

Base Language - Feature #4855

add a 4GL extension **SESSION:SESSION-ID** attribute which returns a **UUID** which uniquely identifies a **FWD session**

08/14/2020 11:53 AM - Greg Shah

Status: Closed	Start date:
Priority: Normal	Due date:
Assignee: Galya B	% Done: 100%
Category:	Estimated time: 0.00 hour
Target version:	version:
billable: No	
vendor_id: GCD	
Description	
Related issues:	
Related to User Interface - Feature #4853: implement support for reporting a ... Test	

History

#1 - 08/14/2020 11:54 AM - Greg Shah

The purpose of this is to allow converted 4GL code to uniquely identify a given session. A customer has requested this to allow them to implement some internal licensing logic.

#2 - 08/14/2020 11:55 AM - Greg Shah

- Related to Feature #4853: implement support for reporting a device-id which is meant to identify a unique client system/browser accessing the FWD server added

#3 - 06/27/2023 07:40 AM - Galya B

How will customers know when older sessions have been ended when implementing licensing rules?

#4 - 06/27/2023 08:16 AM - Greg Shah

I expect that the customer will likely implement their own "licenses in use" table (see #4522) in which one field will be the session id. They can implement [Session Init Term Hooks](#) in Java to remove the record when the session exits.

For the purposes of this task, we just need to have that unique ID for the lifetime of the session.

#5 - 06/27/2023 08:27 AM - Galya B

The hook seem to be in Java, while the customers want to implement it in 4GL. I think they will come back to us to ask for it when this task is completed.

#6 - 06/27/2023 08:51 AM - Greg Shah

Maybe, but it isn't hard to implement the cleanup of a record using Java code. The initial insertion can be done in the 4GL initialization code for the application.

#7 - 06/27/2023 10:08 AM - Galya B

Can we have multiple "connection" sessions (DirectSession / VirtualSession) for one client (not necessarily simultaneous)?

#8 - 06/27/2023 10:10 AM - Galya B

Actually obviously that's how appserver works. I'm just trying to figure out if there is any case where the id should be stable in different BaseSession objects.

#9 - 06/27/2023 10:12 AM - Greg Shah

You can have a virtual session while in an active direct session. You cannot have a virtual session on its own, the idea is that it is a special server-to-server multiplexed session.

You don't have to map the session id to the FWD direct session or FWD virtual session per se. We don't need to distinguish between these FWD session types. From a 4GL code perspective, there is only ever one "session" and that is what we are representing here.

#10 - 06/27/2023 10:17 AM - Greg Shah

I'm just trying to figure out if there is any case where the id should be stable in different BaseSession objects.

I don't want to map this to a FWD session. This is about the 4GL session, which follows [the Highlander principle](#).

#11 - 06/27/2023 10:18 AM - Galya B

OK, leaving BaseSession aside. A session from 4GL perspective should be:

- the unique conversation (with standalone gui/chui/batch processes and web gui/chui/embedded);
- the unique connection to appserver (running external procedures)?
- the unique connection to rest / soap?

#12 - 06/27/2023 10:23 AM - Greg Shah

the unique conversation (with standalone gui/chui/batch processes);

Yes

the unique connection to appserver (running external procedures)?

Yes, though the lifetime is dependent upon the connection type (stateful, state free, state reset...).

the unique connection to rest / soap?

Yes, though the lifetime is dependent upon the agent.

I think we don't need to consider this too much. Calculate the value **only when it is first read** and store it in the SessionUtils.WorkArea. The lifetime of that context will already be properly managed and you don't need to deal with it. If the value is never read, then it is never calculated and is just set to null by default.

#13 - 06/27/2023 10:26 AM - Galya B

- Assignee set to Galya B
- Status changed from New to WIP

Branch 4855a created.

#14 - 06/28/2023 09:06 AM - Galya B

What is client-principal:session-id <https://docs.progress.com/bundle/abl-reference/page/SESSION-ID-attribute.html#SESSION-ID-attribute> . We have that implemented in com.goldencode.p2j.util.ClientPrincipal as SESSION_ID(AttrType.CHAR, 0xf, "SESSION-ID"),.

#15 - 06/28/2023 09:33 AM - Constantin Asofiei

Galya B wrote:

What is client-principal:session-id <https://docs.progress.com/bundle/abl-reference/page/SESSION-ID-attribute.html#SESSION-ID-attribute> . We have that implemented in com.goldencode.p2j.util.ClientPrincipal as SESSION_ID(AttrType.CHAR, 0xf, "SESSION-ID"),.

The attribute is r/w, so it can be set by the application to anything (thus uniqueness is not guaranteed). And it belongs to CLIENT-PRINCIPAL.

#16 - 06/28/2023 09:33 AM - Greg Shah

The client-principal is a kind of 4GL based security-related token that identifies an authenticated identity. It is not guaranteed to be used at all (you have to write your 4GL code to use it) and this usage requires other code to handle the authentication. It is also not guaranteed to be stable for the entire lifetime of the 4GL session. It can be swapped out or logged out and replaced.

Our session-id will have no relation to the client-principal.

#17 - 06/28/2023 09:34 AM - Galya B

Ok, thanks for the explanations, it's just that the new keyword will be quite similar to the old one.

#18 - 06/28/2023 09:35 AM - Greg Shah

I expect we would use the same keyword. There is nothing wrong with that.

#19 - 06/28/2023 09:36 AM - Constantin Asofiei

If we reuse this keyword, SESSION-ID will need to be extracted to its own interface, specific handle.unwrap added for this attribute, and some conversion rules changes, also.

#20 - 06/28/2023 09:37 AM - Galya B

I was going with a similar, but not exactly the same keyword, so I'll stick with it.

#21 - 06/28/2023 09:42 AM - Greg Shah

I was going with a similar, but not exactly the same keyword, so I'll stick with it.

I don't want to add a keyword where there other will do.

#22 - 06/28/2023 09:44 AM - Galya B

I want to know what I'm doing, but I don't.

#23 - 06/28/2023 10:57 AM - Greg Shah

In this case, since we already have a keyword and we even have an attribute with that name, there are no changes needed in progress.g. The other changes are still needed (e.g. methods_attributes.rules etc...).

#24 - 06/29/2023 05:36 AM - Galya B

- Status changed from WIP to Review

- % Done changed from 0 to 100

4855a r14634 for review

#25 - 07/07/2023 02:52 PM - Greg Shah

Code Review Task Branch 4855a Revision 14634

No objections.

#26 - 07/07/2023 02:52 PM - Greg Shah

We are trying to avoid conversion changes in trunk for a short time, so we may not merge this until later next week.

#27 - 07/17/2023 03:43 PM - Greg Shah

You can rebase and merge 4855a to trunk.

#28 - 07/18/2023 12:57 AM - Galya B

- Status changed from Review to Test

Task branch 4855a was merged to trunk as rev 14661 and archived. Email sent.

#29 - 07/18/2023 07:48 AM - Greg Shah

- Status changed from Test to Closed

#30 - 07/19/2023 01:26 AM - Constantin Asofiei

The changes in trunk rev 14661 are not OK. As SESSION-ID exists now at both CLIENT-PRINCIPAL and SESSION, we need this attr in a dedicated interface, with its own handle.unwrap method (as mentioned at [#4855-19](#)).

#31 - 07/19/2023 02:58 AM - Galya B

Do I need to invent something or you can point me to an example or programming principle?

#32 - 07/19/2023 03:03 AM - Galya B

SESSION:SESSION-ID will become writable.

#33 - 07/19/2023 03:05 AM - Galya B

Can I now have a different keyword like I wanted, or we'll make it writable although it makes no sense?

#34 - 07/19/2023 03:23 AM - Constantin Asofiei

This test needs to work properly to validate the fix:

```
def var ch as char.  
def var h as handle.  
create client-principal h.  
  
h:session-id = "abc".  
ch = h:session-id.  
message ch.  
  
session:session-id = "123" no-error. // this will raise an ERROR condition  
message error-status:error error-status:get-message(1).  
ch = session:session-id.  
message ch.
```

The steps are these:

- extract the SESSION-ID getter/setter into their own interface (both setters with character and String parameters)
- add this interface to CommonSession and ClientPrincipal
- add a handle.unwrapSessionId API for this interface
- implement the setter in SessionUtils, to just call readOnlyError
- change methods_attributes.rules to use this 'unwrap' API.

#35 - 07/19/2023 05:26 AM - Galya B

4855b r14663-14664 I can't make it work:

```
Sessionid.java:36: error: cannot find symbol  
[javac]         silent(() -> new character("123").setSessionId());  
[javac]         ^  
[javac]     symbol:   method setSessionId()  
[javac]     location: class character  
[javac] /home/gbb/Code/testcases-working/src/com/goldencode/testcases/galya/Sessionid.java:42: error: no s  
uitable method found for assign(no arguments)
```

```
[javac]         ch.assign();
[javac]         ^
[javac]         method BaseDataType.assign(BaseDataType) is not applicable
[javac]         (actual and formal argument lists differ in length)
[javac]         method BaseDataType.assign(jobject<?>) is not applicable
```

I think all is correct, but I wasn't able to find the right unwrap in methods_attributes.rules. I tried different variants, all seem the same to me.

#36 - 07/19/2023 06:53 AM - Constantin Asofiei

You need this code adjusted for SESSION-ID, in methods-attributes.rules, the section for SESSION handle:

```
<rule>ftype == prog.kw_sessn_id
  <action>hwrap = "SessionId"</action>
  <action>methodText = "SessionUtils.getSessionId"</action>
  <rule>isAssign
    <action>methodText = "SessionUtils.setSessionId"</action>
  </rule>
</rule>
```

#37 - 07/19/2023 07:22 AM - Galya B

4855b r14665 ready for review.

This is the output of the test proc:

```
abc
yes **SESSION-ID is not a settable attribute for PSEUDO-WIDGET. (4052)
c76891eb-50c0-4b60-8c6b-1c4e70baef3b
```

#38 - 07/19/2023 08:02 AM - Constantin Asofiei

Review for 4855b/14665 - I'm OK with the changes.

#39 - 07/19/2023 08:29 AM - Galya B

Do I merge?

#40 - 07/19/2023 10:22 AM - Greg Shah

You can merge to trunk.

#41 - 07/19/2023 11:05 AM - Galya B

Task branch 4855b was merged to trunk as rev 14665 and archived.