User Interface - Bug #5622

TREEVIEW widget issues

08/26/2021 12:47 PM - Vladimir Tsichevski

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

Description

Related issues:

Related to User Interface - Bug #5118: TREELIST widget issues	**WIP**
Related to User Interface - Feature #5151: Add complete support for tree-list...	**WIP**
Related to Conversion Tools - Feature #6302: Improve conversion of empty OCX ...	**WIP**

History

#1 - 08/26/2021 12:47 PM - Vladimir Tsichevski

- Related to Bug #5118: TREELIST widget issues added

#2 - 08/26/2021 12:58 PM - Vladimir Tsichevski

In this task various issues related to the TREEVIEW FWD widget issues, and its difference from TREELIST FWD widget will be listed and resolved.

The TREEVIEW was designed as the replacement for the *Microsoft TreeView Control 6.0 (SP6)* OCX control.
The TREELIST was designed as the replacement for the customer-specific OCX control.

These two original OCXes behave differently in many aspects.

Currently both TREEVIEW and TREELIST FWD implementation have common base classes, so some of the differences are not implemented.

#3 - 08/26/2021 01:05 PM - Vladimir Tsichevski

In original *Microsoft TreeView Control 6.0 (SP6)* OCX control node keys must be unique across the tree. Currently in FWD TREEVIEW they are not. This is a regression.

Proposed fix: add to the tree a boolean flag keysUnique, which is by default TRUE for TREEVIEW and FALSE for TREELIST. Both TREEVIEW and TREELIST will or will not check node keys for uniqueness depending on this flag value.

#4 - 08/26/2021 01:39 PM - Vladimir Tsichevski

In original *Microsoft TreeView Control 6.0 (SP6)* OCX control the **plus/minus button** is visible if and only if **the node has children**. In FWD TREEVIEW the button visibility depends on the hasChildren flag. This flag is required in TREELIST, and it has no analogous feature in MS TreeView.

Proposed fix: when rendering TREEVIEW, show the plus/minus button if and only if the node has children, ignore the hasChildren flag completely.

#6 - 08/27/2021 04:52 PM - Vladimir Tsichevski

The semantic and implementations for tree:Nodes:remove() in FWD do not match that of *Microsoft TreeView Control 6.0 (SP6)* OCX control.
The argument to this call must be the **node** to remove, not the node **index**.

#7 - 08/27/2021 05:19 PM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

> The semantic and implementations for tree:Nodes:remove() in FWD do not match that of *Microsoft TreeView Control 6.0 (SP6)* OCX control.
> The argument to this call must be the **node** to remove, not the node **index**.

False alarm: the argument to the remove() must be either node key or node index.

#8 - 08/31/2021 01:18 PM - Vladimir Tsichevski

Conversion produces invalid Java statement:

```
ASSIGN treeView1:ImageList  = chImage:ImageList.
```

is converted as:

```
new handle(treeView1.unwrapTreeView().setImageList(hImageList.unwrapImageList()));
```

which is invalid Java (the new handle(..) must be removed.

#9 - 08/31/2021 01:27 PM - Vladimir Tsichevski

REJECTED. Conversion/runtime issue: missing method. The 4gl:

```
listimages = images:ListImages
```

is (correctly) converted as

```
listimages.assign(images.unwrapImageListWidget().getListImages());
```

But the method handle:unwrapImageListWidget does not exist.

#10 - 08/31/2021 02:05 PM - Vladimir Tsichevski

Conversion crashes on the following valid 4gl:

```
nodes:Item(nodeIndex):Expanded = FALSE.
```

A workaround exist: extract the node expression to a local variable:

```
DEFINE VARIABLE theNode AS COMPONENT-HANDLE.
theNode = nodes:Item(nodeIndex).
theNode:Expanded = FALSE.
```

#11 - 08/31/2021 02:08 PM - Vladimir Tsichevski

- File treeview.p.ext-hints added

- File treeview.p added

- Status changed from New to WIP

The test application to demonstrate conversion problems above: treeview.p, treeview.p.ext-hints.

This application works well in 4gl.

#12 - 08/31/2021 06:01 PM - Vladimir Tsichevski

Conversion issue: the valid 4gl call:

```
nodes:Add(?, 4, "Root", "Root", img1, img2)
```

is converted as:

```
nodes.unwrapTreeNodeCollection().add(new integer(4), new character("Root"), new character("Root"), img1, img2)
```

So, if the undefined value is passed as the first argument, this argument is completely omitted, so the call has 5 arguments instead of 6.

#13 - 08/31/2021 06:03 PM - Vladimir Tsichevski

- File treeview.p.ext-hints added

Solved: the .ext-hints file was invalid.
The corrected one is attached.

Vladimir Tsichevski wrote:

> REJECTED. Conversion/runtime issue: missing method. The 4gl:
>
> [...]
>
> is (correctly) converted as
> [...]
>
> But the method handle:unwrapImageListWidget does not exist.

#14 - 08/31/2021 06:24 PM - Vladimir Tsichevski

RESOLVED. TreeViewWidget.clearAll invalidates the nodes, which may be referenced by 4gl as the result of ``tree:Nodes``. As the result, it is impossible to add new nodes after this point.

The stack where the bad thing happens:

```
TreeNodeCollectionResource.delete() line: 604
TreeViewNodeResource(TreeNodeResource)._removeCollection() line: 1253
TreeViewWidget(TreeWidgetBase).clearAll() line: 1114
```

The offending line is:

```
this.tree = null;
```

Hynek, what is the idea behind this? Can this behavior be changed, so the result of tree:Nodes stays valid after a call to TreeViewWidget.clearAll? The 4gl does not suffer from this issue.

#15 - 08/31/2021 06:43 PM - Vladimir Tsichevski

The TreeNodeCollectionResource.add(NumberType anchorIndex, NumberType relationType, character key, character text) works incorrectly when the node created should not have parents (is a root node), so the anchorIndex is null.

4gl creates such node, while FWD does not.

#16 - 09/01/2021 01:55 AM - Hynek Cihlar

Vladimir Tsichevski wrote:

> TreeViewWidget.clearAll invalidates the nodes, which may be referenced by 4gl as the result of ``tree:Nodes``. As the result, it is impossible to add new nodes after this point.

getNodes allocates new TreeNodeCollectionResource after clearAll. So using tree:Nodes after clearAll should work. Or are you saying clearAll should not invalidate the nodes already referenced by 4GL?

#17 - 09/01/2021 08:27 AM - Vladimir Tsichevski

Hynek Cihlar wrote:

> Vladimir Tsichevski wrote:
>
>> TreeViewWidget.clearAll invalidates the nodes, which may be referenced by 4gl as the result of ``tree:Nodes``. As the result, it is impossible to add new nodes after this point.
>
> getNodes allocates new TreeNodeCollectionResource after clearAll. So using tree:Nodes after clearAll should work.

Yes, this workaround works.

In fact, this is not a problem for TREEVIEW, since there is no such call in *Microsoft TreeView Control 6.0 (SP6)* OCX control, so with TREEVIEW you should use the tree:Nodes:Clear, the later should be mapped to TreeNodeCollection.clearNodes() method in ocx_conversion.rules, which does node deletion properly.

For TREELIST this as also not a problem, since TREELIST nodelist cannot be addressed as an object in original OCX API.

So this very issue shall be considered solved.

> Or are you saying clearAll should not invalidate the nodes already referenced by 4GL?

Not the nodes, but the nodelist object (the value returned by tree:Nodes). For the sake of compatibility with the original TreeView OCX.
But, ass I wrote above, this clearAll call should never be used with TREEVIEW objects, so this is not a problem.

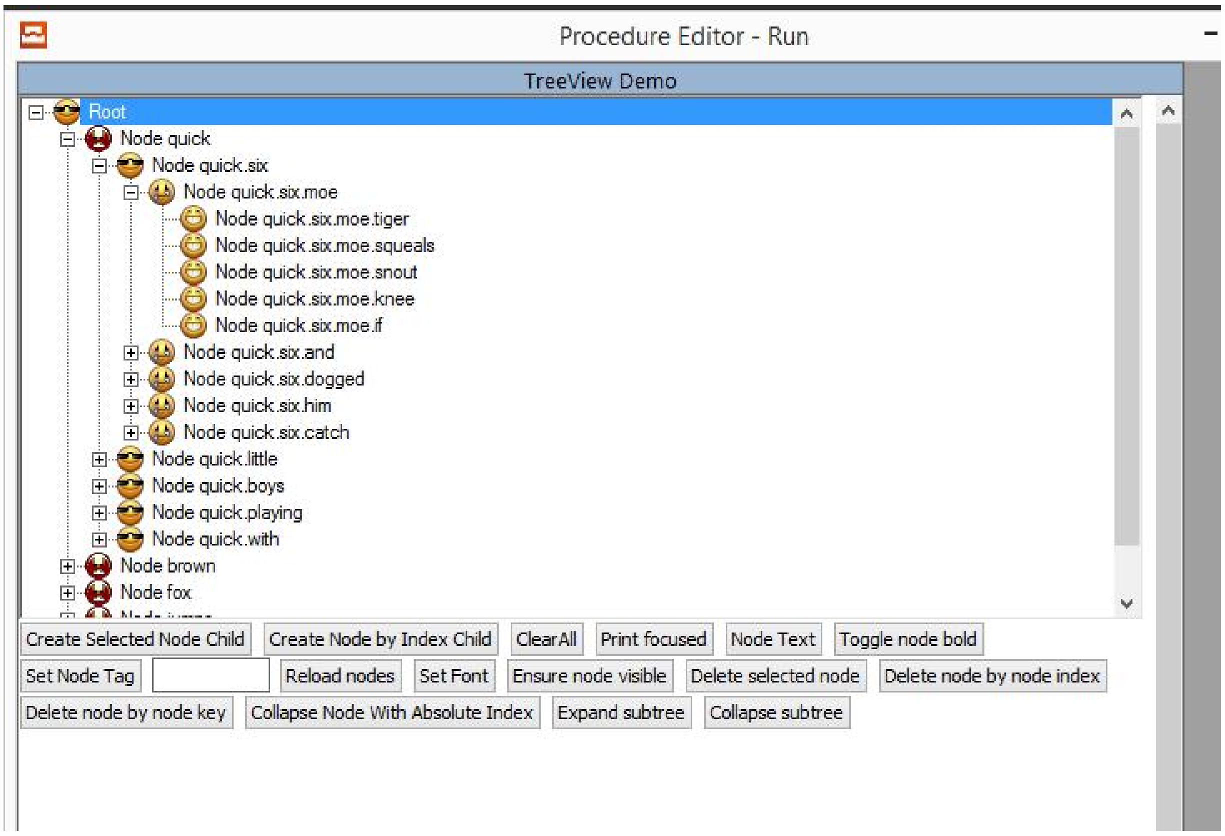
As to particular nodes:

1. Obviously, tree nodes should not be used by the application after they are detached by clearAll, so they **should** be invalidated. The TreeNodeCollectionResource.clearNodes() **does** this, which is correct IMO.
2. The TreeViewWidget.clearAll does **not** use TreeNodeCollectionResource.clearNodes(), and does **not** delete every node, which is **not** correct.
3. This is a TREELIST problem, not a TREEVIEW problem. Some testing should be done to discover how nodes deleted in such manner do behave in TREELIST.

#18 - 09/01/2021 08:34 AM - Vladimir Tsichevski

- File 5622-treeview-demoapp-OE.png added

The treeview demo app reference screen in OE:



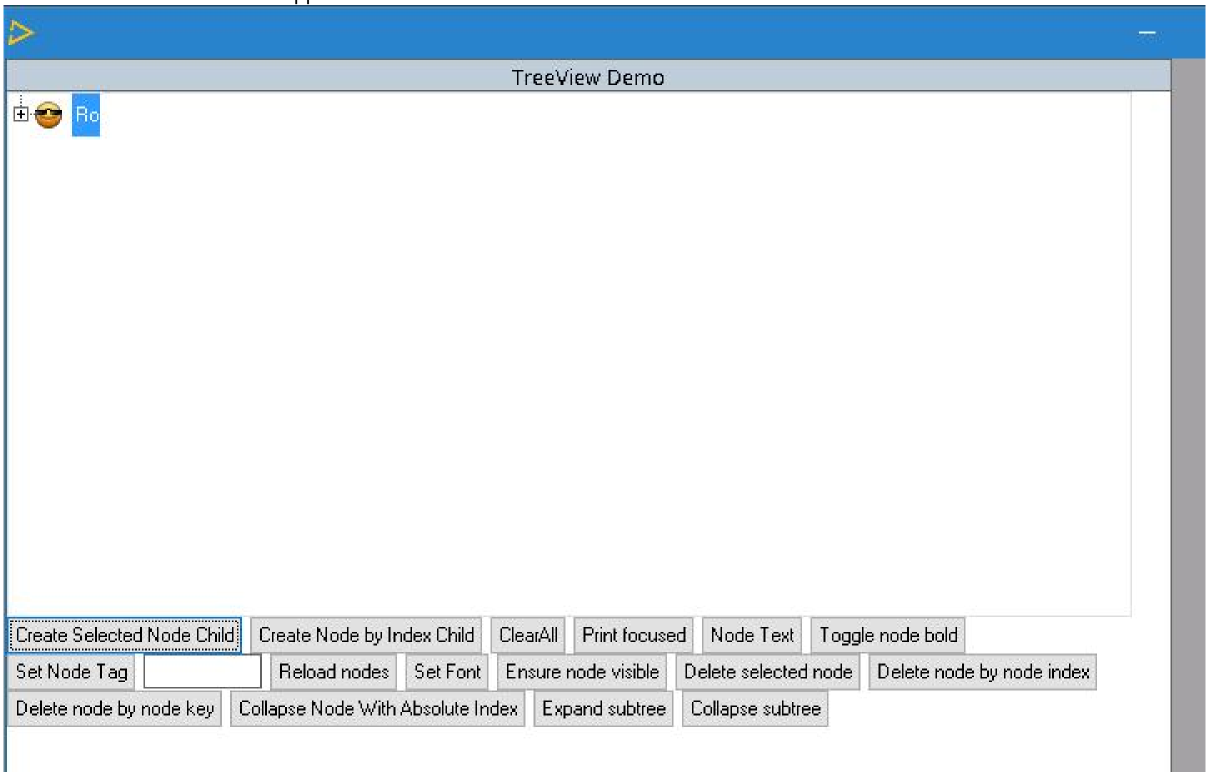
#19 - 09/01/2021 08:38 AM - Vladimir Tsichevski

- File 5622-treeview-demoapp-initial-FWD.png added

Treeview rendering issues:

1. Extra line up from the first root node
2. The node text is clipped from the left.

See the initial treeview demo app screen in FWD:



#20 - 09/01/2021 08:45 AM - Vladimir Tsichevski

SOLVED Scrolling issue:

1. Run treeview.p application

2. Expand enough nodes to make vertical scrollbar visible.
3. Drag the vertical scrollbar button down.
4. An error box is displayed: SCROLL-NODE-COUNT is not a queryable attribute for TREEVIEW widget (4052)

#21 - 09/01/2021 09:32 AM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

> TreeViewWidget.clearAll invalidates the nodes...

the proper tree:Nodes:Clear conversion support added in rev. 12880.

#23 - 09/07/2021 03:31 PM - Vladimir Tsichevski

In 3821c rev. 12897 the following issues has been fixed:

1. Partial (mostly conversion only) support added for TREEVIEW properties: MousePointer, LineStyle and Style, and also for TREEVIEW node properties: Bold, Checked and Tag.
2. The remove TREEVIEW node method now accepts both character node key and integer node index as the argument to match the MS TreeView OCX.

#24 - 09/07/2021 03:56 PM - Vladimir Tsichevski

Hynek, can you please help me?

What is the proper way to set the widget default mouse pointer icon? By this I mean the pointer icon when the mouse pointer hovers the widget.

I tried AbstractWidget.setMousePointer(MousePtrWrapper), but with no visible effect.

#25 - 09/07/2021 04:49 PM - Eugenie Lyzenko

Vladimir,

Does the ocx_conversion.rules require urgent reconversion of big customer application? I need to take my system resource for DB import and want to know how fast we need to see the changes you provided.

Please answer today if possible.

#26 - 09/07/2021 06:19 PM - Vladimir Tsichevski

Eugenie Lyzenko wrote:

> Vladimir,
>
> Does the ocx_conversion.rules require urgent reconversion of big customer application? I need to take my system resource for DB import and want to know how fast we need to see the changes you provided.

Definitely **no**.

> Please answer today if possible.

Sorry for the inconvenience.

#27 - 09/07/2021 06:27 PM - Eugenie Lyzenko

Vladimir Tsichevski wrote:

> Eugenie Lyzenko wrote:
>
>> Vladimir,
>>
>> Does the ocx_conversion.rules require urgent reconversion of big customer application? I need to take my system resource for DB import and want to know how fast we need to see the changes you provided.
>
> Definitely **no**.
>
>> Please answer today if possible.
>
> Sorry for the inconvenience.

No problem. Thanks for reply.

#28 - 09/08/2021 04:22 AM - Hynek Cihlar

Vladimir Tsichevski wrote:

> Hynek, can you please help me?
>
> What is the proper way to set the widget default mouse pointer icon? By this I mean the pointer icon when the mouse pointer hovers the widget.
>
> I tried AbstractWidget.setMousePointer(MousePtrWrapper), but with no visible effect.

According to the sources, the widget you set a mouse pointer for must be "hoverable" to allow to set a custom mouse cursor. That is, it must be registered with Web JS driver to be sensitive to cursor hover. I believe the widget must be enabled with the call to AbstractWidget._setEnabled for this (see the call createMouseHoverAction in the method body).

#29 - 09/08/2021 09:18 AM - Roger Borrello

If needed, there was a conversion of the customer application using 12899. Please see #5034-1173 for details.

#30 - 09/08/2021 02:48 PM - Vladimir Tsichevski

- Related to Feature #5151: Add complete support for tree-list OCX triggers added

#31 - 09/08/2021 05:21 PM - Vladimir Tsichevski

Some of TREEVIEW events expect arguments of INPUT-OUTPUT type. For example, the argument KeyCode of the KeyPress or KeyDown event expects the user can change the key code on return from this event.

The event handling is currently implemented using triggers.
Currently all trigger parameters are implemented as INPUT.

Is it possible to define INPUT-OUTPUT trigger parameters in FWD?

#32 - 09/08/2021 05:31 PM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

> Some of TREEVIEW events expect arguments of INPUT-OUTPUT type. For example, the argument KeyCode of the KeyPress or KeyDown event expects the user can change the key code on return from this event.
>
> The event handling is currently implemented using triggers.
> Currently all trigger parameters are implemented as INPUT.
>
> Is it possible to define INPUT-OUTPUT trigger parameters in FWD?

Just discovered some other events with INPUT-OUTPUT parameters, will try to fix the incorrect even generation.

#33 - 09/09/2021 03:01 AM - Hynek Cihlar

Code review 3821c revision 12897.

Please test how the method remove will behave when a convertibe number is passed to it. I.e. when "1" is assigned to it, will it search a node by its key or will it convert to a number and search by the index.

TreeNodeFace, some of the new methods are declared with native Java types instead of BaseDataType types. Also the newly added properties are not exposed as 4GL legacy attributes.

#34 - 09/10/2021 05:14 PM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

> Scrolling issue:
>
> 1. Run treeview.p application
> 2. Expand enough nodes to make vertical scrollbar visible.
> 3. Drag the vertical scrollbar button down.
> 4. An error box is displayed: SCROLL-NODE-COUNT is not a queryable attribute for TREEVIEW widget (4052)

This was due to defining TREELIST 4gl event handlers for TREEVIEW by mistake. This situation better be checked and forbidden during the conversion.

#35 - 09/10/2021 05:17 PM - Vladimir Tsichevski

Then TREEVIEW loses keyboard focus, the selected node highlight should be removed.

#36 - 09/10/2021 05:19 PM - Vladimir Tsichevski

KeyDown event handler conversion error: the first event argument must be of INPUT-OUTPUT type instead of INPUT, the return value must be used as the actual key.

#37 - 09/10/2021 07:06 PM - Vladimir Tsichevski

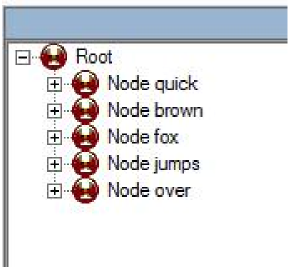
In 3821c rev. 12916:

1. added conversion of DRAGGED-NODE and DRAGGED-OVER-NODE getters (See #5172-81);
2. fixed: KeyDown trigger procedure conversion (see #5622-36);
3. fixed: Keys must be unique in TREEVIEW. See #5622-3;
4. Fixed: mouse coordinates passed to OLE events, must be widget-relative;
5. MOUSEDOWN and MOUSEMOVE events support added.

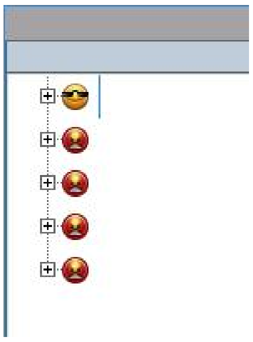
#38 - 09/20/2021 07:51 AM - Vladimir Tsichevski

The EnsureVisible operation is broken for TREEVIEW. Here I call the operation for the first child of the Root node.

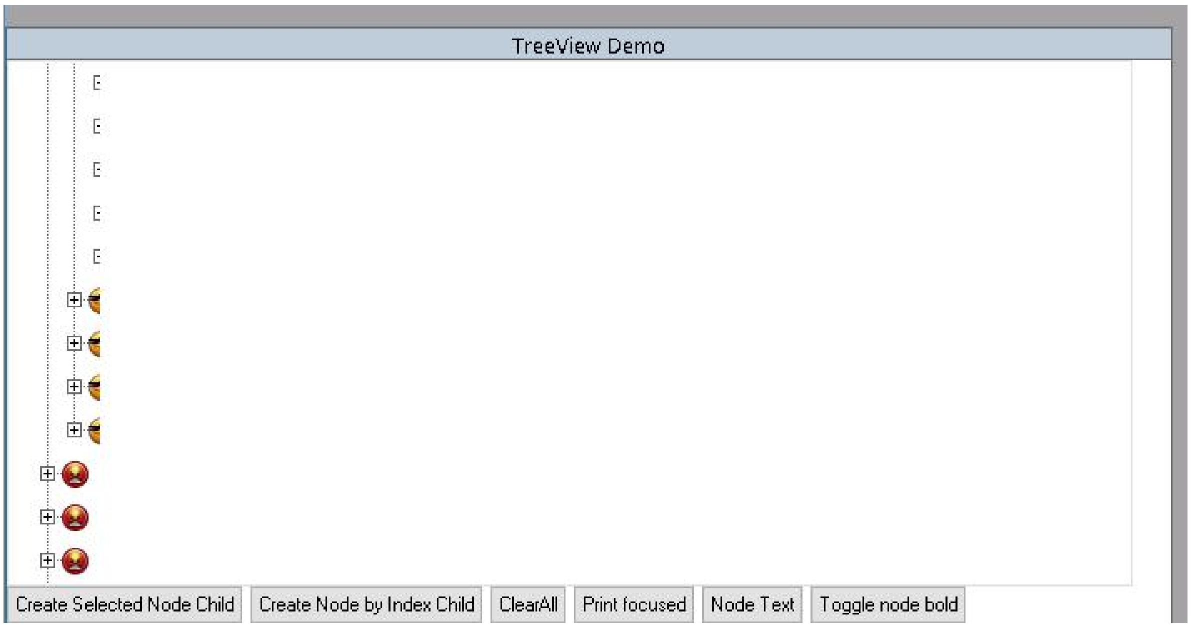
The original OE screen:



The FWD screen:



Also, the scrollbars are not updated after EnsureVisible, and the widget width is not recalculated, so the tree nodes view is trimmed more than usual:



#39 - 09/20/2021 09:57 AM - Vladimir Tsichevski

- File 5622-38-FWD-more-nodes.png added

- File 5622-38-FWD.png added

- File 5622-38-OE.png added

#40 - 09/21/2021 04:53 PM - Vladimir Tsichevski

The Parent setter is not currently implemented.

In OCX, setting the node parent node always makes the node first sibling, even if the parent is not changed.
This makes it possible to re-order node children, which can be used in some customer's applications.

#41 - 09/21/2021 05:59 PM - Vladimir Tsichevski

Node navigation: Next, Previous, FirstSibling, LastSibling. Then applied to the tree root, these methods must return unknown. Now they raise NPE when accessing the node parent.

#42 - 09/21/2021 06:06 PM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

> The Parent setter is not currently implemented.
>
> In OCX, setting the node parent node always makes the node first sibling, even if the parent is not changed.
> This makes it possible to re-order node children, which can be used in some customer's applications.

Implemented in 3821c rev. 12966.

#43 - 09/21/2021 06:06 PM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

> Node navigation: Next, Previous, FirstSibling, LastSibling. Then applied to the tree root, these methods must return unknown. Now they raise NPE when accessing the node parent.

Fixed in 3821c rev. 12966.

#44 - 09/22/2021 11:46 AM - Vladimir Tsichevski

Hynek Cihlar wrote:

> Code review 3821c revision 12897.

> Please test how the method remove will behave when a convertibe number is passed to it. I.e. when "1" is assigned to it, will it search a node by its key or will it convert to a number and search by the index.

Tested. The OCX does not convert text to numbers, an error is reported if I try to use a text which is not a valid key.

> TreeNodeFace, some of the new methods are declared with native Java types instead of BaseDataType types. Also the newly added properties are not exposed as 4GL legacy attributes.

Fixed in 3821c rev. 12975

#45 - 09/22/2021 12:44 PM - Hynek Cihlar

Vladimir Tsichevski wrote:

> Fixed in 3821c rev. 12975

I think the changes are OK, just move the static methods below the constructor.

The next time, please don't mix massive non-functional changes with functional changes. It is almost impossible to do a proper code review.

#46 - 09/22/2021 01:34 PM - Vladimir Tsichevski

Hynek Cihlar wrote:

> Vladimir Tsichevski wrote:
>
>> Fixed in 3821c rev. 12975
>
> I think the changes are OK, just move the static methods below the constructor.

Fixed in 12978.

> The next time, please don't mix massive non-functional changes with functional changes. It is almost impossible to do a proper code review.

I hope, this this was the last Tree-relates class with members unsorted.

#47 - 04/27/2022 02:33 AM - Tijs Wickardt

On Wed 2022-04-20 we parked some treeview issues. See xfer testcases , ocx_to_fwd/mscomctl-treeview-ocx.readme.txt .

Most notably: the hNode:assign does not work in the button click 'Fill Tree' event handler. See workaround/hack in the readme.txt. Investigation aborted due to prioritization.

#48 - 04/27/2022 05:32 AM - Hynek Cihlar

Tijs Wickardt wrote:

> On Wed 2022-04-20 we parked some treeview issues. See xfer testcases , ocx_to_fwd/mscomctl-treeview-ocx.readme.txt .
>
> Most notably: the hNode:assign does not work in the button click 'Fill Tree' event handler. See workaround/hack in the readme.txt. Investigation aborted due to prioritization.

I assume you mean the following statement: hNode = chCfTreeView:TreeView:Nodes:Add(,,cParentNodeID,"Root node").

Did you see any error on the screen or in the client logs?

#49 - 04/27/2022 01:16 PM - Vladimir Tsichevski

Tijs Wickardt wrote:

> On Wed 2022-04-20 we parked some treeview issues. See xfer testcases , ocx_to_fwd/mscomctl-treeview-ocx.readme.txt .
>
> Most notably: the hNode:assign does not work in the button click 'Fill Tree' event handler. See workaround/hack in the readme.txt. Investigation aborted due to prioritization.

See #5622-12. Is the problem you mentioned a compilation problem?

#50 - 04/27/2022 06:17 PM - Vladimir Tsichevski

Another difference: in OCX the first argument to the add procedure is a 1-based node index, and cannot be zero.
In FWD passing zero as the first argument ends with NPE when referencing the parent of the root node, which has no parent by definition.

#51 - 04/27/2022 06:51 PM - Vladimir Tsichevski

Hynek Cihlar wrote:

> I assume you mean the following statement: hNode = chCfTreeView:TreeView:Nodes:Add(,,cParentNodeID,"Root node").

FWD converts this to the following Java statement:

```
               hNode.assign(hTree.unwrapTreeView().getNodes().unwrapTreeNodeCollection().add(new integer(), ne
w integer(), cParentNodeId, new character("Root node")));
```

Which is incorrect: the first two arguments must be default values: 1 and 4.
4gl undefined values must never be passed to OCX.
We can pass Java null as these arguments to denote default values should be used, and replace nulls by default values in Java method implementation.

#52 - 04/27/2022 07:24 PM - Tijs Wickardt

Vladimir Tsichevski wrote:

> Tijs Wickardt wrote:
>
> > On Wed 2022-04-20 we parked some treeview issues. See xfer testcases , ocx_to_fwd/mscomctl-treeview-ocx.readme.txt .
> >
> > Most notably: the hNode:assign does not work in the button click 'Fill Tree' event handler. See workaround/hack in the readme.txt.
> > Investigation aborted due to prioritization.
>
> See #5622-12. Is the problem you mentioned a compilation problem?

No, runtime.

#53 - 04/28/2022 03:27 AM - Hynek Cihlar

Vladimir Tsichevski wrote:

> Hynek Cihlar wrote:
>
> > I assume you mean the following statement: hNode = chCfTreeView:TreeView:Nodes:Add(,,cParentNodeID,"Root node").
>
> FWD converts this to the following Java statement:
>
> [...]
>
> Which is incorrect: the first two arguments must be default values: 1 and 4.

It is on the OCX object to determine what the default value is when an empty argument is given.

> 4gl undefined values must never be passed to OCX.

Why not?

> We can pass Java null as these arguments to denote default values should be used, and replace nulls by default values in Java method implementation.

Currently we use unknown for that, which I agree is most likely incorrect. According to OpenEdge documentation, passing unknown as an OCX argument is converted by OE to COM null value. However there is no mention of how empty arguments get converted. This is something that should be investigated. My guess is an empty argument will be converted to a default (zero) value of the data type declared for the particular parameter.

#54 - 04/28/2022 03:30 AM - Hynek Cihlar

Hynek Cihlar wrote:

> Vladimir Tsichevski wrote:
>
>> Hynek Cihlar wrote:
>>
>>> I assume you mean the following statement: hNode = chCfTreeView:TreeView:Nodes:Add(,,cParentNodeID,"Root node").
>>
>> FWD converts this to the following Java statement:
>>
>> [...]
>>
>> Which is incorrect: the first two arguments must be default values: 1 and 4.
>
> It is on the OCX object to determine what the default value is when an empty argument is given.

By default value I meant how the OCX will cope with an empty argument (converted by OE to whatever "zero" value).

#55 - 04/28/2022 08:30 AM - Vladimir Tsichevski

Hynek Cihlar wrote:

> It is on the OCX object to determine what the default value is when an empty argument is given.

Exactly so. So should our replacement implementation do. And in order to allow it to do that, it should receive some special values meaning "replace this by defaults".
And this value must **not** be the 4gl undefined, because, by my observations, sending undefined to **any** OCX in OE is strictly invalid and causes an error.

>> 4gl undefined values must never be passed to OCX.
>
> Why not?

Because it is incompatible with OE.

>> We can pass Java null as these arguments to denote default values should be used, and replace nulls by default values in Java method

> > implementation.
>
> Currently we use unknown for that, which I agree is most likely incorrect. According to OpenEdge documentation, passing unknown as an OCX argument is converted by OE to COM null value.

By my observations, this always causes an error. At least, in all situations with original TREEVIEW and TREELIST.

> However there is no mention of how empty arguments get converted.

What do you mean by "empty" here? Do you mean empty expression in 4gl call syntax?

> This is something that should be investigated. My guess is an empty argument will be converted to a default (zero) value of the data type declared for the particular parameter.

If you mean empty expression in 4gl call syntax, like in the call to add we are discussing, then your assumption is incorrect: as I already mentioned earlier, this call receives values 1 and 4 (defaults for this OCX call), but not 0 and 0, as you probably assumed.

#56 - 04/28/2022 08:41 AM - Hynek Cihlar

Vladimir Tsichevski wrote:

> Hynek Cihlar wrote:
>
> > It is on the OCX object to determine what the default value is when an empty argument is given.
>
> Exactly so. So should our replacement implementation do. And in order to allow it to do that, it should receive some special values meaning "replace this by defaults".
> And this value must **not** be the 4gl undefined, because, by my observations, sending undefined to **any** OCX in OE is strictly invalid and causes an error.
>
> > > 4gl undefined values must never be passed to OCX.

> > Why not?

> Because it is incompatible with OE.

According to OE documentation it is perfectly valid to pass an unknown value (?). Of course you need the expected parameter type, otherwise you will get a runtime error obviously.

> > > We can pass Java null as these arguments to denote default values should be used, and replace nulls by default values in Java method implementation.

> > Currently we use unknown for that, which I agree is most likely incorrect. According to OpenEdge documentation, passing unknown as an OCX argument is converted by OE to COM null value.

> By my observations, this always causes an error. At least, in all situations with original TREEVIEW and TREELIST.

This makes sense. You should expect a runtime error when you pass ? to an int OCX parameter.

> > However there is no mention of how empty arguments get converted.

> What do you mean by "empty" here? Do you mean empty expression in 4gl call syntax?

Yes.

> > This is something that should be investigated. My guess is an empty argument will be converted to a default (zero) value of the data type declared for the particular parameter.

> If you mean empty expression in 4gl call syntax, like in the call to add we are discussing, then your assumption is incorrect: as I already mentioned earlier, this call receives values 1 and 4 (defaults for this OCX call), but not 0 and 0, as you probably assumed.

I see two explanations here. OE either looks in the type library where default param values can be specified or it passes some default values (zeros), which get interpreted by the OCX component somehow.

#57 - 04/28/2022 08:44 AM - Hynek Cihlar

I created #6302 to cover the empty arguments issue.

#58 - 04/28/2022 09:05 AM - Vladimir Tsichevski

- Related to Feature #6302: Improve conversion of empty OCX arguments added

#59 - 04/28/2022 09:34 AM - Vladimir Tsichevski

Hynek Cihlar wrote:

> > And this value must **not** be the 4gl undefined, because, by my observations, sending undefined to **any** OCX in OE is strictly invalid and causes an error.
> >
> > > 4gl undefined values must never be passed to OCX.
> >
> > > Why not?
> >
> > Because it is incompatible with OE.
>
> According to OE documentation it is perfectly valid to pass an unknown value (?). Of course you need the expected parameter type, otherwise you will get a runtime error obviously.

May be, OE silently converts omitted arguments to null pointers, and the OCXes refuse to accept it?

> > This is something that should be investigated. My guess is an empty argument will be converted to a default (zero) value of the data type declared for the particular parameter.
>
> > If you mean empty expression in 4gl call syntax, like in the call to add we are discussing, then your assumption is incorrect: as I already mentioned earlier, this call receives values 1 and 4 (defaults for this OCX call), but not 0 and 0, as you probably assumed.
>
> I see two explanations here. OE either looks in the type library where default param values can be specified or it passes some default values (zeros), which get interpreted by the OCX component somehow.

Default parameter values are denoted with the defaultvalue expression in IDL. I think, this information can be extracted from the OCX executable.

See https://docs.microsoft.com/en-us/windows/win32/midl/defaultvalue

#60 - 04/28/2022 10:01 AM - Tijs Wickardt

Vladimir Tsichevski wrote:

> > I see two explanations here. OE either looks in the type library where default param values can be specified or it passes some default values (zeros), which get interpreted by the OCX component somehow.
>
> Default parameter values are denoted with the defaultvalue expression in IDL. I think, this information can be extracted from the OCX executable.
>
> See https://docs.microsoft.com/en-us/windows/win32/midl/defaultvalue

Interesting. A possible third explanation: OE stores the default OCX values in the wrx. If the OCX really can't handle a null in itself.

#61 - 04/28/2022 10:35 AM - Vladimir Tsichevski

Tijs Wickardt wrote:

> Interesting. A possible third explanation: OE stores the default OCX values in the wrx. If the OCX really can't handle a null in itself.

I doubt it. WRX files are created by AppBuilder and allow the developer configure the control properties as name/value dictionary (using Property Editor). It does not allow to configure anything else. Considering this, there is no point in keeping the copy of values in WRX files while they can be extracted from the DLL.

Considering this, the answer on this question is not relevant to the issue: our FWD implementation does not depend on it.

#62 - 04/28/2022 10:48 AM - Tijs Wickardt

Vladimir Tsichevski wrote:

> Tijs Wickardt wrote:
>
> > Interesting. A possible third explanation: OE stores the default OCX values in the wrx. If the OCX really can't handle a null in itself.
>
> I doubt it. WRX files are created by AppBuilder and allow the developer configure the control properties as name/value dictionary (using Property

> Editor). It does not allow to configure anything else. Considering this, there is no point in keeping the copy of values in WRX files while they can be extracted from the DLL.
>
> Considering this, the answer on this question is not relevant to the issue: our FWD implementation does not depend on it.

Both remarks: agree. In which case we could probably use null like we currently do at the conversion (or replace it by a sentinel), see: #6302 .

#63 - 07/07/2022 11:34 AM - Vladimir Tsichevski

Unlike the TREELIST, the Parent property applied to the tree visible root, must return an invalid handle.

#64 - 07/07/2022 12:01 PM - Tijs Wickardt

Vladimir Tsichevski wrote:

> Unlike the TREELIST, the Parent property applied to the tree visible root, must return an invalid handle.

You are correct. This is the cause of bug #6481 which I'm working on. The VMA business logic gets a valid handle causing the bug. It should be invalid in this case. The problem seems to stem from the FWD dummy 'parent' node with id==0 . It should be passed to converted 4GL code as an invalid handle.

#65 - 07/07/2022 12:02 PM - Vladimir Tsichevski

Tijs Wickardt wrote:

> Vladimir Tsichevski wrote:
>
>> Unlike the TREELIST, the Parent property applied to the tree visible root, must return an invalid handle.
>
> You are correct. This is the cause of bug #6481 which I'm working on. The VMA business logic gets a valid handle causing the bug. It should be invalid in this case. The problem seems to stem from the FWD dummy 'parent' node with id==0 . It should be passed to converted 4GL code as an invalid handle.

I will fix this.

#68 - 07/07/2022 12:05 PM - Tijs Wickardt

Vladimir Tsichevski wrote:

> I will fix this.

Great, I'll move on to another issue. I'll test the VMA bug after your fix.

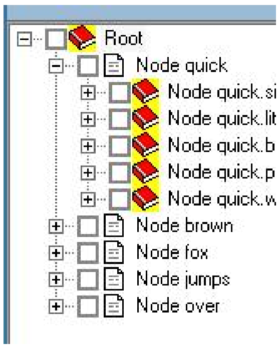
#69 - 07/18/2022 12:06 PM - Vladimir Tsichevski

- File 6244-118-OE.png added

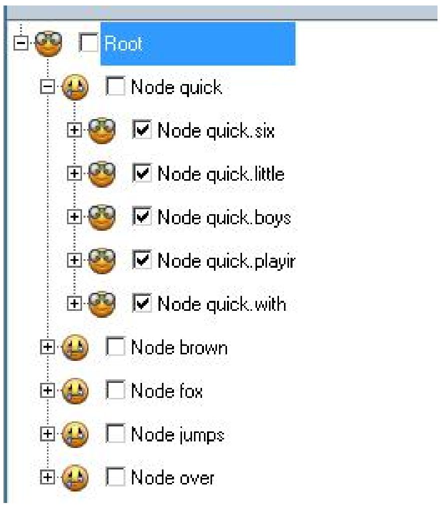
- File 6244-118-FWD.png added

In FWD both icons and checkboxes are visible, but they are displayed the opposite order:

In OE icons are right from checkboxes:



and in FWD icons are left from checkboxes:



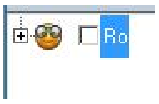
#70 - 07/20/2022 12:51 PM - Vladimir Tsichevski

- File 5622-70.png added

Vladimir Tsichevski wrote:

> In FWD both icons and checkboxes are visible, but they are displayed the opposite order:

Also the text is right clipped after checkboxes are switched ON:



#71 - 07/20/2022 05:01 PM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

> In FWD both icons and checkboxes are visible, but they are displayed the opposite order:
>
> In OE icons are right from checkboxes:
>
> and in FWD icons are left from checkboxes:

Fixed in 3821c rev. 14093.

Files

treeview.p	21.8 KB	08/31/2021	Vladimir Tsichevski
treeview.p.ext-hints	1.37 KB	08/31/2021	Vladimir Tsichevski
treeview.p.ext-hints	1.5 KB	08/31/2021	Vladimir Tsichevski
5622-treeview-demoapp-OE.png	43.8 KB	09/01/2021	Vladimir Tsichevski
5622-treeview-demoapp-initial-FWD.png	7.54 KB	09/01/2021	Vladimir Tsichevski
5622-38-FWD-more-nodes.png	9.1 KB	09/20/2021	Vladimir Tsichevski
5622-38-OE.png	2.65 KB	09/20/2021	Vladimir Tsichevski
5622-38-FWD.png	2.83 KB	09/20/2021	Vladimir Tsichevski

6244-118-OE.png	2.44 KB	07/18/2022	Vladimir Tsichevski
6244-118-FWD.png	6.44 KB	07/18/2022	Vladimir Tsichevski
5622-70.png	1.44 KB	07/20/2022	Vladimir Tsichevski