User Interface - Bug #8353

Fully support stanza INI LOAD/USE/UNLOAD scenarios

02/27/2024 11:02 AM - Roger Borrello

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

Description

Related issues:

Related to User Interface - Feature #4517: optionally back the 4GL features f...	**Closed**

History

#1 - 02/27/2024 11:33 AM - Roger Borrello

- Related to Feature #4517: optionally back the 4GL features for Registry access with the user-specific offline storage features added

#2 - 02/27/2024 12:18 PM - Roger Borrello

Continuing progress on #4517...

When I introduced a testcase, there is other breakage since there are strange aspects to how Progress deals with LOAD/USE/UNLOAD in the areas of:

- The .ini extension doesn't seem to be required, it is added when needed
- The case of the INI value must match between LOAD/USE/UNLOAD.

The testing setup is taking longer than the code update, but I believe my updates in revision 15010 of this branch now handle what I've been able to test much better.

The rules I've found via checks on Progress are:

- The fname and envname are 2 distinct entities in LOAD/USE/UNLOAD
- In LOAD, envname is the fname stripped and fname is forced to .ini extension, if none given.
  - fname is the primary key. Lookups are done matching the envname to the filename and the first match is returned.
- In USE/UNLOAD, if an extension is passed it must match the fname, or fail.
  - If an extension is not passed, first envname is checked for. If not found, then .ini appended and checked against the fname.
- All checks are case-sensitive

I am trying to form tests that work in the new methodology, but so far I haven't had luck. Mostly because of my lack of 4GL skills. The testcase should:

1. use various combinations of names passed to LOAD and attempt USE/UNLOAD with matching/non-matching combinations.
2. utilize multiple LOAD statements in various orders, and verify using GET-KEY-VALUE to validate values.
3. determine if a new file is created properly by writing/reading
4. read groups of sections and keys by passing null where appropriate
5. be performed in GUI and ChUI

There is a bug in the reading of a file when a section is the last line:

```
[section1]
key1 = value3txt1
key2 = value3txt2
[section2]
```

I have to add a newline at the end of the file for **section2** to be found.

Here is my currently manual testcase:

```
def var path as char init "testcases_v2/abl/stanza_ini".
def var inidir as char.
inidir = "z:/" + path.
if opsys = "unix" then
   inidir= "/home/rfb/" + path.
if opsys <> "win32" then
   message opsys "is an unsupported OS".

DEFINE VARIABLE keyval AS CHARACTER NO-UNDO FORMAT "x(128)".

/* Normal checks */
load "env.1.inifile" dir inidir base-key "INI".
use "env.1.inifile".
GET-KEY-VALUE SECTION "section1" KEY "key1" VALUE keyval.
message "keyval="keyval.
use "".
unload "env.1.inifile".

load "env1.ini" dir inidir base-key "INI".
use "env1".
GET-KEY-VALUE SECTION "section1" KEY "key1" VALUE keyval.
message "keyval="keyval.
use "".
unload "env1".

load "env1" dir inidir base-key "INI".
use "env1.ini".
GET-KEY-VALUE SECTION "section1" KEY "key1" VALUE keyval.
message "keyval="keyval.
use "".
unload "env1".

load "env1.ini" dir inidir base-key "INI".
use "env1.ini".
GET-KEY-VALUE SECTION "section1" KEY "key1" VALUE keyval.
message "keyval="keyval.
use "".
unload "env1.ini".

load "env3.txt" dir inidir base-key "INI".
load "env3" dir inidir base-key "INI".
use "env3".
GET-KEY-VALUE SECTION "section1" KEY "key1" VALUE keyval.
message "keyval="keyval.

GET-KEY-VALUE SECTION "" KEY "key1" VALUE keyval.
message "keyval="keyval.

GET-KEY-VALUE SECTION "section1" KEY "" VALUE keyval.
message "keyval="keyval.

GET-KEY-VALUE SECTION "section" KEY DEFAULT VALUE keyval.
message "keyval="keyval.
unload "env3".

use "".
GET-KEY-VALUE SECTION "section1" KEY "key1" VALUE keyval.
message "keyval="keyval.
```

#3 - 03/08/2024 06:57 PM - Roger Borrello

I was able to learn enough to be dangerous for creating an ABL Unit testcase for stanza_ini. These 2 files are in tests/stanza_ini.

Suite.cls:

```
block-level on error undo, throw.

@TestSuite(classes="tests.stanza_ini.Test").
class tests.stanza_ini.Suite:
end class.

/* EOF */
```

Test.cls:

```
using OpenEdge.Core.Assert.

block-level on error undo, throw.

/* Core tests for stanza INI file usage, refs #8353 */

class tests.stanza_ini.Test:

   @Before.
   method public void UT_SceneSetup():
      session:suppress-warnings = false.
   end method.

   @After.
   method public void UT_SceneTeardown():
   end method.

   @Setup.
   method public void UT_TestSetup():
      os-delete ./env.ini.
   end method.

   @TearDown.
   method public void UT_TestTeardown():
      os-delete ./env.ini.
   end method.

   @Test.
   method public void TestSimpleNoIni():
      /* Case 1 load should fail. No oddities */
      LOAD "env.ini" DIR "./".
      support.test.AssertExt:Error(4449, 'The LOAD of env.ini in path ./ failed. (4449)').
   end method.

end class. /* Test */

/* EOF */
```

But there is a failure in conversion:

```
     [java] ------------------------------------------------------------------------------
     [java] Code Conversion Annotations
     [java] ------------------------------------------------------------------------------
     [java] 
     [java] Optional rule set [customer_specific_annotations_prep] not found.
     [java] ./tests/stanza_ini/Suite.cls
     [java] ./tests/stanza_ini/Test.cls
     [java] Null annotation (full-java-class) for support.test.AssertExt [CLASS_NAME] @32:7 (231928234154)
     [java] Null annotation (simple-java-class) for support.test.AssertExt [CLASS_NAME] @32:7 (231928234154)
     [java] Null annotation (containing-package) for support.test.AssertExt [CLASS_NAME] @32:7 (231928234154)
     [java] Null annotation (found-in-full-java-class) for Error [OO_METH_VOID] @32:30 (231928234155)
     [java] EXPRESSION EXECUTION ERROR:
     [java] ---------------------------
     [java] persist()
     [java] ^  { null value for annotation 'full-java-class':support.test.AssertExt [CLASS_NAME]:231928234154 
@32:7
     [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 {231928233985}
     [java] Copy AST  :  [ block ] BLOCK/ @0:0 {231928233985}
     [java] Condition :  persist()
     [java] Loop      :  false
     [java] --- END RULE REPORT ---
```

I'm not too familiar with OO conversion. Is there any pointers to what this might be?

#4 - 04/03/2024 04:52 PM - Roger Borrello

The failure was because I didn't include support/test/AssertExt.cls in the conversion file. Now I get:

```
    [javac] /home/rfb/projects/testcases_v2_dev2/src/com/goldencode/testcases/tests/stanza_ini/Suite.java:14: 
error: incompatible types: Suite cannot be converted to Annotation
    [javac] @Suite
    [javac]  ^
    [javac] /home/rfb/projects/testcases_v2_dev2/src/com/goldencode/testcases/tests/stanza_ini/Test.java:80: e
rror: incompatible types: Test cannot be converted to Annotation
    [javac]    @Test(order = 1)
    [javac]     ^
    [javac] 2 errors
```

The Suite.java was created, so I'm not sure what I did wrong. Vladimir, if there's any help getting this "Hello World" testcase going, I'd appreciate it.

```
...
/**
 * Business logic (converted to Java from the 4GL source code
 * in tests/stanza_ini/Suite.cls).
 */
@Suite
@SelectClasses(value = 
{
   "tests.stanza_ini.Test" 
})
```

```
public class Suite
extends BaseObject
{
...
```

#5 - 04/19/2024 07:51 PM - Roger Borrello

- Status changed from New to WIP

- Assignee set to Roger Borrello

By changing my test to a procedure, I was able to get it running.

I am finding that LOAD gIniName DIR gDir. results in NPE when we get to the client side EnvironmentDaemon.load, as baseKey is null:

```
   public boolean load(String env,
                       String directory,
                       boolean flagNew,
                       String baseKey)
   {
      EnvironmentReader envAcc;
      Map<String, EnvironmentReader> envs;

      boolean isIniType = EnvironmentReader.TYPE_INI.equalsIgnoreCase(baseKey);
      if (!isIniType && !PlatformHelper.isUnderWindowsFamily())
      {
         envs = envFallbackMap;
         env = env.toLowerCase();
         envAcc = new FallbackEnvironmentReader(env, clientStorage, baseKey.toLowerCase());   // <-- NPE here
      }
```

I'm thinking we should invoke from the server side EnvironmentOps.load with more protection:

```
      if (baseKey != null)
      {
         baseKeyAsString = baseKey.toStringMessage();
      }
      else  // Add an else case
      {
         baseKeyAsString = new String("");
      }

      if (!Environments.load(envAsString, directoryAsString, flagNewAsBoolean, baseKeyAsString))
      {
         onLoadError(env, directory);
      }
```

#6 - 04/20/2024 12:11 AM - Roger Borrello

Actually, the fact that baseKey is null should be OK on the client side. It just needs to be a little more protected, since FallbackEnvironmentReader can accept a null:

```
      boolean isIniType = EnvironmentReader.TYPE_INI.equalsIgnoreCase(baseKey);
      if (!isIniType && !PlatformHelper.isUnderWindowsFamily())
      {
         envs = envFallbackMap;
         env = env.toLowerCase();
         baseKey = (baseKey != null) ? baseKey.toLowerCase() : baseKey;
         envAcc = new FallbackEnvironmentReader(env, clientStorage, baseKey);
      }
```

There is a problem with this case of trying to load a file that doesn't exist, such as my first testcase, LOAD "env.ini" DIR "./". because we get to the fallback environment. In Progress, we get an error, because somehow it knows that env.ini is a file, even though it doesn't exist, and we don't specify basekey "INI".

The LOAD documentation for *environment* has:

> A CHARACTER expression that evaluates to one of the following:
>
> - The name of a registry key to create
> - The name of an initialization file to create
> - The name of an existing registry key
> - The name of an existing initialization file

So I am not sure how to get boolean isIniType to be true, if we don't have a file, nor pass the basekey. Perhaps assuming it is INI if there is .ini at the end of the environment?

#7 - 04/20/2024 11:01 AM - Greg Shah

> Actually, the fact that baseKey is null should be OK on the client side. It just needs to be a little more protected, since

> FallbackEnvironmentReader can accept a null:

Correct, the null has meaning.

> Perhaps assuming it is INI if there is .ini at the end of the environment?

It seems reasonable approach if basekey is null or possibly if it is the empty string (please check this in the 4GL).