Base Language - Bug #4824

Conversion error for property with indeterminate extent

07/29/2020 05:51 AM - Marian Edu

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Constantin Asofiei	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD		

Description

History

#1 - 07/30/2020 08:48 AM - Greg Shah

- Start date deleted (07/29/2020)

- Description updated

- Assignee deleted (Greg Shah)

From Marian:

> If we have a property defined as an indeterminate extent the generated code doesn't (fully) work. It will generate a 'getter' that have one input parameter (int64) and two methods lengthOf and resize for the property. Those seems to solve the use case when items from the array are being accessed as well as finding out the length or resize the array. There is nothing that returns the whole array:
>
> ```
> public object<? extends com.goldencode.p2j.oo.lang._BaseObject_> getValue_1(final int64 _idx);
> ```
>
> Shouldn't something like this be added as well?
>
> ```
> public object<? extends com.goldencode.p2j.oo.lang._BaseObject_>[] getValue_1();
> ```
>
> This is how the call is being made when accessing that property, not just one element of the array:
>
> ```
> oaObject[0] = (object<? extends _BaseObject_>[]) assignMulti(oaObject[0], inst.ref().getValue_1());
> ```
>
> Well, I'm cheating here since the generated code uses the 4GL class name (Progress.Lang.Object) instead of the FWD equivalent (*BaseObject*) but I think we already reported that one.

#2 - 07/30/2020 09:32 AM - Greg Shah

- Assignee set to Constantin Asofiei

#3 - 08/03/2020 05:27 PM - Constantin Asofiei

- Status changed from New to WIP

Marian is right, we need 'bulk' getter and setter for extent properties. I have most of the changes, there's a tricky part where the setter/getter is emitted with the 4GL's GET/SET clauses, and I need to refactor that so the additional one can be emitted.

#4 - 08/04/2020 12:08 PM - Constantin Asofiei

- % Done changed from 0 to 100

Fixed in 3821c rev 11436. Please review.

#5 - 08/04/2020 01:54 PM - Constantin Asofiei

3821c/11437 fixes the case of abstract extent properties (forgot to check them).

#6 - 08/04/2020 02:11 PM - Greg Shah

- Status changed from WIP to Test

I'm good with the changes.

#7 - 02/05/2021 07:28 AM - Marian Edu

This seems to work on 4384h, at least for oo/Discovery/OpenEdge/Core/IObjectArrayHolder.cls.

#8 - 02/05/2021 08:23 AM - Greg Shah

- Status changed from Test to Closed

#9 - 03/18/2021 09:25 AM - Marian Edu

Constantin, I have a conversion error when using one of the .NET classes ClientSocketConnectionParameters. There are two extent properties in that class and although I do see two getters (one for whole array, one for specific array item), two setters (again, whole array or specific item) and two extra methods one of type LENGTH and one RESIZE when we ran the conversion code that is checking the size of the property array is not using the corresponding LENGTH method but the getter returning the whole array :(

```
extent(obj:SslCiphers) -> obj.ref().getSslCiphers()
```

However, running the same conversion on a test we have for this oo/quirks/property_extent/test_property_extent_type.p is different (correct).

```
extent(obj:ExtentProperty) -> obj.ref().lengthOfExtentProperty()
```

One issue there with the converted code is that if we try to reset the property by setting the extent to null (?) we get this:

```
obj.ref().resizeExtentProperty(new unknown())
```

which obviously does not compile as the method expects an int64.

#10 - 03/18/2021 09:39 AM - Constantin Asofiei

Marian, please post the 4GL code snippet which doesn't convert properly. Thanks!

#11 - 03/18/2021 09:42 AM - Marian Edu

Constantin Asofiei wrote:

> Marian, please post the 4GL code snippet which doesn't convert properly. Thanks!

The code is in oo/openedge/net/server_connection/client_socket_connection_parameters/test_property_ssl_ciphers.p but basically it's this that doesn't convert right:

```
define variable oCSCP         as ClientSocketConnectionParameters no-undo.

define variable numExt        as integer                          no-undo.

oCSCP = new ClientSocketConnectionParameters() no-error.
numExt = extent(oCSCP:SslCiphers).
```

#12 - 03/18/2021 11:20 AM - Constantin Asofiei

- File builtin_prop_extent.patch added

Marian, please use the attached patch. I still need to reconvert some apps before deciding that's OK for 3821c.

#13 - 03/19/2021 03:31 AM - Marian Edu

Constantin Asofiei wrote:

> Marian, please use the attached patch. I still need to reconvert some apps before deciding that's OK for 3821c.

Constantin, did applied the patch and did a gradle ant-all in goldencode followed by ant deploy.all in testcases project. When running conversion it looks like it's still using the getter instead of length when calling extent on that property, some step I've forgot about maybe?

And I don't know what is causing the property indeterminate array to be removed from the dynamic list, right now whenever we try to set it's value it throws error because the extent is not seen as being uninitialized (isDynamicArray returns false).

Now, the TypeFactory.characterExtent register the array as dynamic but pending, at some point I can however see the arrays are moved into the dynamic stack but there is a place where those are being 'unregistered' and haven't yet found it... something extra that need to be done for indeterminate array properties after your changes introducing 'pending' registration?

#14 - 03/19/2021 03:47 AM - Constantin Asofiei

Marian, the patch fixes the example you posted, with sslCiphers, and I've checked other cases and it works. Maybe the patch was not applied fully?

Please let me know an example for the runtime issue with dynamic extent.

#15 - 03/19/2021 05:16 AM - Marian Edu

Constantin Asofiei wrote:

> Marian, the patch fixes the example you posted, with sslCiphers, and I've checked other cases and it works. Maybe the patch was not applied fully?
>
> Please let me know an example for the runtime issue with dynamic extent.

Of course the patch was not applied, the project name do not match so just hitting next without looking what actually happened didn't help... will give that a try in a bit.

#16 - 03/22/2021 08:16 AM - Marian Edu

Constantin Asofiei wrote:

> Marian, the patch fixes the example you posted, with sslCiphers, and I've checked other cases and it works. Maybe the patch was not applied fully?
>
> Please let me know an example for the runtime issue with dynamic extent.

Constantin, with the patch applied the conversion does use the 'lengthOf' method instead of the getter as it was before. Thanks for that.

#17 - 03/22/2021 08:24 AM - Marian Edu

Constantin Asofiei wrote:

> Marian, the patch fixes the example you posted, with sslCiphers, and I've checked other cases and it works. Maybe the patch was not applied fully?
>
> Please let me know an example for the runtime issue with dynamic extent.

It was failing for all tests where the class being tested had an indeterminate extent property, found out for handle the array was not registered as dynamic when initialized (TypeFactory.handleExtent), then for objects the registration was made without the extra flag you've added for pending (default to true) so the array property was left as not being dynamic at the end. Changing the call to registerDynamicArray to pass the second parameter set as false solved the issue for object data type. Last thing I've changed the call to registerDynamicArray in initExtent to do the same thing

and now I get the expected behavior for character data type extents as well.

A test is in oo/openedge/net/server_connection/client_socket_connection_parameters/test_property_ssl_ciphers.p.

#18 - 03/22/2021 02:27 PM - Constantin Asofiei

Marian Edu wrote:

> A test is in oo/openedge/net/server_connection/client_socket_connection_parameters/test_property_ssl_ciphers.p.

Thanks for this. The issue is with persistent programs (which legacy OO instances are) - the dynamic extent (and other) info must be kept for each persistent program, and 'pushed' on the stack when something in that persistent program is called. I'll work on fixing it.

#19 - 03/22/2021 06:11 PM - Constantin Asofiei

- File dyn_array_in_props.txt added

Marian, see the attached patch for the #4824-17 issue. Now the test passes.

I've also fixed some enum issues.

#20 - 03/22/2021 06:22 PM - Constantin Asofiei

BTW, please revert your changes made related to #4824-17.

#21 - 03/23/2021 02:25 AM - Marian Edu

Constantin Asofiei wrote:

> Marian, see the attached patch for the #4824-17 issue. Now the test passes.

I can't apply the patch as-is, even if I remove the changes for TextOps and ClientSocketConnectionParameters it still can't find the patch segment for ObjectOps so I will try to do it manually.

> I've also fixed some enum issues.

Yeah, we did the same for substitute on our branch so guess this is a conflict that will have to be resolved at some point :)

#22 - 03/23/2021 02:27 AM - Marian Edu

Constantin Asofiei wrote:

> BTW, please revert your changes made related to #4824-17.

You probably mean removing the second parameter to registerDynamicArray, pretty sure this has to be called for when an extent handle/object are created with size 0.

#23 - 03/23/2021 02:34 AM - Marian Edu

Constantin Asofiei wrote:

> Marian, see the attached patch for the #4824-17 issue. Now the test passes.

OK, applied the changes on ArrayAssigner and ObjectOps (no idea why Eclipse didn't find the patch segment as it was right at that line). The test now passes without using the second parameter for registerDynamicArray on initExtent.

#24 - 03/23/2021 06:17 AM - Constantin Asofiei

The patches from #4824-19 and #4824-12 are in 3821c rev 12170.

#25 - 10/28/2021 07:18 AM - Marian Edu

Constantin, I see this is closed already but one issue we still see with conversion is when unknown (?) is used to set the size of an extent property (resize).

You can see that in oo/quirks/property_extent/test_property_extent_type.p.

```
extent(extentProperty) = ?.
```

translates to

```
obj.ref().resizeExtentProperty(new unknown()).
```

That comes from literals.rules but the 'classname' note there is null when property's resize method is being called while for other regular OO methods the class name is correctly set to the method's parameter data type.

#26 - 10/28/2021 07:51 AM - Marian Edu

Argh, was beating around the wrong bush... the fix come by adding the 'int64' annotation when the method is rewritten in oo_references.rules (#781):

```
<action>childref.putAnnotation("classname", "int64")</action>
```

Files

builtin_prop_extent.patch	6.71 KB	03/18/2021	Constantin Asofiei
dyn_array_in_props.txt	8.11 KB	03/22/2021	Constantin Asofiei