

User Interface - Bug #2495

APPLY event TO handle does not work

01/22/2015 08:56 AM - Greg Shah

Status:	Closed	Start date:	01/21/2015
Priority:	Normal	Due date:	
Assignee:	Constantin Asofiei	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:	GUI Support for a Complex ADM2 App	case_num:	
billable:	No		
vendor_id:	GCD		
Description			

History

#1 - 01/22/2015 08:59 AM - Greg Shah

I have found a common idiom which we don't handle properly. Here is an example:

```
ON F4, END-ERROR OF FRAME some-frame DO:
  HIDE MESSAGE.
  APPLY "CLOSE" TO THIS-PROCEDURE. <--- this is the problem
  RETURN NO-APPLY.
END.

ON WINDOW-CLOSE OF CURRENT-WINDOW DO:
  APPLY "CLOSE":U TO THIS-PROCEDURE. <--- this is the problem
  RETURN NO-APPLY.
END.

ON F12 ANYWHERE DO:
  APPLY "CLOSE":U TO THIS-PROCEDURE. <--- this is the problem
  RETURN NO-APPLY.
END.
```

This successfully converts and compiles to:

```
public class TriggerBlock15
  extends Trigger
  {
    public void body()
    {
      hideMessage(false);

      thisProcedure().unwrapWidget().apply("CLOSE");
      returnConsume();
    }
  }
```

I guess this could be used as a trigger event, but in this case it is used as a GO-ON clause for a WAIT-FOR:

```
EventList list18 = new EventList();
list18.addEvent("CLOSE", thisProcedure());
LogicalTerminal.waitFor(null, list18, (GenericWidget) hFocus.unwrapWidget(), tmOut);
```

Of course, at runtime the unwrapWidget() will return a proxy which yields:

**apply is not a queryable attribute for PROCEDURE widget. (4052)

The related code is in `handle.InvalidAttributeAccess.invoke()` but it is expected in this case.

The AST looks like this:

```
<ast col="0" id="605590402631" line="0" text="statement" type="STATEMENT">
  <ast col="3" id="605590402632" line="2190" text="APPLY" type="KW_APPLY">
    <annotation datatype="java.lang.Long" key="peerid" value="11617886542521"/>
    <ast col="17" id="605590402639" line="2190" text="TO" type="KW_TO">
      <annotation datatype="java.lang.Long" key="peerid" value="11617886542522"/>
      <ast col="20" id="605590402641" line="2190" text="THIS-PROCEDURE" type="SYS_HANDLE">
        <annotation datatype="java.lang.Long" key="oldtype" value="827"/>
        <annotation datatype="java.lang.Long" key="tempidx" value="261"/>
        <annotation datatype="java.lang.Long" key="peerid" value="11617886542523"/>
      </ast>
    </ast>
  </ast>
  <ast col="0" id="605590402635" line="0" text="expression" type="EXPRESSION">
    <annotation datatype="java.lang.Long" key="peerid" value="11617886542524"/>
    <ast col="9" id="605590402636" line="2190" text="&quot;CLOSE&quot;" type="STRING">
      <annotation datatype="java.lang.Boolean" key="is-literal" value="true"/>
      <annotation datatype="java.lang.Long" key="peerid" value="11617886542525"/>
    </ast>
  </ast>
</ast>
</ast>
</ast>
```

The conversion rules `ui_statements.rules`:

```
<!-- emit apply -->
<rule>type == prog.kw_apply
```

```
<!-- this works for frame and widget forms (not for the global
form), if the widget form is used, the widget will emit
automatically downstream) -->
<action>methodText = "apply"</action>
```

```
<rule>
  numImmediateChildren == 1      or
  downPath("KW_TO/KW_FRAME/WID_FRAME") or
  downPath("KW_TO/WID_FRAME")
```

```
<rule>isNote("frame-id")
```

```
<!-- frame form (the "TO FRAME frame" has been hidden by
now), force an artificial frame reference to be
emitted below -->
```

```
<action>
  refName = #(java.lang.String)
            execLib("get_framename", this)
</action>
```

```
<!-- global form with no widget phrase -->
<action on="false">methodText = "apply"</action>
<action on="false">importLT = true</action>
<action on="false">methodType = java.static_method_call</action>
```

```
</rule>
</rule>
</rule>
```

...

```
<!-- handle references need a cast -->
<rule>type == prog.kw_to and parent.type == prog.kw_apply
```

```
<action>ref = copy.getChildAt(0)</action>
```

```
<rule>
```

```
ref.type == prog.var_handle or
ref.type == prog.field_handle or
ref.type == prog.func_handle or
ref.type == prog.meth_handle or
ref.type == prog.attr_handle or
evalLib("converted_to_handle", ref)
```

```
<action>
  createPeerAst(java.method_call, "unwrapWidget", closestPeerId)
</action>
</rule>
</rule>
```

Not only is the conversion wrong, but we don't support this idiom at all at runtime. Even if we called the LT version, here are the `LogicalTerminal.apply()` signatures:

```
public static void apply(GenericFrame frame, String event)
public static void apply(GenericFrame frame, int eventCode)
public static void apply(GenericFrame frame, int widgetId, String event)
public static void apply(GenericFrame frame, int widgetId, int eventCode)
public static void apply(String event)
public static void apply(int64 eventCode)
public static void apply(double eventCode)
public static void apply(long eventCode)
```

Our widget-focused event model is coming back to haunt us here. I'm open to any ideas on the best way forward.

#2 - 01/22/2015 09:00 AM - Greg Shah

- File `ca_upd20150122a.zip` added

----- Forwarded Message -----

Subject: Re: common idiom which we don't handle properly
Date: Thu, 22 Jan 2015 13:03:57 +0200
From: Constantin Asofiei <ca@goldencode.com>
Reply-To: ca@goldencode.com
To: ges@goldencode.com

CC: Faulhaber, Eric <ecf@goldencode.com>

Greg,

I think I have a fix for this (conversion + runtime). I need to clean it up and test it + regression testing.

You'll have the update in 2 hours or so.

The resulted code for a APPLY <event> TO <handle>. will look like LogicalTerminal.apply(<handle or WrappedResource>, <event>).

Thanks,
Constantin

#3 - 01/22/2015 09:56 AM - Greg Shah

Code Review ca_upd20150122a.zip

This is really good. I like how you solved the ID lookup problem by using a different apply() signature in ClientExports. Really good work!

1. ThinClient needs a history entry.
2. On LogicalTerminal line 6469, please change the use of getValue() to toStringMessage().
3. In LogicalTerminal.apply(WrappedResource res, double eventCode), could a non-widget/non-frame resource ever be something other than a ExternalProgramWrapper? For example, could we see someone apply events to a SERVER-SOCKET?
4. In LogicalTerminal, the error 3190 generation can be put into a method to reduce code duplication (perhaps with a boolean flag to add the 3135 error to the list or not).
5. Please add parenthesis to the inner ternary on line 386 of EventDefinition to make it easier to read.

#4 - 01/22/2015 10:24 AM - Constantin Asofiei

Greg Shah wrote:

3. In LogicalTerminal.apply(WrappedResource res, double eventCode), could a non-widget/non-frame resource ever be something other than a ExternalProgramWrapper? For example, could we see someone apply events to a SERVER-SOCKET?

I think the only exceptions are the system handles (like SESSION, ERROR-STATUS, etc). Other resources - socket,server-socket,buffer,temp-table accept APPLY and trigger phrases.

I'll upload a revised update later today.

#5 - 01/22/2015 12:37 PM - Constantin Asofiei

There is this case also:

```
on "4" of ht, hb, hs, hss, default-window do:  
  message "4" self:type " resource "  
end.
```

which combines widgets with handles, for the same event. Now I get a compile error, as the `EventList.addEvent` does not accept mixed `GenericWidget/handle` targets. Should I continue fixing this?

Also, note that with this conversion change, constructs like `apply "e" to current-window` will convert to `apply(currentWindow(), "e")` instead of `currentWindow().unwrapWidget().apply("e")`.

#6 - 01/22/2015 12:52 PM - Greg Shah

Should I continue fixing this?

Yes, if it can be done in a reasonable amount of time. I think it is time to get this right.

#7 - 01/22/2015 01:19 PM - Constantin Asofiei

- File `ca_upd20150122c.zip` added

Attached fixes the issues in notes 3 and 5. Testing is under way.

#8 - 01/22/2015 01:37 PM - Greg Shah

Code Review `ca_upd20150122c.zip`

It looks good.

#9 - 01/23/2015 07:13 AM - Constantin Asofiei

Greg Shah wrote:

Code Review `ca_upd20150122c.zip`

It looks good.

`0122c.zip` passed runtime/conversion testing.

#10 - 01/23/2015 08:28 AM - Greg Shah

Please check it in.

#11 - 01/23/2015 01:29 PM - Constantin Asofiei

- Status changed from WIP to Review

Greg Shah wrote:

Please check it in.

Released to rev 10719.

#12 - 01/23/2015 01:34 PM - Greg Shah

- Status changed from Review to Closed

#13 - 01/27/2015 04:26 AM - Constantin Asofiei

- File *ca_upd20150127a.zip* added

Fixed some conversion issues related to APPLY TO handle.

Anything which has a frame-id needs to convert as a GenericWidget:

```
def temp-table ttl field f1 as int field f2 as int.  
  
form ttl.f1 ttl.f2 with frame tf.  
  
on "any-printable" of ttl.f1, ttl.f2 in frame tf do:  
  message "a".  
end.
```

and importing of LogicalTerminal when there is only:

```
apply "close" to this-procedure.
```

#14 - 01/27/2015 04:26 AM - Constantin Asofiei

ca_upd20150127a.zip passed conversion testing and was released to bzs rev 10723

#15 - 11/16/2016 12:13 PM - Greg Shah

- Target version changed from Milestone 12 to GUI Support for a Complex ADM2 App

Files

ca_upd20150122a.zip	265 KB	01/22/2015	Greg Shah
ca_upd20150122c.zip	269 KB	01/22/2015	Constantin Asofiei
ca_upd20150127a.zip	22.6 KB	01/27/2015	Constantin Asofiei