

## User Interface - Feature #3858

### implement a widget that replaces the Microsoft ProgressBar OCX (VB 6.0 common control)

01/10/2019 12:21 PM - Greg Shah

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Marius Gligor	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>vendor_id:</b>	GCD
<b>billable:</b>	No		
<b>Description</b>			
<b>Related issues:</b>			
Related to Base Language - Feature #3262: implement COM/OLE Automation support			<b>Closed</b>

## History

### #1 - 01/10/2019 12:34 PM - Greg Shah

See <https://docs.microsoft.com/en-us/windows/desktop/controls/progress-bar-control-reference> for documentation. This is a pretty simple progress indicator.

### #2 - 01/14/2019 06:11 PM - Greg Shah

From Hynek:

While the progress bar control is indeed primitive, the actual effort may depend on the visual style(s) we choose to implement. For example one of the visual styles uses gradients to draw the progress bar, AFAIK we don't support gradients in the GUI drivers.

### #3 - 09/13/2019 11:09 AM - Greg Shah

- Assignee set to Marius Gligor

### #4 - 09/16/2019 02:08 AM - Marius Gligor

- Status changed from New to WIP

### #5 - 09/16/2019 12:02 PM - Marius Gligor

- File smart-window.png added

Working on this feature, first I tried to read the 4GL documentation to find out how OCX are handled.

Next I tried to create a simple UAST test using 4gl AppBilder tool as follow:

- First I created a Smart Window.
- Then I tried to insert an OCX widget in the Smart Window frame from widgets palette (button OCX).
- I succeeded to do that for Windows Media Player and Progress Timer Control OCX's for example.
- But when I tried to add a Progress Bar OCX I realized that the object is not in the list of available OCX's!
- I saved the code generated in a .p file for further analyses.

Progress Timer Control is an OCX provided by Progress Software licensed for development.

It seems that in 4gl OCX's must have a valid license in order to be used.

I found an interesting article at the following address <https://knowledgebase.progress.com/articles/Article/P28362> which state:

#### Resolution

Whilst COMCTL32.OCX and MSCOMCTL.OCX are included in the OpenEdge installation, Progress does not provide deve

lopment licenses for them.

Development licenses for these control must be obtained from Microsoft, for example, by installing a Microsoft Development product.

Per Microsoft documentation (see Notes section below), it has been confirmed that up to the Visual Studio 6 series,

the required development license for the version 5 controls are included.

Could be that the reason why AppBuilder does not list those OCX controls in the list of available OCX's? For me is not clear.

When 4gl use an OCX control, a binary file \*.wrx is created in which I found the COM object Class ID and a name.

Example:

{F0B88A90-F5DA-11CF-B545-0020AF6ED35A} PSTimer

Using this file I succeeded to execute the UAST test com-handle/pstime-demo2.w already available in testcases/uast.

This is a useful test but a better one is a test for a Progress Bar OCX widget but unfortunately I cannot create it in my OE VM!

#### #7 - 09/16/2019 02:12 PM - Greg Shah

- Related to Feature #3262: implement COM/OLE Automation support added

#### #8 - 09/16/2019 02:15 PM - Greg Shah

Vladimir recently encountered similar problems, please see #3820 for some useful notes.

I suspect you are right that the MS VB dev tools must be installed to provide that control to the Appbuilder.

#### #9 - 09/17/2019 11:21 AM - Marius Gligor

- File progress-bar.xlsx added

- File not-licensed.png added

1. I installed a MS tool OLE/COM Object Viewer in my VM to discover more about properties, methods and events exposed by registered COM objects.

When I tried to access the ProgresBar OCX control I've got an error which inform me that the control is not licensed for use (see attached picture). Definitely, this is the reason why the MS Common Controls OCX's are not available with AppBuilder.

Searching on the net seems to be a common trouble for developers which are using ActiveX MS Common Controls!

2. I was lucky enough to find out more about ProgressBar OCX in ABL using the PRO\*Tools / COM Object Viewer  
COM objects expose properties, methods and events via interfaces.

Exploring the ProgressBar COM interfaces IProgressbar and IProgressbarEvents I extracted all properties, methods and events exposed using ABL syntax.

PROPERTIES and METHODS exposed by IProgressbar:

Appearance - Returns/sets whether or not controls, Forms or an MDIForm are painted at runtime with 3-D effects.

AppearanceConstants - cc3D=1 ccFlat=0

Property Get: [ Integer-Var = ] <com-handle>:Appearance.

Property Set: <com-handle>:Appearance [ = Integer-Var ].

BorderStyle - Returns/sets border style for an object.

BorderStyleConstants - ccFixedSingle=1 ccNone=0

Property Get: [ Integer-Var = ] <com-handle>:BorderStyle.

Property Set: <com-handle>:BorderStyle [ = Integer-Var ].

Enabled - Returns/sets a value that determines whether a form or control can respond to user-generated events.

Property Get: [ Logical-Var = ] <com-handle>:Enabled.

Property Set: <com-handle>:Enabled [ = Logical-Var ].

Min - Returns/sets a control's minimum value.

Property Get: [ Decimal-Var = ] <com-handle>:Min.

Property Set: <com-handle>:Min [ = Decimal-Var ].

Max - Returns/sets a control's maximum value.

Property Get: [ Decimal-Var = ] <com-handle>:Min.

Property Set: <com-handle>:Min [ = Decimal-Var ].

MouseIcon - Sets a custom mouse icon.

Property Get: [ <user-defined>-Var = ] <com-handle>:MouseIcon.

Property Set: <com-handle>:MouseIcon [ = <user-defined>-Var ].

MousePointer - Returns/sets the type of mouse pointer displayed when over part of an object.

MousePointerConstants - ccArrow=1 ccArrowHourglass=13 ccArrowQuestion=14 ccCross=2 ccCustom=99

ccDefault=0 ccHourglass=11 ccIBeam=3 ccIcon=4 ccNoDrop=12 ccSize=5 ccSizeAll=15 ccSizeEW=9

ccSizeNESW=6 ccSizeNS=7 ccSizeNWSE=8 ccUpArrow=10

Property Get: [ Integer-Var = ] <com-handle>:MousePointer.

Property Set: <com-handle>:MousePointer [ = Integer-Var ].

OLEDrag( ) -Start an OLE drag/drop event with the given control as the source.

NO-RETURN-VALUE <com-handle>:OLEDrag ( ).

OLEDropMode - Returns/Sets whether this control can act as an OLE drop target.

OLEDropConstants - ccOLEDropManual=1 ccOLEDropNone=0

Property Get: [ Integer-Var = ] <com-handle>:OLEDropMode.

Property Set: <com-handle>:OLEDropMode [ = Integer-Var ].

Orientation - Returns/sets a value that determines whether the Progress Bar is displayed vertically or horizontally.

OrientationConstants - ccOrientationHorizontal=0 ccOrientationVertical=1

Property Get: [ Integer-Var = ] <com-handle>:Orientation.

Property Set: <com-handle>:Orientation [ = Integer-Var ].

NO-RETURN-VALUE <com-handle>:Refresh ( )

Scrolling - Returns/sets a value that determines whether the control displays progress with a standard segmented bar or a smooth bar.

ScrollingConstants - ccScrollingSmooth=1 ccScrollingStandard=0

Property Get: [ Integer-Var = ] <com-handle>:Scrolling.

Property Set: <com-handle>:Scrolling [ = Integer-Var ].

Value - Returns or sets a control's current Value property.

Property Get: [ Decimal-Var = ] <com-handle>:Value.

Property Set: <com-handle>:Value [ = Decimal-Var ].

EVENTS exposed by IProgressBarEvents:

PROCEDURE <event-proc-prefix>.Click :

END PROCEDURE.

PROCEDURE <event-proc-prefix>.MouseDown :

DEFINE INPUT PARAMETER Button AS INTEGER.

DEFINE INPUT PARAMETER Shift AS INTEGER.

DEFINE INPUT PARAMETER x AS INTEGER.

DEFINE INPUT PARAMETER y AS INTEGER.

END PROCEDURE.

PROCEDURE <event-proc-prefix>.MouseMove :

DEFINE INPUT PARAMETER Button AS INTEGER.

DEFINE INPUT PARAMETER Shift AS INTEGER.

DEFINE INPUT PARAMETER x AS INTEGER.

DEFINE INPUT PARAMETER y AS INTEGER.

END PROCEDURE.

```

PROCEDURE <event-proc-prefix>.MouseUp :
    DEFINE INPUT PARAMETER Button AS INTEGER.
    DEFINE INPUT PARAMETER Shift AS INTEGER.
    DEFINE INPUT PARAMETER x AS INTEGER.
    DEFINE INPUT PARAMETER y AS INTEGER.
END PROCEDURE.

PROCEDURE <event-proc-prefix>.OLECompleteDrag :
    DEFINE INPUT-OUTPUT PARAMETER Effect AS INTEGER.
END PROCEDURE.

PROCEDURE <event-proc-prefix>.OLEDragDrop :
    DEFINE INPUT-OUTPUT PARAMETER Data AS COM-HANDLE. /* IVBDataObject */
    DEFINE INPUT-OUTPUT PARAMETER Effect AS INTEGER.
    DEFINE INPUT-OUTPUT PARAMETER Button AS INTEGER.
    DEFINE INPUT-OUTPUT PARAMETER Shift AS INTEGER.
    DEFINE INPUT-OUTPUT PARAMETER x AS DECIMAL.
    DEFINE INPUT-OUTPUT PARAMETER y AS DECIMAL.
END PROCEDURE.

PROCEDURE <event-proc-prefix>.OLEDragOver :
    DEFINE INPUT-OUTPUT PARAMETER Data AS COM-HANDLE. /* IVBDataObject */
    DEFINE INPUT-OUTPUT PARAMETER Effect AS INTEGER.
    DEFINE INPUT-OUTPUT PARAMETER Button AS INTEGER.
    DEFINE INPUT-OUTPUT PARAMETER Shift AS INTEGER.
    DEFINE INPUT-OUTPUT PARAMETER x AS DECIMAL.
    DEFINE INPUT-OUTPUT PARAMETER y AS DECIMAL.
    DEFINE INPUT-OUTPUT PARAMETER State AS INTEGER.
END PROCEDURE.

PROCEDURE <event-proc-prefix>.OLEGiveFeedback :
    DEFINE INPUT-OUTPUT PARAMETER Effect AS INTEGER.
    DEFINE INPUT-OUTPUT PARAMETER DefaultCursors AS LOGICAL.
END PROCEDURE.

PROCEDURE <event-proc-prefix>.OLESetData :
    DEFINE INPUT-OUTPUT PARAMETER Data AS COM-HANDLE. /* IVBDataObject */
    DEFINE INPUT-OUTPUT PARAMETER DataFormat AS INTEGER.
END PROCEDURE.

PROCEDURE <event-proc-prefix>.OLEStartDrag :
    DEFINE INPUT-OUTPUT PARAMETER Data AS COM-HANDLE. /* IVBDataObject */
    DEFINE INPUT-OUTPUT PARAMETER AllowedEffects AS INTEGER.
END PROCEDURE.

```

Our implementation should provide all these features.

The ProgressBar OCX is very simple, no color settings, no animation, supporting only partial features of the newer ProgressBar control from .NET Looks like a Win 95 style widget!

The main problem is that we cannot create UAST tests due to license missing for mscomctl.ocx controls.

- A possible solution is to create an UAST on another system which have license for ProgressBar OCX control.

Obviously, this UAST test will not work on our VM but we can use it for conversion, asserting that is successfully compiled in ABL.

- Another solution is to install a MS product, something legal in our VM and obtain a license for those components.

I'm not sure if this could be achieve by installing a trial version of a MS product for example.

The question here is, which product should we install?

- Finally we could try to create an UAST for ProgressBar on our VM even if it will not work properly, but just to help us to fix the conversion.

#### #10 - 09/18/2019 10:17 AM - Marius Gligor

I created an UAST test progressbar-demo.w starting from an existing UAST test for PStime OCX but using the ProgressBar OCX properties, methods and events.

I used the ABL Procedure Editor window option Compile/Check Syntax to validate the code and I've got no errors.

On my WM I found a wrx file at C:\Progress\OE116\_64\gui\webutil\\_probar.wrx which is for a ProgressBar VB 5.0 common control.

Unfortunately using this file we still cannot execute the test due to missing license for MS Common Controls on VM machine.

Next I tried to convert the test in FWD and it was converted without errors because we already have a generic scaffold to handle COM objects implemented by Ovidiu.

However reading the wiki page for TreeView widget implemented by Hynek, I saw that he implemented the OCX like a 4gl widget (FWD extension) which is a substitute for control frame

defining additional rules for conversion in ocx\_conversion.rules

I think that is a best way to follow for all OCX components replacements including the ProgressBar OCX FWD implementation.

#### #11 - 09/19/2019 03:37 AM - Marius Gligor

created task branch 3858a rev 11328

#### #12 - 09/19/2019 01:04 PM - Greg Shah

From Marius:

- I did the changes in the conversion rules for ProgressBar widget following the same rules like already implemented TreeView, HtmlBrowse widgets. My changes has been committed in 3858a/11329
- I created a simple UAST test to create a ProgressBar widget without properties, methods and events.
- I created a <test\_file\_name.p>.ext-hints file for my test as I found in ocx\_conversion.rules.
- I tried to convert my UAST test but did not work due to a NPE!

```
com.goldencode.p2j.pattern.TreeWalkException: ERROR! Active Rule:
```

```
-----  
      RULE REPORT  
-----  
Rule Type :    WALK  
Source AST: [ pbProgBar ] BLOCK/STATEMENT/CREATE_WIDGET/VAR_HANDLE/ @0:0 {12884902245}  
Copy AST  : [ pbProgBar ] BLOCK/STATEMENT/CREATE_WIDGET/VAR_HANDLE/ @0:0 {12884902245}  
Condition : ref.isAnnotation("extent")  
Loop      : false  
--- END RULE REPORT ---
```

```
at com.goldencode.p2j.pattern.PatternEngine.run(PatternEngine.java:1070)  
at com.goldencode.p2j.convert.TransformDriver.processTrees(TransformDriver.java:540)  
at com.goldencode.p2j.convert.ConversionDriver.back(ConversionDriver.java:580)  
at com.goldencode.p2j.convert.TransformDriver.executeJob(TransformDriver.java:874)  
at com.goldencode.p2j.convert.ConversionDriver.main(ConversionDriver.java:983)  
Caused by: java.lang.NullPointerException  
at com.goldencode.expr.CE10663.execute(Unknown Source)  
at com.goldencode.expr.Expression.execute(Expression.java:373)
```

- So far, debugging the code in \*.rules cannot help me to find the cause. Seems that is not from code in ocx\_conversion.rules
- I found that Ovidiu also provide a way to convert generic ocx's using extension hints.
- I tried such a conversion and I succeeded but without honoring the drop-procedure hint.
- He created a custom conversion set of rules special for the PStimer OCX.
- Now I'm a little bit confused. Which is the best way to follow, the rules for TreeView, HtmlBrowse or for a generic ocx?

**#13 - 09/19/2019 01:06 PM - Greg Shah**

I think you should branch from 4124a because the OCX conversion is heavily modified and enhanced in that branch. For example, Ovidiu's previous code is no longer there. We are trying to get it into the trunk ASAP, but you should work "on top" of 4124a for the next few days.

**#14 - 09/20/2019 01:56 AM - Marius Gligor**

I rebased my task branch 3858a from branch 4124a

**#15 - 09/20/2019 08:55 AM - Marius Gligor**

- File *progress\_bar\_test.p.ast* added

Working on top of the branch 4124a I have the same negative experience trying to convert my uast test for progress bar widget from `testcases/uast/progressbar/progress_bar_test.p`. I'm using the following extension hints:

```
<extension>
  <ocx control-frame="CtrlFrame" control-frame-handle="CtrlFrame" com-handle="chCtrlFrame" target-handle="hPr
ogressBar" target-widget="progress-bar" />
  <drop-procedure name="control_load" />
</extension>
```

During the conversion, at one moment I've got the following warning messages:

```
-----
Code Conversion Annotations
-----
```

```
Optional rule set [customer_specific_annotations_prep] not found.
./progressbar/progress_bar_test.p
INFO: FWD OCX Extension: replaced DEFINE VARIABLE [CtrlFrame] with [hProgressBar]
INFO: FWD OCX Extension: replaced CREATE ActiveX [CTRLFRAME] with [hProgressBar]
WARN: FWD OCX Extension: Failed to resolve legacy name for LoadControls. Is LoadControls defined in interface
com.goldencode.p2j.ui.ProgressBar?
WARNING: Variable hProgressBar (refid 12884901891) missing in dictionary [VAR_HANDLE]!
WARNING: Variable hProgressBar (refid 12884901891) missing in dictionary [VAR_HANDLE]!
WARNING: Variable hProgressBar (refid 12884901891) missing in dictionary [VAR_HANDLE]!
WARNING: Variable hProgressBar (refid 12884901891) missing in dictionary [VAR_HANDLE]!
WARNING: Variable hProgressBar (refid 12884901891) missing in dictionary [VAR_HANDLE]!
WARNING: Variable hProgressBar (refid 12884901891) missing in dictionary [VAR_HANDLE]!
WARNING: Variable hProgressBar (refid 12884901891) missing in dictionary [VAR_HANDLE]!
WARNING: Variable hProgressBar (refid 12884901891) missing in dictionary [VAR_HANDLE]!
/home/mag/gcd/testcases/uast/src/com/goldencode/testcases/name_map.xml
Elapsed job time: 00:00:01.807
```

finally I've got the same NPE:

```
-----
Core Code Conversion
-----
```

```
...
Caused by: java.lang.NullPointerException
    at com.goldencode.expr.CE10941.execute(Unknown Source)
```

From where is called `com.goldencode.expr.CE10941.execute()`? I saw different numbers `CExxxx` on each execution!

Finally looking in the `progress_bar_test.p.ast` file (attached file) seems that conversion was done correct and the `CREATE CONTROL-FRAME CtrlFrame` statement is replaced with `CREATE PROGRESS-BAR hProgressBar` statement. Could be the warnings of missing `hProgressBar` ref in dictionary the cause of NPE?

Without extension hints the uast test is successfully converted but without substitution of `CONTROL-FRAME` widget.

**#16 - 09/20/2019 09:43 AM - Vladimir Tsichevski**

Marius, could you please attach the 4gl example, which causes troubles?

I work on another OCX widget, and at the moment I have no problem with converting and running it.

PS: my branch is 3820a, it was created by merging the 4124a, rev. 11329, and the trunk branch at 2019-09-04.

After the merge, I have fixed the COM-related runtime a lot, but I never had any problem with conversion.

**#17 - 09/20/2019 09:52 AM - Marius Gligor**

- File *progress\_bar\_test.p.ext-hints* added

- File *progress\_bar\_test.p* added

Vladimir, I attached both the uast test and ext-hints file.

The uast tests are also committed in testcases project uast/progressbar/progress\_bar\_test.p

Please take a look over my test and please tell me what I'm doing wrong. Thanks.

**#18 - 09/20/2019 10:22 AM - Vladimir Tsichevski**

Marius Gligor wrote:

Vladimir, I attached both the uast test and ext-hints file.

The problem is in the target-widget="progress-bar" in the progress\_bar\_test.p.ext-hints. You probably should ask Hynek, why is this.

You can try the following:

1. Replace the line

```
UIB_S = chCtrlFrame:LoadControls(OCXFile, "CtrlFrame":U).
```

by

```
UIB_S = chCtrlFrame:LoadControls(OCXFile, "MyClass":U).
```

where MyClass is the name of your OCX widget.

2. Remove the progress\_bar\_test.p.ext-hints from your project.

**#19 - 09/20/2019 10:34 AM - Marius Gligor**

Vladimir, many thanks for your feedback.

Without ext-hints file the conversion is done with success but the result is not what I expect.

Basically the conversion is done like in the initial implementation without replacing the CONTROL-FRAME with CREATE <widget>

I need to implement the ProgressBar like a 4Gi widget extension.

I have to talk with Hynek, I'm going to write a mail to him.

**#20 - 09/20/2019 10:42 AM - Vladimir Tsichevski**

Marius Gligor wrote:

I need to implement the ProgressBar like a 4Gi widget extension.

Ok, now I see that.

I have to talk with Hynek, I'm going to write a mail to him.

You probably should.

**#21 - 09/20/2019 10:47 AM - Hynek Cihlar**

Marius, regarding the NPE. Did you extend widgetKwords and widgetClasses maps in ocx\_conversion.rules with the keyword id and the widget class?

**#22 - 09/20/2019 10:54 AM - Marius Gligor**

Hi Hynek, Yes I did. Please take a look over my changes in branch 3858a (at the top)

```
<action>widgetKwords.put("PROGRESS-BAR", prog.kw_prog_bar)</action>  
<action>widgetClasses.put("PROGRESS-BAR", classForName("com.goldencode.p2j.ui.ProgressBar"))</action>
```

Please see my changes in branch 3858a at the top, I did a single commit so far.

**#23 - 09/20/2019 11:28 AM - Hynek Cihlar**

Marius Gligor wrote:



Hi Hynek, Yes I did. Please take a look over my changes in branch 3858a (at the top)

[...]

Please see my changes in branch 3858a at the top, I did a single commit so far.

The problem was with the control frame's com handle variable set to control-frame-handle in the ext-hints file. The value must be set to chCtrlFrame. Also the com-handle value should be something like chCtrlFrame:ProgressBar (depending on the name of the loaded com component in the control frame).

**#24 - 09/20/2019 11:41 AM - Marius Gligor**

Hynek Cihlar wrote:

Marius Gligor wrote:

Hi Hynek, Yes I did. Please take a look over my changes in branch 3858a (at the top)

[...]

Please see my changes in branch 3858a at the top, I did a single commit so far.

The problem was with the control frame's com handle variable set to control-frame-handle in the ext-hints file. The value must be set to chCtrlFrame. Also the com-handle value should be something like chCtrlFrame:ProgressBar (depending on the name of the loaded com component in the control frame).

I did the changes and now the conversion works properly. Thanks once again.

**#25 - 09/24/2019 11:48 AM - Marius Gligor**

I'm working to make changes in the rules for ProgressBar events conversion. So far I did changes for mouse events. I have a question regarding mouse events conversion.

For example ProgressBar OCX define a MouseDown event as follow:

```
PROCEDURE CtrlFrame.ProgressBar.MouseDown :
  DEFINE INPUT PARAMETER Button AS INTEGER.
  DEFINE INPUT PARAMETER Shift AS INTEGER.
  DEFINE INPUT PARAMETER x AS INTEGER.
  DEFINE INPUT PARAMETER y AS INTEGER.
  MESSAGE "MouseDown!".
END PROCEDURE.
```

This is a generic event handler having the mouse button as a parameter.

However in FWD I found specific events for each mouse button: LEFT-MOUSE-DOWN, MIDDLE-MOUSE-DOWN and RIGHT-MOUSE-DOWN

The converted Java code looks like:

```
registerTrigger(new EventList("LEFT-MOUSE-DOWN", hProgressBar), ProgressBarTest1.this, () -> new Trigger()
{
  integer y = TypeFactory.integer();
  integer x = TypeFactory.integer();
  integer shift = TypeFactory.integer();
  integer button = TypeFactory.integer();

  public void body()
  {
    y.assign(hProgressBar.unwrapProgressBar().getMouseY());
    x.assign(hProgressBar.unwrapProgressBar().getMouseX());
    shift.assign(hProgressBar.unwrapProgressBar().getMouseShift());
    button.assign(hProgressBar.unwrapProgressBar().getMouseButton());
    ControlFlowOps.invokeWithMode("CtrlFrame.ProgressBar.MouseDown", "IIII", button, shift, x, y);
  }
});

public void ctrlFrameDotProgressBarDotMouseDown(final integer _button, final integer _shift, final integer
_x, final integer _y)
{
  integer button = UndoableFactory.initInput(_button);
  integer shift = UndoableFactory.initInput(_shift);
  integer x = UndoableFactory.initInput(_x);
  integer y = UndoableFactory.initInput(_y);

  internalProcedure(new Block((Body) () ->
{
  message("MouseDown!");
}));
}
```

The same pattern code apply for other mouse events as well.

The question is:

Should we register triggers for all LEFT-MOUSE-DOWN, MIDDLE-MOUSE-DOWN and RIGHT-MOUSE-DOWN events or we could define a generic MOUSE-DOWN event?

**#26 - 09/24/2019 12:06 PM - Marius Gligor**

- % Done changed from 0 to 10

**#27 - 09/25/2019 02:46 AM - Marius Gligor**

created new task branch 3858b from trunk and cherry-picked changes for PROGRESS-BAR widget implementation from 3858a

**#28 - 09/25/2019 03:41 AM - Marius Gligor**

Task branch 3858b has been rebased with trunk 11330, new revision is 11337.

Task branch 3858a has been archived, moved to merged folder. I'm not sure but maybe the dead folder is the right place for this branch.

**#29 - 09/25/2019 08:09 AM - Greg Shah**

I moved 3858a to dead from merged.

**#30 - 09/25/2019 11:30 AM - Marius Gligor**

I've fixed a bug in OCX conversion for events regarding to procedure parameters type which can be INPUT, OUTPUT or INPUT-OUTPUT. When the event handler is called in the converted code, the mode argument should have correct values: I, O or U.

The mode parameter is set depending on the parameter type.

```
ControlFlowOps.invokeWithMode("CtrlFrame.ProgressBar.OLECompleteDrag", "U", effect);
ControlFlowOps.invokeWithMode("CtrlFrame.ProgressBar.MouseUp", "IIII", button, shift, x, y);
```

**#31 - 09/26/2019 12:29 PM - Marius Gligor**

- % Done changed from 10 to 20

**#32 - 10/11/2019 12:02 PM - Marius Gligor**

- % Done changed from 20 to 30

- File progress-bar-classic-ui-theme.zip added

I attached some snapshots for an early implementation of PROGRESS-BAR widget, Classic UI Theme. The corresponding uast tests:

progressbar/progress\_bar\_test2.p for horizontal progress bar

progressbar/progress\_bar\_test3.p for vertical progress bar

**#33 - 10/14/2019 11:29 AM - Marius Gligor**

- File progress-bar-ui-themes.zip added

1. I implemented the drawing code for progress bar for available UI themes.

Without having the possibility to do a Progress Bar OCX test in OE, so far I'm inspired by some pictures found in the Internet.

Is this OK or we have other options to find out the layout of progress bar in each UI theme?

2. Next, I added Mouse Events support to Progress Bar widget. Here I found a conversion issue. Let me to explain:

Bellow is snippet code form my uast test:

```
/* Progress Bar events */
PROCEDURE CtrlFrame.ProgressBar.Click :
    MESSAGE "Click!".
```

```

END PROCEDURE.

MAIN-BLOCK:
DO ON ERROR UNDO MAIN-BLOCK, LEAVE MAIN-BLOCK
ON END-KEY UNDO MAIN-BLOCK, LEAVE MAIN-BLOCK:

RUN setup_control.
RUN enable_UI.
IF NOT THIS-PROCEDURE:PERSISTENT THEN
WAIT-FOR CLOSE OF THIS-PROCEDURE.
END.

```

The ProgressBar events are converted into triggers and the generated Java code looks like:

```

    OnPhrase[] onPhrase0 = new OnPhrase[]
    {
        new OnPhrase(Condition.ERROR, Action.LEAVE, "mainBlock"),
        new OnPhrase(Condition.ENDKEY, Action.LEAVE, "mainBlock")
    };

doBlock(TransactionType.SUB, "mainBlock", onPhrase0, new Block((Body) () ->
{
    ControlFlowOps.invoke("setup_control");
    ControlFlowOps.invoke("enable_UI");

    if (!_not(thisProcedure().unwrapPersistableProcedure().isPersistent()))
    {
        LogicalTerminal.waitFor(null, new EventList("CLOSE", thisProcedure()));
    }
}));

registerTrigger(new EventList("OCX-MOUSE-CLICK", hProgressBar), ProgressBarTest2.this, new Trigger((B
ody) () ->
{
    ControlFlowOps.invoke("CtrlFrame.ProgressBar.Click");
}));

```

The converted Java code is OK except that the ProgressBar triggers are registered after executing doBlock which wait for application exit.

```
LogicalTerminal.waitFor(null, new EventList("CLOSE", thisProcedure()))
```

As a result the ProgressBar mouse triggers are not registered and cannot be invoked from client side! The triggers registration MUST be done before doBlock

I moved "by hand" the doBlock statement at the end, letting progress bar events to be registered. Then, I had rebuild the testcases.jar and when I tested I found that everything works properly now. I have to check the conversion rules and found a way to fix that.

3. In the widget class I often found a pattern like:

```

if (!config.realized)
{
    if (config.prop == value)
    {
        return;
    }

    config.prop = value;
    pushWidgetAttr("prop", config.prop);
}
else
{
    ErrorManager.recordOrShowError(4053, String.format(
        "Unable to set PROP because the %s %s has been realised", type(),
        widgetName()), false);
}

```

which does not allow to sent the value downstream to client.  
But some properties MUST be sent downstream to client, case on which I changed the code as follow:

```
if (config.prop == value)
{
    return;
}

config.prop = value;
pushWidgetAttr("prop", config.prop);
```

My question is how we decide which properties can be pushed and which not?

#### #34 - 10/14/2019 12:15 PM - Hynek Cihlar

- Assignee changed from Marius Gligor to Hynek Cihlar

- Tracker changed from Feature to Bug

- % Done changed from 30 to 0

Marius Gligor wrote:

2. Next, I added Mouse Events support to Progress Bar widget. Here I found a conversion issue. Let me to explain:  
Bellow is snippet code form my uast test:

[...]

The ProgressBar events are converted into triggers and the generated Java code looks like:

[...]

Btw, why don't you use the standard MOUSE-CLICKED event?

3. In the widget class I often found a pattern like:

[...]

which does not allow to sent the value downstream to client.

But some properties MUST be sent downstream to client, case on which I changed the code as follow:

[...]

My question is how we decide which properties can be pushed and which not?

This is specific to each attribute. Some can be changed after widget is realized, some not. This comes from the limitations of the widget implementations in OpenEdge.

**#35 - 10/14/2019 12:17 PM - Hynek Cihlar**

- % Done changed from 0 to 30
- Tracker changed from Bug to Feature
- Assignee changed from Hynek Cihlar to Marius Gligor

**#36 - 10/15/2019 02:28 AM - Marius Gligor**

Hynek Cihlar wrote:

Marius Gligor wrote:

2. Next, I added Mouse Events support to Progress Bar widget. Here I found a conversion issue. Let me to explain:  
Bellow is snippet code form my uast test:

[...]

The ProgressBar events are converted into triggers and the generated Java code looks like:

[...]

Btw, why don't you use the standard MOUSE-CLICKED event?

Progress bar OCX mouse events are generic, the mouse button is an input parameter, that's why I need a generic MOUSE-DOWN event for example, instead specific LEFT-MOUSE-DOWN, MIDDLE-MOUSE-DOWN and RIGHT-MOUSE-DOWN events. I found no such generic events defined in FWD yet, that's why I defined new generic events OCX-MOUSE-XXX. I could remove the OCX\_ prefix to make the definition even more generic. Please see 9 and 25 notes.

**#37 - 10/15/2019 03:26 AM - Hynek Cihlar**

Marius Gligor wrote:

Progress bar OCX mouse events are generic, the mouse button is an input parameter, that's why I need a generic MOUSE-DOWN event for example, instead specific LEFT-MOUSE-DOWN, MIDDLE-MOUSE-DOWN and RIGHT-MOUSE-DOWN events. I found no such generic events defined in FWD yet, that's why I defined new generic events OCX-MOUSE-XXX. I could remove the OCX\_ prefix to make the definition even more generic. Please see 9 and 25 notes.

I think it makes sense to introduce the generic mouse event. To read the async mouse and key states put a suitable set of methods in a place that can be accessed by any widget (or eventually 4gl logic), possibly to LogicalTerminal.

Also see `KeyReader.getLastKeyState` and `KeyReader.getLastKey`. These should already return the last key states.

#### #38 - 10/15/2019 01:45 PM - Greg Shah

I could remove the `OCX_` prefix to make the definition even more generic.

Yes, this would be good.

#### #39 - 10/17/2019 11:52 AM - Marius Gligor

- % Done changed from 30 to 60

#### #40 - 10/18/2019 08:39 AM - Marius Gligor

Some OCX Progress Bar event handlers for Drag & Drop operations are using a parameter of type COM-HANDLE:

```
PROCEDURE CtrlFrame.ProgressBar.OLEStartDrag :
    DEFINE INPUT-OUTPUT PARAMETER Data AS COM-HANDLE. /* IVBDataObject */
    DEFINE INPUT-OUTPUT PARAMETER AllowedEffects AS INTEGER.
END PROCEDURE.
```

The Data AS COM-HANDLE is a handle of a IVBDataObject COM Interface. Searching over Internet I found a VB description of this interface:

```
' Interface Name : IVBDataObject
' Class Name : DataObject
' ClassID : $CLSID_DataObject
' This Interface cannot be created directly it can only
' be returned by a Method or Property in this library.
Interface IVBDataObject $IID_IVBDataObject
    Inherit IDispatch

    Method Clear <1> ()
    Method GetData <2> (Byval sFormat As Integer) As Variant
    Method GetFormat <3> (Byval sFormat As Integer) As Integer
    Method SetData <4> (Opt Byval vValue As Variant, Opt Byval vFormat As Variant)
    Property Get Files <5> () As IVBDataObjectFiles
End Interface

' Interface Name : IVBDataObjectFiles
' Class Name : DataObjectFiles
' ClassID : $CLSID_DataObjectFiles
' This Interface cannot be created directly it can only
' be returned by a Method or Property in this library.
Interface IVBDataObjectFiles $IID_IVBDataObjectFiles
    Inherit IDispatch

    Property Get Item <0> (ByVal lIndex As Long) As WString
    Property Get Count <1> () As Long
    Method Add <2> (Byval bstrFilename As WString, Opt Byval vIndex As Variant)
    Method Clear <3> ()
    Method Remove <4> (Byval vIndex As Variant)
    Method Meth__NewEnum <-4> () As IUnknown
End Interface
```

Actually they are 2 COM interfaces IVBDataObject and IVBDataObjectFiles Those interfaces are used as a storage for Drag & Drop operations. My question is: Should we implement these interfaces as FWD ComObject objects and accessed via comhandle or we have to find another solution in which use handle instead comhandle?  
I already added an option to replace the comhandle with handle for those parameters. Using the handle we have to implements COM interfaces inside the widget code.

**#41 - 10/18/2019 09:26 AM - Hynek Cihlar**

Marius Gligor wrote:

My question is: Should we implement these interfaces as FWD ComObject objects and accessed via comhandle or we have to find another solution in which use handle instead comhandle?  
I already added an option to replace the comhandle with handle for those parameters. Using the handle we have to implements COM interfaces inside the widget code.

I faced the same problem when I was working on conversion of MS Tree COM control. Among others the control exposes Node sub-object. I converted this to its own legacy resource. See `TreeViewNode`, `LegacyResource.TREEVIEW_NODE`, `TreeViewNodeResource` and related classes. This way the Node comhandle converts to handle and is accessed as any other legacy resource.

The problem you will face is that the handle must be unwrapped to a type that must be known during conversion. Currently the unwrapping happens based on the accessed property or method name. The names in your interfaces are pretty generic, so you may get conflicts. You will have to either obfuscate the method/prop names or derive new interfaces to capture the conflicting methods/props.

**#42 - 10/18/2019 09:47 AM - Marius Gligor**

Hynek Cihlar wrote:

Marius Gligor wrote:

My question is: Should we implement these interfaces as FWD ComObject objects and accessed via comhandle or we have to find another solution in which use handle instead comhandle?  
I already added an option to replace the comhandle with handle for those parameters. Using the handle we have to implements COM interfaces inside the widget code.

I faced the same problem when I was working on conversion of MS Tree COM control. Among others the control exposes Node sub-object. I converted this to its own legacy resource. See `TreeViewNode`, `LegacyResource.TREEVIEW_NODE`, `TreeViewNodeResource` and related classes. This way the Node comhandle converts to handle and is accessed as any other legacy resource.

The problem you will face is that the handle must be unwrapped to a type that must be known during conversion. Currently the unwrapping happens based on the accessed property or method name. The names in your interfaces are pretty generic, so you may get conflicts. You will



have to either obfuscate the method/prop names or derive new interfaces to capture the conflicting methods/props.

Hynek, many thanks for your hints. I'm looking to use a similar solution thinking that IVBDataObject and IVBDataObjectFiles COM interfaces are common for all ActiveX which support Drag & Drop operations, not only for Progress Bar.

#### #43 - 10/21/2019 11:18 AM - Marius Gligor

- % Done changed from 60 to 70

IVBDataObject and IVBDataObjectFiles are COM interfaces used to implement Drag & Drop.

1. On server side I implemented the interfaces as a ComObject extension wrapped in a comhandle due to the following advantages.

- very easy to implement.
- no need to define rules to substitute comhandle with handle in conversion.
- no need to unwrap comhandle to a specific object because comhandle is implemented using Java reflection.

2. On client side I found the DragDropHelper class which is basically a partial implementation of IVBDataObjectFiles. As I understood p4gl support only files Drag & Drop.

I added missing methods in DragDropHelper to make it 100% IVBDataObjectFiles compliant, then I exported the missing methods to server side (ClientExports)

The implementation of IVBDataObjectFiles interface is now very simple, all server side calls on this interfaces are delegated to client side DragDropHelper interface.

On server side, for IVBDataObject I provided a naive implementation since p4gl support only files Drag & Drop.

IVBDataObject is important because the IVBDataObjectFiles interface instance is obtained from IVBDataObject files() method.

The implementation changes has been committed in branch 3858b rev 11369

#### #44 - 10/25/2019 09:42 AM - Marius Gligor

- File progressbar.p.ast.zip added

I'm struggle to convert my uast test with Hotel GUI application but did not work. Doing many tests and conversions finally I found that the cause is my rule to remove/insert main block in ocx\_conversion.rules file.

So I eliminated these rules and doing a conversion everything works fine, the conversion and execution in Swing GUI and Web GUI clients.

However the converted Java code is not accurate because the code for registering Progress Bar triggers is inserted by conversion after the main block.

As a consequence the triggers are registered after application is terminated which is too late, because cannot be fired by application! That's why I added the rules to remove insert main block. The conversion works fine but not with Hotel GUI application.

In hotel GUI application conversion an Ast node for a FUNCTION named fwdGetADMVersion is injected in the AST tree.

The generated Java code for this node is:

```
public character fwdGetAdmversion()  
{  
    return function(this, "fwdGetADMVersion", character.class, new Block((Body) () ->
```

```
{
    returnNormal("ADM2.2");
});
}
```

If my remove/insert rules are in place I found a wrong AST node for fwdGetADMVersion injected in the .ast file (see progressbar.p.bad.ast) ? Comparing with the expected value (see progressbar.p.ast) some annotations are missing for example javaname and classname Later when the ast is converted into Java code a NPE is raised:

```
[java] ftype = ftype.toUpperCase()
[java]         ^ { java.lang.NullPointerException }
```

I tried to fix that but so far I didn't succeed.

A potential solution is to change the conversion rules by eliminating the remove/insert but we have to insert the nodes for triggers before of main node.

It's possible to insert an AST node before another node, when both are siblings nodes?

#### #45 - 10/25/2019 10:06 AM - Hynek Cihlar

Marius Gligor wrote:

It's possible to insert an AST node before another node, when both are siblings nodes?

You can move a node with Aast.move().

#### #46 - 10/25/2019 11:25 AM - Marius Gligor

- % Done changed from 70 to 80

Hynek Cihlar wrote:

Marius Gligor wrote:

It's possible to insert an AST node before another node, when both are siblings nodes?

You can move a node with `Aast.move()`.

I eliminated the remove/insert rules and I changed the trigger creation rule to move before main block when created. It works fine in normal conversion but when I tried with Hotel GUI for an unknown reason trigger nodes are not moved! I'll try to find out why and how to fix it. Definitely, using `Aast.move()@` is from far a better solution because now the GUI hotel conversion and build succeeded.

Actually I found an error in Hotel GUI conversion but not a fatal one: `/abl/guests-frame.w:4519:10: unexpected token: export-report-html`

#### #47 - 10/27/2019 01:00 PM - Constantin Asofiei

There might be issues in Hotel GUI, this is meant (for now) to work with 4069a.

#### #48 - 10/29/2019 11:25 AM - Marius Gligor

- File `progress_bar_test6.p` added

Some thoughts regarding PROGRESS-BAR implementation.

PROGRESS-BAR widget is designed to replace the Microsoft ProgressBar OCX (VB 6.0 common control). It's implemented like a standard widget as a 4gl extension.

1. The conversion from OCX is done using `ocx_conversion.rule` rules. The name and the properties are the same as the OCX control.

```
chCtrlFrame:ProgressBar:Orientation = 0.  
chCtrlFrame:ProgressBar:Scrolling = 1.  
chCtrlFrame:ProgressBar:Max = 100.  
chCtrlFrame:ProgressBar:Min = 0.  
chCtrlFrame:ProgressBar:Value = 30.  
chCtrlFrame:ProgressBar:Refresh( ).
```

These are converted into appropriate Java calls without using `progress.g` definitions.

2. PROGRESS-BAR is an extended 4gl widget and can be used like any other 4gl widget.

```
define variable progBar as handle no-undo.
```

```
create progress-bar progBar  
  assign  
    frame = frame f:handle  
    row    = 3.62  
    column = 5  
    height = 1.2  
    width  = 104  
    visible = true.
```

Normally, the attributes and methods should be used like:

```
progBar:Orientation = 0.  
progBar:Scrolling = 1.  
progBar:Max = 100.  
progBar:Min = 0.  
progBar:Value = 30.  
progBar:Refresh( ).
```

Unfortunately this is not possible for all arguments and methods due to collision with already defined or reserved tokens. For example Scrolling Value Refresh( ) tokens are already defined in progress.g and cannot be used/reused. As a solution I used a prefix pb- (abbreviation for Progress Bar) in front of attributes and methods named which basically define unique extended tokens.

```
progBar:pb-Orientation = 0.  
progBar:pb-Scrolling = 1.  
progBar:pb-Max = 100.  
progBar:pb-Min = 0.  
progBar:pb-Value = 30.  
progBar:pb-Refresh( ).
```

Finally with these tokens in place, I designed a new uast test uast/progressbar/progress\_bar\_test6.p (attached), which looks like a normal 4gl procedure file and is completed converted into Java code. The idea was to create this uast test having the same behavior like uast/progressbar/progress\_bar\_test4.p uast test which is converted from an OCX.

progress\_bar\_test6.p and progress\_bar\_test4.p converted codes are virtual equivalent. Even the Java generated code differs on how event triggers are called, the execution behavior is the same.

#### #49 - 10/31/2019 10:38 AM - Marius Gligor

1. I've finished to write the PROGRESS-BAR widget documentation ([Progress Indicator Widget and OCX Replacement](#)).

I have a small observation regarding conversion extension hints. For TREEVIEW is specified:

```
<extension>  
<ocx control-frame="CtrlFrame" control-frame-handle="chCtrlFrame" control="tree" target-handle="hTree" />  
<drop-procedure name="control_load" />
```

```
</extension>
```

But for PROGRESS-BAR I used:

```
<extension>
  <ocx control-frame="CtrlFrame" control-frame-handle="chCtrlFrame" com-handle="chCtrlFrame:ProgressBar" target-handle="hProgressBar" target-widget="progress-bar" />
  <drop-procedure name="control_load" />
</extension>
```

The difference is control attribute for TREEVIEW but target-widget for PROGRESS-BAR!

Looking inside ocx\_conversion.rule file I found:

```
<!-- gather the list of OCXs-->
<rule>xmlNode.text.equals("ocx")
  <action>cfName = xml.getAttributeValue(xmlNode, "control-frame")</action>
  <action>cfhName = xml.getAttributeValue(xmlNode, "control-frame-handle")</action>
  <action>chName = xml.getAttributeValue(xmlNode, "com-handle")</action>
  <action>wType = xml.getAttributeValue(xmlNode, "target-widget")</action>
  <action>thName = xml.getAttributeValue(xmlNode, "target-handle")</action>
```

Seems that in the mean time control attribute has been replaced by target-widget. Maybe we have to update the documentation for TREEVIEW as well.

2. Searching in the VM AsiGUI I found a progress bar OCX procedure \_probar.w together with a \_probar.wrx file. I created and successfully converted an uast test from this one.

Unfortunately looking over the code I found that is a very simple use of Progress Bar OCX, default properties and only update the Value property. No mouse events, no OLE Drag & Drop events. Progress Bar OCX widget is very simple and perhaps minimum features are used.

The implementation for PROGRESS-BAR widget in branch 3858b is ready for a first code review.

**#50 - 11/01/2019 06:23 PM - Hynek Cihlar**

Marius, note that drop-procedure is no longer part of the conversion logic. Removing the init method was problematic, there were cases where the ocxes were loaded from arbitrary places and where the init method contained additional useful statements.

The OCX loading logic must be currently guarded by FWD-VERSION preprocessor variable.

**#51 - 11/06/2019 08:37 AM - Greg Shah**

Hynek: Please do a code review.

**#52 - 12/04/2019 08:07 AM - Vladimir Tsichevski**

- File error.log added

I was trying to convert and run the progressbar demo application:

1. Obtained the latest 3856b branch for conversion and runtime;
2. Obtained the latest demo from testcases/uast/progressbar (progressbar-demo.w and progress\_bar\_test.p.ext-hints);
3. Created an empty dummy pbar.wrx file;

I've got a NPE when converting (see the log attached).

**#53 - 12/04/2019 12:32 PM - Marius Gligor**

Vladimir Tsichevski wrote:

I was trying to convert and run the progressbar demo application:

1. Obtained the latest 3856b branch for conversion and runtime;
2. Obtained the latest demo from testcases/uast/progressbar (progressbar-demo.w and progress\_bar\_test.p.ext-hints);
3. Created an empty dummy pbar.wrx file;

I've got a NPE when converting (see the log attached).

I successfully converted following tests:

```
./image-list/image_list_test1.p
./image-list/image_list_test2.p
./image-list/image_list_test3.p
./image-list/image_list_test4.p
./progressbar/progress_bar_test1.p
./progressbar/progress_bar_test2.p
./progressbar/progress_bar_test3.p
./progressbar/progress_bar_test4.p
./progressbar/progress_bar_test5.p
./progressbar/progress_bar_test6.p
./progressbar/progressbar-demo.w
```

prepare:

```
[mkdir] Created dir: /home/mag/gcd/testcases/uast/build
[mkdir] Created dir: /home/mag/gcd/testcases/uast/build/lib
[mkdir] Created dir: /home/mag/gcd/testcases/uast/build/classes
[copy] Copying 1 file to /home/mag/gcd/testcases/uast/build/classes
[copy] Copying 17 files to /home/mag/gcd/testcases/uast/build/classes
[copy] Copying 2 files to /home/mag/gcd/testcases/uast/build/classes
[copy] Copying 1 file to /home/mag/gcd/testcases/uast/build/classes
```

```
[copy] Copying 3 files to /home/mag/gcd/testcases/uast/build/classes/cfg
[copy] Copying 4 files to /home/mag/gcd/testcases/uast/build/classes/data

compile:
[javac] Compiling 50 source files to /home/mag/gcd/testcases/uast/build/classes
[echo] Compiling Java Sources Done.

jar:
[jar] Building jar: /home/mag/gcd/testcases/uast/build/lib/testcases.jar

BUILD SUCCESSFUL
Total time: 53 seconds
mag@mag-N751JK:~/gcd$
```

Please use these tests. You should have both <test>.p and <test>.p.ext-hints files. You don't need to create a .wrx file.

Please check if you have the correct link to p2j project in testcases/uast  
In -s \$HOME/<your branch location>/3856b/ p2j  
Using a wrong link could cause a NPE exception.  
Also do a build of 3856b branch before starting the conversion.

#### #54 - 12/04/2019 04:44 PM - Vladimir Tsichevski

Marius Gligor wrote:

Please use these tests. You should have both <test>.p and <test>.p.ext-hints files. You don't need to create a .wrx file.

Please check if you have the correct link to p2j project in testcases/uast  
In -s \$HOME/<your branch location>/3856b/ p2j  
Using a wrong link could cause a NPE exception.  
Also do a build of 3856b branch before starting the conversion.

The progressbar-demo.w.ext-hints is missing in the repository. I copied it from other hint file in the same directory first, but it was a mistake, since the name of the handle in progressbar-demo.w differs from that in all other tests.

I fixed the name in the hint file, and now the demo compiles Ok.

I think, you just forgot to add this hint file to the repository.

## #55 - 12/04/2019 04:59 PM - Vladimir Tsichevski

For the progressbar-demo.w conversion produces uncomilable Java:

The line:

```
RELEASE OBJECT pbCOM.
```

converts to:

```
ComServer.release(hProgressBar);
```

which causes the following compilation error:

```
[javac] /home/vvt/Projects/test3856b/src/com/test3856b/Test.java:58: error: incompatible types: handle can
not be converted to comhandle
[javac]         ComServer.release(hProgressBar);
[javac]         ^
```

## #56 - 12/04/2019 05:04 PM - Vladimir Tsichevski

After I have the offending RELEASE OBJECT commented, the demo compiles Ok, but running it causes no window displayed, and this endless output:

client:

```
[java] SLF4J: Class path contains multiple SLF4J bindings.
[java] SLF4J: Found binding in [jar:file:/home/vvt/p2j/branches/3856b/build/lib/slf4j-jdk14-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
[java] SLF4J: Found binding in [jar:file:/home/vvt/p2j/branches/3856b/build/lib/slf4j-simple-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
[java] SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
[java] SLF4J: Actual binding is of type [org.slf4j.impl.JDK14LoggerFactory]
[java] UI Theme successfully changed to 'material'
[java] DRAW: Ignoring event com.goldencode.p2j.ui.client.event.PaintEvent 9 raised while drawing widget com.goldencode.p2j.ui.client.gui.CaptionButton with id [-10].
[java] Dec 05, 2019 1:05:49 AM com.goldencode.p2j.ui.client.event.EventManager postEvent
[java] WARNING: Wasted event: com.goldencode.p2j.ui.client.event.PaintEvent@bcc2dcfa
[java] DRAW: Ignoring event com.goldencode.p2j.ui.client.event.PaintEvent 9 raised while drawing widget com.goldencode.p2j.ui.client.gui.CaptionButton with id [-10].
[java] Dec 05, 2019 1:05:50 AM com.goldencode.p2j.ui.client.event.EventManager postEvent
[java] WARNING: Wasted event: com.goldencode.p2j.ui.client.event.PaintEvent@c6d2d20
[java] DRAW: Ignoring event com.goldencode.p2j.ui.client.event.PaintEvent 9 raised while drawing widget c
```



```
om.goldencode.p2j.ui.client.gui.CaptionButton with id [-10].  
...
```

#### #57 - 12/05/2019 01:44 AM - Marius Gligor

Vladimir Tsichevski wrote:

Marius Gligor wrote:

Please use these tests. You should have both <test>.p and <test>.p.ext-hints files. You don't need to create a .wrx file.

Please check if you have the correct link to p2j project in testcases/uast  
In -s \$HOME/<your branch location>/3856b/ p2j  
Using a wrong link could cause a NPE exception.  
Also do a build of 3856b branch before starting the conversion.

The progressbar-demo.w.ext-hints is missing in the repository. I copied it from other hint file in the same directory first, but it was a mistake, since the name of the handle in progressbar-demo.w differs from that in all other tests.

I fixed the name in the hint file, and now the demo compiles Ok.

I think, you just forgot to add this hint file to the repository.

Indeed I forgot to push progressbar-demo.w.ext-hints I'll fix that

#### #58 - 12/05/2019 01:57 AM - Marius Gligor

Vladimir Tsichevski wrote:

For the progressbar-demo.w conversion produces uncompileable Java:

The line:  
[...]

converts to:

[...]

which causes the following compilation error:

[...]

progressbar-demo.w is an early test created "by hand" because we cannot use OpenEdge environment to create tests for Microsoft VB 6.0 common control due to missing the valid license.

Original statement: `RELEASE OBJECT chCtrlFrame.` is converted into Java as `ComServer.release(hProgressBar)`; but after conversion `chCtrlFrame` of type `COM-HANDLE` is converted into `hProgressBar` which is of type `HANDLE`

The solution is to completely remove the statement `RELEASE OBJECT chCtrlFrame.` at conversion time.

#### **#59 - 12/05/2019 02:05 AM - Marius Gligor**

Vladimir Tsichevski wrote:

After I have the offending `RELEASE OBJECT` commented, the demo compiles Ok, but running it causes no window displayed, and this endless output:

[...]

Not all tests are designed to show the main window. `progress_bar_test.p` as an early test was designed to test conversion only.

If you take a look inside ABL code we don't have any wait statement.

As a result when we run the test he is executed and then exit because we don't have a wait statement.

#### **#60 - 12/05/2019 03:53 AM - Hynek Cihlar**

Marius Gligor wrote:

The solution is to completely remove the statement `RELEASE OBJECT chCtrlFrame.` at conversion time.

I think the `RELEASE OBJECT` statement should be converted to `DELETE OBJECT`.

#### **#61 - 12/05/2019 04:02 AM - Marius Gligor**

Hynek Cihlar wrote:

Marius Gligor wrote:

The solution is to completely remove the statement `RELEASE OBJECT chCtrlFrame.` at conversion time.

I think the `RELEASE OBJECT` statement should be converted to `DELETE OBJECT`.

Yes, good point, `DELETE OBJECT` is the correct replacement.

**#62 - 12/05/2019 05:44 AM - Vladimir Tsichevski**

Marius Gligor wrote:

Not all tests are designed to show the main window. `progress_bar_test.p` as an early test was designed to test conversion only. If you take a look inside ABL code we don't have any wait statement. As a result when we run the test he is executed and then exit because we don't have a wait statement.

I am trying to run the `progressbar-demo.w` from the testcases. It ends with:

```
WAIT-FOR CLOSE OF DEFAULT-WINDOW.
```

And it is designed to run tests and display multiple messages.

**#63 - 12/05/2019 06:19 AM - Marius Gligor**

Vladimir Tsichevski wrote:

Marius Gligor wrote:

Not all tests are designed to show the main window. `progress_bar_test.p` as an early test was designed to test conversion only. If you take a look inside ABL code we don't have any wait statement. As a result when we run the test he is executed and then exit because we don't have a wait statement.

I am trying to run the progressbar-demo.w from the testcases. It ends with:

[...]

And it is designed to run tests and display multiple messages.

You can try to convert and run progress\_bar\_test3.p and progress\_bar\_test4.p which are more complex and allow you to interact with the progress bar widget using some buttons and you can test also mouse events.

#### #64 - 12/05/2019 06:37 AM - Marius Gligor

I added rules to convert RELEASE OBJECT into DELETE OBJECT and I converted my uast test

I have the following ABL statements in the test:

```
RELEASE OBJECT chCtrlFrame.  
DELETE OBJECT CtrlFrame.
```

which are converted in Java as follow:

```
HandleOps.delete(hProgressBar);  
HandleOps.delete(hProgressBar);
```

As you can see we have the same statement twice. I think that a better solution is to cut off RELEASE OBJECT chCtrlFrame. statement at conversion time. That's because I think the pair statements RELEASE and DELETE are always present in the code together.

#### #65 - 12/05/2019 07:35 AM - Vladimir Tsichevski

Marius Gligor wrote:

I added rules to convert RELEASE OBJECT into DELETE OBJECT and I converted my uast test

I do not see any updates in 3856b :(

I have the following ABL statements in the test:

[...]

which are converted in Java as follow:

[...]

As you can see we have the same statement twice. I think that a better solution is to cut off `RELEASE OBJECT chCtrlFrame.` statement at conversion time. That's because I think the pair statements `RELEASE` and `DELETE` are always present in the code together.

I am not sure, but I think, the `RELEASE` is used for COM handles, and `DELETE` for ordinary handles, so they will never be presented simultaneously? So only the `RELEASE` statement should be left in the OCX-based sources.

**#66 - 12/05/2019 07:38 AM - Vladimir Tsichevski**

Marius Gligor wrote:

You can try to convert and run `progress_bar_test3.p` and `progress_bar_test4.p` which are more complex and allow you to interact with the progress bar widget using some buttons and you can test also mouse events.

Then, I think, in order to prevent any confusion in the future, the `progressbar-demo.w` shall either be fixed, or removed from the repository completely.

**#67 - 12/05/2019 09:18 AM - Marius Gligor**

Vladimir Tsichevski wrote:

Marius Gligor wrote:

I added rules to convert `RELEASE OBJECT` into `DELETE OBJECT` and I converted my uast test

I do not see any updates in 3856b :(

I have the following ABL statements in the test:

[...]

which are converted in Java as follow:

[...]

As you can see we have the same statement twice. I think that a better solution is to cut off `RELEASE OBJECT chCtrlFrame.` statement at conversion time. That's because I think the pair statements `RELEASE` and `DELETE` are always present in the code together.

I am not sure, but I think, the `RELEASE` is used for COM handles, and `DELETE` for ordinary handles, so they will never be presented simultaneously? So only the `RELEASE` statement should be left in the OCX-based sources.

If we keep only `RELEASE` we have to transform into `DELETE`, but another possible solution is to keep both if not raise runtime problems. I have to check what's happen at runtime when we execute `HandleOps.delete(hProgressBar);` twice.

**#68 - 12/05/2019 09:29 AM - Hynek Cihlar**

Marius Gligor wrote:

If we keep only RELEASE we have to transform into DELETE, but another possible solution is to keep both if not raise runtime problems. I have to check what's happen at runtime when we execute `HandleOps.delete(hProgressBar)`; twice.

Also consider more complicated cases. Like passing a comhandle to an external procedure and releasing it there.

**#69 - 12/05/2019 10:10 AM - Marius Gligor**

Hynek Cihlar wrote:

Marius Gligor wrote:

If we keep only RELEASE we have to transform into DELETE, but another possible solution is to keep both if not raise runtime problems. I have to check what's happen at runtime when we execute `HandleOps.delete(hProgressBar)`; twice.

Also consider more complicated cases. Like passing a comhandle to an external procedure and releasing it there.

In that case I'm affraid we have to do changes in the external procedure because we transform com-handle into handle in the current procedure. Do you see another option?

But when the com-handle is released in the current procedure file we could do as follow. Before transforming RELEASE into DELETE I always take care to check if the com-handle is defined in `.ext-hints` file: `control-frame-handle="chCtrlFrame"` and do the conversion only if com-handle name match.

Even so, unfortunately is not 100% safe. If we have something like:

```
define variable chCtrlFrame as component-handle no-undo.  
...  
procedure close_handle :  
  define input parameter chCtrlFrame as com-handle.  
  ...  
  release object chCtrlFrame.  
end procedure.
```

it is a problem. It's less probable to have this pattern but is not zero.

Marius Gligor wrote:

Hynek Cihlar wrote:

Marius Gligor wrote:

If we keep only RELEASE we have to transform into DELETE, but another possible solution is to keep both if not raise runtime problems.

I have to check what's happen at runtime when we execute `HandleOps.delete(hProgressBar)`; twice.

Also consider more complicated cases. Like passing a comhandle to an external procedure and releasing it there.

In that case I'm afraid we have to do changes in the external procedure because we transform com-handle into handle in the current procedure. Do you see another option?

There was such a case in one of the customer apps. I added an ext-hints file for the external procedure and removed some assumptions from the rules files to properly convert the comhandle to a handle in the other external procedure. You should test this case as well. Nothing complicated, passing a comhandle and referencing a prop and a method should do it.

In general converting comhandles in other procedure files shouldn't be any different. Only the structure is different - no LoadControls, no control frame handles, etc.

But when the com-handle is released in the current procedure file we could do as follow.

Before transforming RELEASE into DELETE I always take care to check if the com-handle is defined in .ext-hints file: `control-frame-handle="chCtrlFrame"` and do the conversion only if com-handle name match.

This makes sense.

Even so, unfortunately is not 100% safe. If we have something like:

[...]

it is a problem. It's less probable to have this pattern but is not zero.

IMHO this could be covered with the same rules. There just could be additional variables.

#71 - 12/05/2019 10:49 AM - Hynek Cihlar

Code review 3856b. I only have a few minor points.

ocx\_conversion.rules

- The aliases for TREELIST must also include TREEVIEW aliases.
- Regarding dropping the init procedure. It would be safer to drop the procedure body, than the whole assignment where the procedure is referenced. I've encountered a case where such a drop caused removal of other useful logic.

checkParams in ComObject is never called.

I encountered several places where file history numbering was not correct, for example multiple numbers where there should be only one.

Widget.refreshWidget

The only implementation only forwards the call to repaint(). Why not to call repaint() directly?

Btw, to repaint a widget do:

```
eventDrawingBracket(widget, widget::repaint)
```

Similarly as above, for oleDrag use eventDrawingBracket. Btw. if you need the repaints to happen out of the os event loop, you may need to also call processRepaints.

A small formatting issue - conditionals with one line bodies are sometime missing curly braces.

getMoselconLocation => getMouseIconLocation, also the method should be part of Theme interface and the image resources part of the theme packages.

GWD.mimeType

The method should be moved among other private methods. The input argument should be trimmed and lower-cased.

drawProgressBar and drawProgressBarBar

Params are incorrectly formatted.

ImageListWidget.listImages is missing comment.

ILW.resourceDelete(), global images are already removed on the client as part of widget destroy in AbstractWidget.destroy().

Are OLE-X and OLE-Y in pixels? If so, they should be ints.

OcxMouseHelper class is missing javadoc.

OleDragDrop class is missing javadoc.

OleDragDropHelper.dirty() a good candidate for a common method.

ProgressBarGuiImpl.height() is missing javadoc.



**#72 - 12/05/2019 05:16 PM - Vladimir Tsichevski**

Marius Gligor wrote:

You can try to convert and run `progress_bar_test3.p` and `progress_bar_test4.p` which are more complex and allow you to interact with the progress bar widget using some buttons and you can test also mouse events.

I tried all the progressbar tests in the testcases (I removed all the offending `RELEASE` lines):

1. `progressbar-demo.w` does not work
2. `progress_bar_test.p` presumably works, but shows nothing, I think this test shall be removed.
3. `progress_bar_test1.p` presumably works, but shows nothing, I think this test shall be removed.
4. `progress_bar_test2.p` demonstrates the horizontal bar capabilities, works perfectly.
5. `progress_bar_test3.p` demonstrates the vertical bar capabilities, works perfectly.
6. `progress_bar_test4.p` demonstrates the horizontal bar capabilities, works perfectly. I suppose, it is an advanced version of test2, if so the test2 shall be probably removed.
7. `progress_bar_test5.p` does not work
8. `progress_bar_test6.p` demonstrates the horizontal bar capabilities again. I do not see any difference from the test2 and test4 :(
9. `progress_bar_test7.p` need some arguments to run, but no any description provided on how to run it.

I think, the following shall be done with the tests:

1. remove the test which do not work or make them working
2. remove the duplicate test, if one test is the advanced version of another, remove the less advanced one
3. provide the test names in window titles (now all test window titles are the same or have no meaning)
4. provide a short `README` describing what is demonstrated in each test.
5. optionally provide a `run-all.p` which runs all tests in sequence.

**#73 - 12/06/2019 02:07 AM - Marius Gligor**

Vladimir Tsichevski wrote:

Marius Gligor wrote:

You can try to convert and run `progress_bar_test3.p` and `progress_bar_test4.p` which are more complex and allow you to interact with the progress bar widget using some buttons and you can test also mouse events.

I tried all the progressbar tests in the testcases (I removed all the offending `RELEASE` lines):

1. `progressbar-demo.w` does not work
2. `progress_bar_test.p` presumably works, but shows nothing, I think this test shall be removed.
3. `progress_bar_test1.p` presumably works, but shows nothing, I think this test shall be removed.

4. progress\_bar\_test2.p demonstrates the horizontal bar capabilities, works perfectly.
5. progress\_bar\_test3.p demonstrates the vertical bar capabilities, works perfectly.
6. progress\_bar\_test4.p demonstrates the horizontal bar capabilities, works perfectly. I suppose, it is an advanced version of test2, if so the test2 shall be probably removed.
7. progress\_bar\_test5.p does not work
8. progress\_bar\_test6.p demonstrates the horizontal bar capabilities again. I do not see any difference from the test2 and test4 :(
9. progress\_bar\_test7.p need some arguments to run, but no any description provided on how to run it.

I think, the following shall be done with the tests:

1. remove the test which do not work or make them working
2. remove the duplicate test, if one test is the advanced version of another, remove the less advanced one
3. provide the test names in window titles (now all test window titles are the same or have no meaning)
4. provide a short README describing what is demonstrated in each test.
5. optionally provide a run-all.p which runs all tests in sequence.

If you take a look over the ABL code inside progress\_bar\_test6.p you will see PROGRESS-BAR widget as a 4gl extension. Once implemented PROGRESS-BAR can be used as any 4gl widget.

**#74 - 12/06/2019 10:21 AM - Vladimir Tsichevski**

- File stacktrace.log added

Got NPE on the following operation:

```
DEFINE VARIABLE pbar AS COMPONENT-HANDLE NO-UNDO.  
ASSIGN pbar = chCtrlFrame:ProgressBar.
```

In Java:

```
if (res != null && res.id() == 0)
```

the origin: re.id() is Long, and its value is null.

#75 - 12/06/2019 12:01 PM - Vladimir Tsichevski

Some code review notes:

## Conversion

The body of the control\_load procedure is completely erased by conversion. Meanwhile, it may originally contain code manually inserted by the programmer, for example, in a large customer application I have seen such code.

All methods are marked as LegacyMethod in the ProgressBar interface. No @ComMethod/@ComProperty is used.

Hence there is no reliable way to tell COM properties from COM methods. It may make things complicated when implementing the [#4438](#) task.

## Error checking

No error checking is implemented at all for property setters. I can assign any values to any property given that they are either of the compatible type or can silently be converted into a compatible type.

## Compiler warnings

The @Override annotations are missing in most places where they are appropriate.

## Code style

4gl example code does not follow the 3-space tab indent convention.

## Javadocs

1. Mixed style of Javadoc comments: in some cases just the {@inheritDoc} placeholder is used (which is the right thing IMHO), in some other the Javadoc comments copy-pasted from the overridden methods (which is not the right thing).

2. The constructor ProgressBarWidget(boolean dynamic) is described as "Default constructor.", which it is not.  
no class Javadoc

3. progBarRefresh(), setProgBarMousePointer(), getMouseIcon(): It is unclear from the Javadoc what do these methods do.

#76 - 12/06/2019 01:01 PM - Greg Shah

remove the test which do not work or make them working

remove the duplicate test, if one test is the advanced version of another, remove the less advanced one

Vladimir, it is OK to have either of these cases in our testcases/uast/ project if the extra tests have some purpose. In other words, we do not require that all testcases in testcases/uast/ are functional or well factored. In fact, we have many tests that are only useful for looking at parsing or full conversion, but which have no correct runtime behavior.

I agree that it is a good practice to leave behind some comments in that case so that future readers are clear as to the purpose.

Mixed style of Javadoc comments: in some cases just the {@inheritDoc} placeholder is used (which is the right thing IMHO), in some other the Javadoc comments copy-pasted from the overridden methods (which is not the right thing).

You may be right on this, but our current coding standards discourages use of {@inheritDoc}:

Do NOT use @inheritDoc. Having nothing but "@inheritDoc" at the source code level for a method's documentation makes it necessary for a developer to either:

- keep a browser with previously compiled javadoc open alongside the editor; or
- know which superclass to navigate to in order to read the javadoc for a given method.

I guess in an IDE this is not an issue but when processing the source code without specialized tools it becomes an issue. A few of us Luddites don't use IDEs for development. :)

#### #77 - 12/10/2019 07:54 AM - Marius Gligor

Hynek Cihlar wrote:

- Regarding dropping the init procedure. It would be safer to drop the procedure body, than the whole assignment where the procedure is referenced. I've encountered a case where such a drop caused removal of other useful logic.

Did you fix that?

I'm a little bit confused about drop-procedure option.

1. We have an option to specify drop procedure name in .ext-hint file <drop-procedure name="control\_load" /> Is this optional or is mandatory? IMO should be mandatory according with my below description.

2. With this option in place the procedure:

```
PROCEDURE control_load :
```

```

DEFINE VARIABLE UIB_S      AS LOGICAL      NO-UNDO.
DEFINE VARIABLE OCXFile    AS CHARACTER    NO-UNDO.

OCXFile = SEARCH( "pbar.wrx" ).

IF OCXFile = ? THEN
  OCXFile = SEARCH(SUBSTRING(THIS-PROCEDURE:FILE-NAME, 1,
    R-INDEX(THIS-PROCEDURE:FILE-NAME, " ":U), "CHARACTER":U) + "wrx":U).

IF OCXFile <> ? THEN
DO:
  ASSIGN
    chCtrlFrame = CtrlFrame:COM-HANDLE.
    UIB_S = chCtrlFrame:LoadControls(OCXFile, "ProgressBar":U).
END.
ELSE MESSAGE "pbar.wrx" SKIP(1)
  "The binary control file could not be found. The controls cannot be loaded."
  VIEW-AS ALERT-BOX TITLE "Controls Not Loaded".
END PROCEDURE.

```

is converted into Java code as follow:

```

@LegacySignature(type = Type.PROCEDURE, name = "control_load")
public void controlLoad()
{
  internalProcedure(new Block());
}

```

which means the procedure is not drooped, instead the original body is replaced by an empty body. Any potential call of this procedure RUN control\_load. is converted into ControlFlowOps.invoke("control\_load");  
Is this behavior OK or we should take care about other things?

BTW I created a small application using Application Builder having a PSTimer OCX keeping only generated code. control\_load is called automatically when the application starts, no explicit call is needed! Looks like in converted FWD will be never called!

**#78 - 12/10/2019 08:47 AM - Hynek Cihlar**

Marius Gligor wrote:

which means the procedure is not drooped, instead the original body is replaced by an empty body. Any potential call of this procedure RUN control\_load. is converted into ControlFlowOps.invoke("control\_load");  
Is this behavior OK or we should take care about other things?

control\_load must not be dropped for multiple reasons. (1) The call to LoadControls is needed for COM objects converted to ComObjects (not to widgets). (2) control\_load may call the procedure initialize-controls. (3) There are cases where properties are assigned in control\_load.

BTW I created a small application using Application Builder having a PSTimer OCX keeping only generated code. control\_load is called automatically when the application starts, no explicit call is needed! Looks like in converted FWD will be never called!

This is strange, I see run control\_load statements in the legacy code. Also I've debugged the execution of the converted control\_load procedure. Can you post the converted code?

**#79 - 12/10/2019 09:00 AM - Marius Gligor**

- File pb-timer.w added

Hynek Cihlar wrote:

Marius Gligor wrote:

which means the procedure is not drooped, instead the original body is replaced by an empty body. Any potential call of this procedure RUN control\_load. is converted into ControlFlowOps.invoke("control\_load");  
Is this behavior OK or we should take care about other things?

control\_load must not be dropped for multiple reasons. (1) The call to LoadControls is needed for COM objects converted to ComObjects (not to widgets). (2) control\_load may call the procedure initialize-controls. (3) There are cases where properties are assigned in control\_load.

BTW I created a small application using Application Builder having a PSTimer OCX keeping only generated code. control\_load is called automatically when the application starts, no explicit call is needed! Looks like in converted FWD will be never called!

This is strange, I see run control\_load statements in the legacy code. Also I've debugged the execution of the converted control\_load procedure. Can you post the converted code?

OK So the drop-procedure option MUST not be used with control\_load procedure. In Open Edge control\_load is automatically generated by Application Builder and normally contains the OCX initialization code. As I understood it is possible for a developer to add extra code to this procedure

"by hand". If this is the case, make sense to not cut off potentially extra code added by developer. I attached my sample.

## #80 - 12/11/2019 10:04 AM - Marius Gligor

Hynek Cihlar wrote:

Code review 3856b. I only have a few minor points.

ocx\_conversion.rules

- The aliases for TREELIST must also include TREEVIEW aliases.
- Regarding dropping the init procedure. It would be safer to drop the procedure body, than the whole assignment where the procedure is referenced. I've encountered a case where such a drop caused removal of other useful logic.

checkParams in ComObject is never called.

I encountered several places where file history numbering was not correct, for example multiple numbers where there should be only one.

Widget.refreshWidget

The only implementation only forwards the call to repaint(). Why not to call repaint() directly?

Btw, to repaint a widget do:

```
eventDrawingBracket(widget, widget::repaint)
```

Similarly as above, for oleDrag use eventDrawingBracket. Btw. if you need the repaints to happen out of the os event loop, you may need to also call processRepaints.

A small formatting issue - conditionals with one line bodies are sometime missing curly braces.

getMoselconLocation => getMouseIconLocation, also the method should be part of Theme interface and the image resources part of the theme packages.

GWD.mimeType

The method should be moved among other private methods. The input argument should be trimmed and lower-cased.

drawProgressBar and drawProgressBarBar

Params are incorrectly formatted.

ImageListWidget.listImages is missing comment.

ILW.resourceDelete(), global images are already removed on the client as part of widget destroy in AbstractWidget.destroy().

Are OLE-X and OLE-Y in pixels? If so, they should be ints.

OcxMouseHelper class is missing javadoc.

OleDragDrop class is missing javadoc.

OleDragDropHelper.dirty() a good candidate for a common method.

ProgressBarGuiImpl.height() is missing javadoc.

I've fixed your code review reported issues. OLE-X and OLE-Y are DECIMAL not INTEGER values.

My changes have been committed on branch 3856b rev 11344 to 11352. Please do a code review on my changes.

## #81 - 12/11/2019 02:06 PM - Hynek Cihlar

Marius Gligor wrote:

OLE-X and OLE-Y are DECIMAL not INTEGER values.

Please update the javadocs to include the information about the units of the OLE-X and OLE-Y attributes. I'm worried that the decimal type may be confused with character units.

My changes have been committed on branch 3856b rev 11344 to 11352. Please do a code review on my changes.

ocx\_conversion.rules:

The copied aliases from TREEVIEW to TREELIST is not optimal. I'm afraid these could get out of sync in the future. Instead of copying the entries, just do `<action>map3.putAll(map2)</action>`.

Broken formatting for the rule `<rule>type prog.expression and parent.type prog.release_object...`

ComObject.java:

The history entry numbering is 006 008.

Theme.java:

Sorry, I should have mentioned this earlier. We already have a method for resolving theme resources. So instead of `getUserDefinedMouseIconLocation`, please use `getBinaryResource`.

ClassicTheme.java:

The order of params in the javadoc for method `drawProgressBar` is incorrect.

## #82 - 12/11/2019 02:43 PM - Vladimir Tsichevski

- File diff added

I've made a code review, and I have compared the FWD Progress bar behavior with that of the original Microsoft OCX.

### Code

1. `Theme#drawProgressBarBar(...)` is private by its nature and need not to be exported
2. `ProgressBarWidget#comData` field has getter and setter, so it has to be declared private
3. Some errors in Javadoc for `Theme#drawProgressBarBar(...)` and `Theme#drawProgressBarBar(...)` see the diff attached.
4. Got NPE in `ImageGuiImpl#draw()`, and has to add a check for null (this note is not about the `ProgressBar` though).



## Conversion

Several screens of messages which, in my opinion, should not appear in the final version:

```
[java] ./abl/progress_bar_errors.p
[java] INFO: FWD OCX Extension: replaced DEFINE VARIABLE [chCtrlFrame] with [hProgressBar]
[java] INFO: FWD OCX Extension: replaced CREATE ActiveX [CTRLFRAME] with [hProgressBar]
[java] INFO: FWD OCX Extension: convert CHCTRLFRAME:PROGRESSBAR to hProgressBar.
... the last message repeats ~15 times
```

```
[java] WARNING: Variable y (refid 12884903473) missing in dictionary [VAR_INT]!
[java] WARNING: Variable x (refid 12884903456) missing in dictionary [VAR_INT]!
[java] WARNING: Variable shift (refid 12884903439) missing in dictionary [VAR_INT]!
[java] WARNING: Variable button (refid 12884903422) missing in dictionary [VAR_INT]!
... the last 4 message repeat several times
```

## Tests and Demos

The "Start" button in examples runs a cycle with incremental progressbar value increase, but it runs too quick for any human to notice the steps. Some small delays between the steps shall probably be added programmatically in demo sources.

### Runtime

1. The cursor in the progressbar area is **always** a clock cursor.

The behavior **differs** significantly from that of the original OCX control:

#### In FWD

1. any value may be assigned to any property. No error other than "incompatible argument type" is ever reported.
2. an attempt to set the progressbar value outside of the current min/max range is silently ignored, and the value stays unchanged
3. If the current progressbar value is outside of the current min/max range, an empty bar is displayed.

#### In the original OCX control

1. only the valid values may be assigned to properties. Otherwise an **error** is raised.
2. an attempt to set the progressbar value outside of the current min/max raises an **error**.
3. If the current progressbar value is outside of the current min/max range, either an empty bar or a full bar is displayed, depending on whether the value is lesser than the minimum or greater than the maximum.
4. It is an **error** to set any of min/max/value properties to a negative value.
5. It is **not** an error to set the the max < value. The full bar is displayed in this case.
6. It is **not** an error to set the the min > value. An empty bar is displayed in this case.

Vladimir Tsichevski wrote:

1. Got NPE in ImageGuiImpl#draw(), and has to add a check for null (this note is not about the ProgressBar though).

How did you get that? ImageGuiImpl is not related to ProgressBar nor ImageList implementation!

1. The cursor in the progressbar area is always a clock cursor.
2. The behavior differs significantly from that of the original OCX control:

If you take a look over ABL code you will see this statement `chCtrlFrame.ProgressBar.MousePointer = 3`.

The ProgressBar mouse cursor is intentionally set to 3 (`ccIBeam=3`) So, when the mouse cursor is over the ProgressBar widget we should see the IBeam cursor otherwise we should see the default cursor. In `progress_par_test3.p` for example the mouse cursor is always the default cursor `ccDefault=0`.

1. The behavior differs significantly from that of the original OCX control:

How did you manage to use original ProgressBar OCX? In both environment local VM and AWS we cannot create tests using ProgressBar OCX due to the missing of a valid license!!

All my ProgressBar tests are created using a skeleton from Progress PSTimer OCX. After that I edited the generated ABL code and I added ABL code specific to ProgressBar.

Whenever we start to implement a new widget we always start with conversion. This is done on the server side by creating the server side widget interface, widget configuration and defining conversion rules. At this stage we create very simple uast tests just to check if conversion works properly, we don't care about functionality and UI. Even though these tests looks useless in fact are very useful to test the conversion part.

Once the conversion part is implemented we move forward to implement client side widget. Here in general we are concentrate to implement the widget visual aspect (draw) and handle events for mouse, keyboard, etc. At this stage we create more complex uast tests because we want to test the widget visual aspect and behavior.

Finally, if the widget is an OCX replacement, we write a simple uast test to check if the widget can be used as a 4gl extension. We keep all our tests in testcases project which is reach collection of uast tests.

At this moment p2j is a conversion tool and we trust in our client ABL code. Once the code is running on OpenEdge client environment we consider that is a valid code. Since we assume that we convert only valid ABL code we don't do syntax checks. It is possible to write and convert invalid ABL code but the runtime behavior is unpredicted.

**#84 - 12/12/2019 08:53 AM - Marius Gligor**

Hynek Cihlar wrote:

Marius Gligor wrote:

OLE-X and OLE-Y are DECIMAL not INTEGER values.

Please update the javadocs to include the information about the units of the OLE-X and OLE-Y attributes. I'm worried that the decimal type may be confused with character units.

My changes have been committed on branch 3856b rev 11344 to 11352. Please do a code review on my changes.

ocx\_conversion.rules:

The copied aliases from TREEVIEW to TREELIST is not optimal. I'm afraid these could get out of sync in the future. Instead of copying the entries, just do `<action>map3.putAll(map2)</action>`.

Broken formatting for the rule `<rule>type prog.expression and parent.type prog.release_object...`

ComObject.java:

The history entry numbering is 006 008.

Theme.java:

Sorry, I should have mentioned this earlier. We already have a method for resolving theme resources. So instead of `getUserDefinedMouseIconLocation`, please use `getBinaryResource`.

ClassicTheme.java:

The order of params in the javadoc for method `drawProgressBar` is incorrect.

I've fixed reported issues. Also I've fixed some of the Vladimir reported issues.  
Please do a review of changes in branch 3856b rev 11353 and 11354

#85 - 12/12/2019 09:05 AM - Vladimir Tsichevski

- File progressbar-in-appbuilder.png added

Marius Gligor wrote:

Vladimir Tsichevski wrote:

1. Got NPE in ImageGuilImpl#draw(), and has to add a check for null (this note is not about the ProgressBar though).

How did you get that? ImageGuilImpl is not related to ProgressBar nor ImageList implementation!

That is why I wrote "this note is not about the ProgressBar though". It is about the ImageList solely. Sorry for the confusion.

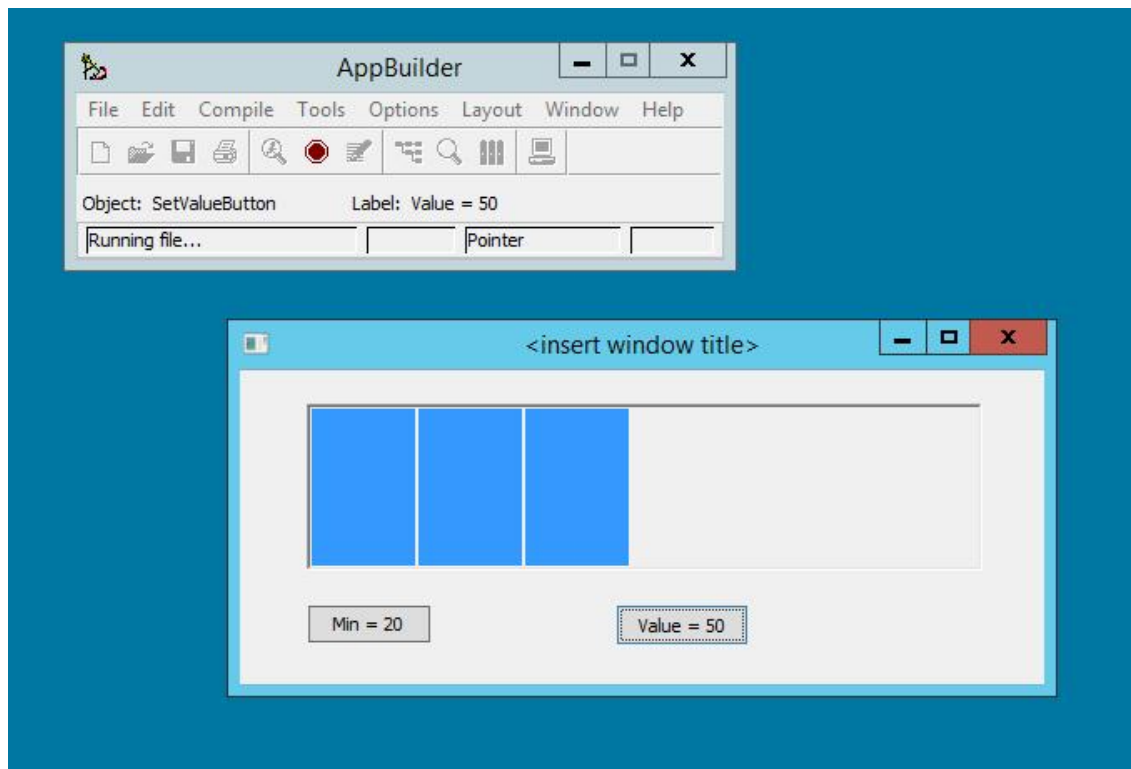
1. The cursor in the progressbar area is always a clock cursor.

If you take a look over ABL code you will see this statement `chCtrlFrame:ProgressBar:MousePointer = 3`. The ProgressBar mouse cursor is intentionally set to 3 (`cclBeam=3`)...

You are right, I did not notice this line in the code.

How did you manage to use original ProgressBar OCX? In both environment local VM and AWS we cannot create tests using ProgressBar OCX due to the missing of a valid license!!

You can install a VisualStudio 2017 community edition 30-day trial.



It is possible to write and convert invalid ABL code but the runtime behavior is unpredicted.

1. Some cases I described are valid usage examples, for which the FWD progress bar behaves differently. For example, if value > max, the OCX bar displays full, and FWD bar displays empty.
2. it is always better to create predictable code than unpredictable :)

**#86 - 12/12/2019 10:12 AM - Marius Gligor**

- File *oe-32.png* added

Vladimir Tsichevski wrote:

1. You can install a VisualStudio 2017 community edition 30-day trial.

I installed VisualStudio 2017 community edition 30-day trial on my VM and finally I succeeded to use MS VB 6.0 common controls. It's works only with the 32 bit version so I created a shortcut to the 32 bit procedure editor and application builder. That's really very useful because now I'm able to do my own tests. After 30 days we could reinstall the VM and VisualStudio 2017 again. I hope will works.

Thanks Vladimir!

**#87 - 12/12/2019 10:25 AM - Vladimir Tsichevski**

Marius Gligor wrote:

Vladimir Tsichevski wrote:

1. You can install a VisualStudio 2017 community edition 30-day trial.

After 30 days we could reinstall the VM and VisualStudio 2017 again. I hope will works.

After 30 days VS will ask you to login to the Microsoft site and confirm you want continue the trial. You will be given another 30 days of trial. No need to reinstall anything!

**#88 - 12/13/2019 05:46 AM - Marius Gligor**

Marius Gligor wrote:

Hynek Cihlar wrote:

Marius Gligor wrote:

OLE-X and OLE-Y are DECIMAL not INTEGER values.

Please update the javadocs to include the information about the units of the OLE-X and OLE-Y attributes. I'm worried that the decimal type may be confused with character units.

My changes have been committed on branch 3856b rev 11344 to 11352. Please do a code review on my changes.

ocx\_conversion.rules:

The copied aliases from TREEVIEW to TREELIST is not optimal. I'm afraid these could get out of sync in the future. Instead of copying the entries, just do `<action>map3.putAll(map2)</action>`.

Broken formatting for the rule `<rule>type prog.expression and parent.type prog.release_object...`

ComObject.java:

The history entry numbering is 006 008.

Theme.java:

Sorry, I should have mentioned this earlier. We already have a method for resolving theme resources. So instead of `getUserDefinedMouseIconLocation`, please use `getBinaryResource`.

ClassicTheme.java:

The order of params in the javadoc for method `drawProgressBar` is incorrect.

I've fixed reported issues. Also I've fixed some of the Vladimir reported issues.

Please do a review of changes in branch 3856b rev 11353 and 11354

I added validations for all ProgressBar widget properties values which support only a set of predefined values.

Changes were committed in branch 3856b rev 11355 and 11356

**#89 - 12/17/2019 11:16 AM - Hynek Cihlar**

Code review 3856b revisions 11353, 11354, 11355 and 11356. The changes look good.

**#91 - 01/29/2020 05:30 PM - Greg Shah**

- Status changed from WIP to Closed

- % Done changed from 80 to 100

Task branch 3856c was merged to trunk as revision 11341.

**#92 - 01/29/2020 05:33 PM - Greg Shah**

Branch 3858b was dead archived.

**#93 - 02/04/2020 08:57 AM - Hynek Cihlar**

I'm going through the progress bar implementation merged to trunk and I think we should remove any OCX references in the classes, methods, field identifiers and legacy API. Like `OcxMouseConfig`, `OCX-MOUSE-BUTTON`, `OCX-MOUSE-SHIFT`, `OCX-MOUSE-X`, `OCX-MOUSE-Y`, `OcxMouse`, `OcxDragDrop`, `OcxDragDropConfig`, `OcxMouseHelper`.

The OCX string in the identifiers makes the APIs misleading as there is no connection to the OCX technology. AFAIK `ProgressBar` widget should be a normal legacy widget.

**#94 - 02/04/2020 09:09 AM - Greg Shah**

AFAIK `ProgressBar` widget should be a normal legacy widget.

Yes, I agree that we want the legacy widget case to be well supported. It does need to also replace the OCX use case.

If there is no conflict in the keywords, then it is OK to remove the OCX. If there is a conflict, then we can use a different keyword and still drop the OCX prefix.

**#95 - 02/04/2020 09:32 AM - Vladimir Tsichevski**

Hynek Cihlar wrote:

I'm going through the progress bar implementation merged to trunk and I think we should remove any OCX references in the classes, methods, field identifiers and legacy API. Like `OcxMouseConfig`, `OCX-MOUSE-BUTTON`, `OCX-MOUSE-SHIFT`, `OCX-MOUSE-X`, `OCX-MOUSE-Y`, `OcxMouse`, `OcxDragDrop`, `OcxDragDropConfig`, `OcxMouseHelper`.

The OCX string in the identifiers makes the APIs misleading as there is no connection to the OCX technology. AFAIK `ProgressBar` widget should be a normal legacy widget.

I support this point of view. Moreover, I think, the same is applicable to any widget we create as a replacement for OCX controls.

Our technology must correctly convert 4gl COM syntax into Java code, which does things, what the 4gl application developers initially intended to do. Period. Beyond that, everything is implementation details.

Hence, we may follow the COM/OCX technology and terminology only as long as it suits us.

Example: in contrast with OpenEdge documentation, from 4gl code only, you cannot tell the "COM method" from "COM property get/set"call. So, in our replacement code, there is no point other than historical, make any difference in support for COM methods and COM properties.

## Files

---

smart-window.png	11 KB	09/16/2019	Marius Gligor
not-licensed.png	72.4 KB	09/17/2019	Marius Gligor
progress-bar.xlsx	11.4 KB	09/17/2019	Marius Gligor
progress_bar_test.p.ast	61.8 KB	09/20/2019	Marius Gligor
progress_bar_test.p	1.3 KB	09/20/2019	Marius Gligor
progress_bar_test.p.ext-hints	275 Bytes	09/20/2019	Marius Gligor
progress-bar-classic-ui-theme.zip	19 KB	10/11/2019	Marius Gligor
progress-bar-ui-themes.zip	12.8 KB	10/14/2019	Marius Gligor
progressbar.p.ast.zip	75.4 KB	10/25/2019	Marius Gligor
progress_bar_test6.p	5.08 KB	10/29/2019	Marius Gligor
error.log	18.9 KB	12/04/2019	Vladimir Tsichevski
stacktrace.log	3.02 KB	12/06/2019	Vladimir Tsichevski
pb-timer.w	10.3 KB	12/10/2019	Marius Gligor
diff	15.5 KB	12/11/2019	Vladimir Tsichevski
progressbar-in-appbuilder.png	12.8 KB	12/12/2019	Vladimir Tsichevski
oe-32.png	64.2 KB	12/12/2019	Marius Gligor