, <https://proj.goldencode.com/issues/4514#note-68>, <https://proj.goldencode.com/issues/4514#note-78>, <https://proj.goldencode.com/issues/4514#note-79>, <https://proj.goldencode.com/issues/4514#note-83>. . Previously it failed by disrupting all other tests. It still does, but at least to this test class only.

- File file-cvt-list.txt added

Some additional context on the testcases 4514a repo, all revisions (1454 to 1456) are safe to put into testcases repo - they mostly contain conversion fixes or clarity for errors when asserting.

For the fwd 4514a branch, here are some notes:

- The 14671 revision contains changes in the TempTableBuilder.java for the prepared state, that i reverted in 14672 - see <https://proj.goldencode.com/issues/4514#note-90> to <https://proj.goldencode.com/issues/4514#note-100>
- The 14674 revision contains changes in the ControlFlowOps.java, which uses org.apache.commons.lang3 as a dependency. I hope that's not a problem.
- The 14678 revision contains changes in the XmlExport.java method writeDataset. The diff shows multiple lines changed, but if you ignore the whitespaces, it's just the try-catch. This revision also contains changes, where i add 4gl error code and message when DeferredLegacyErrorException is thrown (to be available for later error handling).
- The 14679 revision contains changes in the src/com/goldencode/p2j/persist/serial/Util.java file, which removed 4GL object from being serializable. I hope that did not break any other places, as according to the documentation it really isn't. The same revision contains changes in the src/com/goldencode/p2j/util/BaseDataType.java file, which changes the name of the 4GL object to the fully qualified one. Again, i hope it did not break other changes. See <https://proj.goldencode.com/issues/4514#note-127> to <https://proj.goldencode.com/issues/4514#note-130>. I may have fixed it in the wrong place..

Except the last 14679 revision around 4gl object serialization, everything else is safe to put into trunk in my opinion.

Few other notes:

- I have made a suggestion to change the DataSet.java method unableToAssignUnknown to record the 4083 error as warning, as i did one by one in few other places (the same pattern repeats for this error), but did not had time to test everything else. See <https://proj.goldencode.com/issues/4514#note-66> and <https://proj.goldencode.com/issues/4514#note-72>
- I had to link the tests and resources directory in the deploy/client for it to work properly, see <https://proj.goldencode.com/issues/4514#note-85>
- I did not had a chance to fix this conversion issue / typo (not really affecting the code or tests in any way, just converting unwanted code) see <https://proj.goldencode.com/issues/4514#note-54>
- The tests/dataset/TestHandle.cls does not convert correctly, as it results in java compilation error - so i excluded it from my file-cvt-list.txt. To be honest i am not sure what the TestGet is testing, i would change this test to actually change this test to save it to variable and check for the result then. But i guess you could argue if hObjDynamic:handle no-error. is a valid statement in 4GL, why would it not be in FWD ? See <https://proj.goldencode.com/issues/4514#note-48> to <https://proj.goldencode.com/issues/4514#note-51>
- See attached for the file-cvt-list.txt for the dataset tests
- This is the command of how i run the tests from deploy/client: (I used this, instead of Eclipse, as it is does not have UI, so i was not bothered with popups)

```
java -Djava.library.path=/home/mjz/projects/p2j/testcases/p2j/build/native -cp "/home/mjz/projects/p2j/junit/junit-platform-console-standalone-1.9.2.jar:../lib/*:../lib/spi/*" -agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=2081 org.junit.platform.console.ConsoleLauncher --include-engine=fwd-junit -cp "/home/mjz/projects/p2j/testcases/build/lib/testcases.jar" --details=verbose @junit-args
```

where the deploy/client/junit-args contained:

```
--select-class=com.goldencode.testcases.tests.dataset.TestSuiteClass1
--select-class=com.goldencode.testcases.tests.dataset.TestSuiteClass2
--select-class=com.goldencode.testcases.tests.dataset.TestSuiteClass3
```

#134 - 07/20/2023 06:03 AM - Maciej Zieniuk

I have merged the testcases 4514a branch repo revisions 1454 to 1456 into the testcases repo bzr+ssh://xfer.goldencode.com/opt/testcases/

#135 - 09/13/2023 08:18 AM - Greg Shah

- Assignee changed from Maciej Zieniuk to Ovidiu Maxiniuc

Files

| | | | |
|--|-----------|------------|---------------------|
| 4gl_results_20230426_2.xml | 35.4 KB | 04/25/2023 | Maciej Zieniuk |
| 4gl_results_20230426_3.xml | 13.3 KB | 04/25/2023 | Maciej Zieniuk |
| 4gl_results_20230426_1.xml | 42.6 KB | 04/25/2023 | Maciej Zieniuk |
| file-cvt-list.txt | 12.5 KB | 05/02/2023 | Maciej Zieniuk |
| TestOutputDynamicDsInDynamicProc.p | 2.32 KB | 05/08/2023 | Vladimir Tsichevski |
| persitentprocwitherror.p | 117 Bytes | 05/08/2023 | Vladimir Tsichevski |
| testerrorinpersistentproc.p | 193 Bytes | 05/08/2023 | Vladimir Tsichevski |
| TestOutputDynamicDsInDynamicProclInvalidHandle.p | 2.57 KB | 05/09/2023 | Maciej Zieniuk |
| tests_results_trunk.log | 404 KB | 07/19/2023 | Maciej Zieniuk |
| tests_results_4514a.log | 381 KB | 07/19/2023 | Maciej Zieniuk |
| file-cvt-list.txt | 12.4 KB | 07/19/2023 | Maciej Zieniuk |