Conversion Tools - Feature #6302

Improve conversion of empty OCX arguments

04/28/2022 08:23 AM - Hynek Cihlar

Status:	WIP	**Start date:**	
Priority:	Low	**Due date:**	
Assignee:	Vladimir Tsichevski	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**vendor_id:**	GCD

Description

Related issues:

Related to User Interface - Bug #5622: TREEVIEW widget issues	**WIP**

History

#1 - 04/28/2022 08:32 AM - Hynek Cihlar

Empty arguments of converted OCX method invocations are converted into "unknown" instances.

For example the following 4GL assignment

```
hNode = chCfTreeView:TreeView:Nodes:Add(,,cParentNodeID,"Root node").
```

is converted to:

```
hNode.assign(hTree.unwrapTreeView().getNodes().unwrapTreeNodeCollection().add(new integer(), new integer(), cP
arentNodeId, new character("Root node")));
```

It is reasonable to believe that this is incorrect, instead of unknown values the arguments should be "zero" values (or even something else).

Create a set of test cases to find out how OE converts empty arguments and modify the conversion rules accordingly.

#2 - 04/28/2022 08:43 AM - Hynek Cihlar

The function parameter specification in COM type library may contain a default value. Perhaps this is the value used by OE when an empty argument is given. How does OE behave when no defaults are given in the type library?

#3 - 04/28/2022 09:05 AM - Vladimir Tsichevski

- Related to Bug #5622: TREEVIEW widget issues added

#4 - 04/28/2022 09:16 AM - Tijs Wickardt

FYI: see testcases/oxc_to_fwd/mscomctl-treeview-ocx.w :
hNode = chCfTreeView:TreeView:Nodes:Add(,,cParentNodeID,"Root node").

The following is also valid ABL syntax (identical):
hNode = chCfTreeView:TreeView:Nodes:Add(?,?,cParentNodeID,"Root node").

The OpenEdge documentation states explicitly that it's up to the OCX if a parameter is 0 or 1 based, not simply 1-based like the 4GL. And that a 4GL developer must take care of that himself.
The 4GL unknown value (?) translates to null.
It's up to the OCX (and the 4gl runtime) to do something with that.

The following magic number '4' creates a child node:
chCfTreeView:TreeView:Nodes:Add(hNode,4,cChildNodeID,"Child node").

Also see the readme in xfer testcases (next to the source above).

#5 - 04/28/2022 09:20 AM - Tijs Wickardt

Hynek Cihlar wrote:

> hNode.assign(hTree.unwrapTreeView().getNodes().unwrapTreeNodeCollection().add(new integer(), new integer(), cParentNodeId, new character("Root node")));
>
> It is reasonable to believe that this is incorrect, instead of unknown values the arguments should be "zero" values (or even something else).

On first glance, I think the conversion is correct. Zero should not be used I think, that is a value. A sentinel value (null, ellipsis, ...) could make sense.

#6 - 04/28/2022 01:15 PM - Vladimir Tsichevski

This issue must be fixed along with: #5622-12. Otherwise no compilable code is created by conversion for this call.

#7 - 04/28/2022 02:50 PM - Vladimir Tsichevski

- Assignee set to Vladimir Tsichevski

- Status changed from New to WIP

Seems, I was not right here: undefined values **can** be used by OE to pass the "default" value to OCX. Moreover, undefined values are expected by OCX in some situations, for example, it **must** be passed as the parent node reference when creating the first node in the tree.

But the critical issue referenced in #5622-12 is still to be fixed.

#8 - 04/28/2022 03:16 PM - Tijs Wickardt

Vladimir Tsichevski wrote:

> Seems, I was not right here: undefined values **can** be used by OE to pass the "default" value to OCX. Moreover, undefined values are expected by OCX in some situations, for example, it **must** be passed as the parent node reference when creating the first node in the tree.
>
> But the critical issue referenced in #5622-12 is still to be fixed.

Agree.
Nitpick/terminology: The ABL protects against undefined variables. A variable is never uninitialized. If initialized with the unknown value, that's semantically the same as initialized with null/nill/none in other languages. The unknown value is not undefined.
Sorry if that sounded pedantic. Just to take away possible confusion. Please inform me if FWD terminology differs.

#9 - 04/28/2022 03:19 PM - Vladimir Tsichevski

Tijs Wickardt wrote:

> Vladimir Tsichevski wrote:
>
> > Seems, I was not right here: undefined values **can** be used by OE to pass the "default" value to OCX. Moreover, undefined values are expected by OCX in some situations, for example, it **must** be passed as the parent node reference when creating the first node in the tree.
> >
> > But the critical issue referenced in #5622-12 is still to be fixed.
>
> Agree.
> Nitpick/terminology: The ABL protects against undefined variables. A variable is never uninitialized. If initialized with the unknown value, that's the same as initialized with null/nill/none in other languages. The unknown value is not undefined.
> Sorry if that sounded pedantic. Just to take away possible confusion.

Completely agreed. I must use the unknown term everywhere, sorry for the confusion.

#10 - 04/28/2022 06:10 PM - Vladimir Tsichevski

Tijs Wickardt wrote:

> The OpenEdge documentation states explicitly that OCX's are 0-based, not 1-based like the 4GL.

The first argument of the original treeview OCX add call is 1-based node index. This means the first tree node you create has the index 1. And this has no any connection with 4gl.

#11 - 04/28/2022 06:18 PM - Vladimir Tsichevski

Tijs Wickardt wrote:

> FYI: see testcases/oxc_to_fwd/mscomctl-treeview-ocx.w :
> hNode = chCfTreeView:TreeView:Nodes:Add(,,cParentNodeID,"Root node").
>
> The following is also valid ABL syntax (identical):
> hNode = chCfTreeView:TreeView:Nodes:Add(?,?,cParentNodeID,"Root node").

It must be identical, but it is not in FWD, as the latest experiments show:

The following code contains 6 calls, which are effectively should execute the same in runtime:

```
nodes:Add(?, 4, "Key", "Label", img1, img2).
nodes:Add(?, ?, "Key", "Label", img1, img2).
nodes:Add(?, ?, ?, "Label", img1, img2).

nodes:Add(, 4, "Key", "Label", img1, img2).
nodes:Add(, , "Key", "Label", img1, img2).
nodes:Add(, , , "Label", img1, img2).
```

Currently FWD converts them incorrectly and differently:

```
         nodes.unwrapTreeNodeCollection().add(new integer(4), new character("Key"), new character("Label"), im
g1, img2);
         nodes.unwrapTreeNodeCollection().add(new character("Key"), new character("Label"), img1, img2);
         nodes.unwrapTreeNodeCollection().add(new character("Label"), img1, img2);
         nodes.unwrapTreeNodeCollection().add(new integer(), new integer(4), new character("Key"), new charact
er("Label"), img1, img2);
         nodes.unwrapTreeNodeCollection().add(new integer(), new integer(), new character("Key"), new characte
r("Label"), img1, img2);
         nodes.unwrapTreeNodeCollection().add(new integer(), new integer(), new character(), new character("La
bel"), img1, img2);
```

As you may see, OCX method calls with omitting parameters are converted correctly, while the call where unknown literals explicitly passed as parameters, are converted incorrectly: these parameters are just missing in the output!

#12 - 04/28/2022 07:08 PM - Tijs Wickardt

Vladimir Tsichevski wrote:

> Tijs Wickardt wrote:
>
> > The OpenEdge documentation states explicitly that OCX's are 0-based, not 1-based like the 4GL.
>
> The first argument of the original treeview OCX add call is 1-based node index. This means the first tree node you create has the index 1. And this has no any connection with 4gl.

I didn't describe it correctly, sorry about that.
It's up to the OCX if a parameter is 0 or 1 based.

#13 - 04/28/2022 07:29 PM - Vladimir Tsichevski

The .ast file seems to be correct (all parameters in the call are present, but in .jast file parameters for the ? literal are missing. I gather, the problem is in generating Java from .ast.

Can anybody point me to the rule files responsible for generation Java?
Thanks.

#14 - 04/29/2022 03:06 AM - Hynek Cihlar

Vladimir Tsichevski wrote:

> The .ast file seems to be correct (all parameters in the call are present, but in .jast file parameters for the ? literal are missing. I gather, the problem is in generating Java from .ast.
>
> Can anybody point me to the rule files responsible for generation Java?
> Thanks.

Add -Drules.tracing to the ConversionDriver java process. With rules tracing on, you will see the rule file and line on every node or annotation where it originated. This will help you to get to the rules responsible for the argument expressions.

#15 - 04/29/2022 03:25 AM - Hynek Cihlar

Vladimir, is there a customer app failing test case for this? If not I suggest you make this lower priority.

#16 - 05/02/2022 10:26 AM - Vladimir Tsichevski

- Priority changed from Normal to Low

This issue does not appear in known customer scenarios, so its severity reduced to LOW.