

Base Language - Feature #1612

finish SESSION system handle support

10/20/2012 09:34 AM - Greg Shah

Status:	Closed	Start date:	02/18/2013
Priority:	Normal	Due date:	08/05/2013
Assignee:	Igor Skornyakov	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	GUI Support for a Complex ADM2 App	vendor_id:	GCD
billable:	No		
Description			
Subtasks:			
Feature # 2003: conversion support for SESSION handle			Closed
Feature # 2004: runtime support for SESSION handle			Closed

History

#1 - 10/31/2012 01:46 PM - Greg Shah

- Target version set to Milestone 7

#2 - 02/18/2013 05:43 PM - Greg Shah

The following are high priority:

Attributes:

CLIENT-TYPE
FIRST-BUFFER
NUMERIC-SEPARATOR
NUMERIC-DECIMAL-POINT
NUMERIC-FORMAT
PARAMETER
REMOTE
WINDOW-SYSTEM

Methods:

GET-WAIT-STATE
SET-NUMERIC-FORMAT
SET-WAIT-STATE

Lower priority:

Attributes:

Please create the list of remaining attributes (not listed above and not already implemented).

Methods:

GET-PRINTERS

List of not supported attributes:

- progress.g parser errors:

```
current-request-info
current-response-info
error-stack-trace
first-form
first-object
last-form
last-object
local-version-info
server-connection-bound-req
uest
suppress-warnings-list
timezone
wc-admin-app
```

- methods_attributes.rules failure:

```
appl-alert-boxes /* KW_APPL_A_B */
base-ade /* KW_BASE_ADE */
context-help-file /* KW_CTX_H_F */
cpcase /* KW_CPCASE */
cpcoll /* KW_CPCOLL */
cpinternal /* KW_CPINTERN */
cplog /* KW_CPLOG */
cpprint /* KW_CPPRINT */
cprcodein /* KW_CPRCODEI */
cprcodeout /* KW_CPRCODEO */
cpstream /* KW_CPSTREAM */
cpterm /* KW_CPTERM */
data-entry-return /* KW_DATA_E_R */
debug-alert /* KW_DBG_ALRT */
display-timezone /* KW_DISP_TZ */
execution-log /* KW_EXEC_LOG */
first-dataset /* KW_FIR_DSET */
first-data-source /* KW_FIRST_DS */
first-query /* KW_FIRST_QR */
first-server /* KW_FIRST_SR */
first-server-socket /* KW_FIRST_SS */
first-socket /* KW_FIRST_SO */
frame-spacing /* KW_FR_SPACE */
icfparameter /* KW_ICFPARM */
immediate-display /* KW_IMM_DISP */
inherit-bgcolor /* KW_INH_BGC */
inherit-fgcolor /* KW_INH_FGC */
instantiating-procedure /* KW_INST_PRC */
last-server /* KW_LAST_SRV */
last-server-socket /* KW_LAST_SS */
last-socket /* KW_LAST_SOC */
multitasking-interval /* KW_MULTI_IN */
pixels-per-column /* KW_PIX_COL */
pixels-per-row /* KW_PIX_ROW */
printer-control-handle /* KW_PRT_C_H */
printer-hdc /* KW_PRT_HDC */
printer-name /* KW_PRT_NAME */
printer-port /* KW_PRT_PORT */
proxy-password /* KW_PROX_PWD */
proxy-userid /* KW_PROX_UID */
schema-change /* KW_SCH_CHG */
server-connection-bound /* KW_SRV_C_B */
server-connection-context /* KW_SRV_C_C */
server-connection-id /* KW_SRV_C_I */
server-operating-mode /* KW_SRV_OP_M */
startup-parameters /* KW_STUP_PAR */
stream /* KW_STREAM */
suppress-warnings /* KW_SUP_WARN */
system-alert-boxes /* KW_SYS_A_B */
tooltips /* KW_TOOLTIPS */
```

```
v6display /* KW_V6DISP */
work-area-height-pixels /* KW_WA_H_P */
work-area-width-pixels /* KW_WA_W_P */
work-area-x /* KW_WA_X */
work-area-y /* KW_WA_Y */
```

- Generated code incorrect:

```
first-child /* KW_FIRST_CH */
handle /* KW_HANDLE */
last-child /* KW_LAST_CH */
next-sibling /* KW_NEXT_SIB */
three-d /* KW_3d */
```

It will not take long to add stubs for the high-priority attributes/methods. Beside the REMOTE attribute (which is common to procedures too), these attrs/methods should be added to:

- CLIENT-TYPE - LogicalTerminal
- FIRST-BUFFER - Eric: is it OK BufferImpl ?
- NUMERIC-SEPARATOR - NumberType or LogicalTerminal
- NUMERIC-DECIMAL-POINT - NumberType or LogicalTerminal
- NUMERIC-FORMAT - NumberType or LogicalTerminal
- PARAMETER - SessionUtils (is the value of the Parameter (-param) startup parameter specified for the current session).
- REMOTE - need to determine the interface
- WINDOW-SYSTEM - LogicalTerminal
- GET-WAIT-STATE - LogicalTerminal (is related to how the "busy" mouse pointer looks)
- SET-NUMERIC-FORMAT - NumberType or LogicalTerminal
- SET-WAIT-STATE - LogicalTerminal (is related to how the "busy" mouse pointer looks)

I suggest coding the high-priority attrs/methods on top of the [#2002](#) update and test them together.

#4 - 02/21/2013 11:47 AM - Constantin Asofiei

- File *ca_upd20130221d.zip* added

Added support for the high-priority stuff (built on top of revision 10179, was easier that way) plus stubs for some appserver-related attributes.

#5 - 02/22/2013 03:19 AM - Constantin Asofiei

- File *ca_upd20130222b.zip* added

Merged with 10183, fixed some runtime errors too (ERROR-STATUS handle could not initialize its static proxy).

#6 - 02/22/2013 03:22 AM - Constantin Asofiei

- File *ca_upd20130222c.zip* added

Added tests.

#7 - 02/22/2013 11:10 AM - Greg Shah

The code looks fine. I'll re-check after the merge.

#8 - 02/22/2013 11:43 AM - Constantin Asofiei

- File *ca_upd20130222g.zip* added

Merged with revision 10186.

#9 - 02/22/2013 12:02 PM - Greg Shah

The update looks good. I'm testing it now.

#10 - 02/22/2013 01:03 PM - Greg Shah

Conversion testing has passed. Please check in and distribute.

#11 - 02/22/2013 01:16 PM - Constantin Asofiei

Committed to bzt revision 10187.

#12 - 09/26/2013 05:46 AM - Constantin Asofiei

Following attributes will be added first

- client-type
- parameter
- first-buffer

#13 - 09/26/2013 06:46 AM - Constantin Asofiei

Greg, how do you suggest to implement the PARAMETER attribute? The SESSION:PARAMETER attribute holds the value of the -param argument at the program startup. We have several choices in P2J:

1. via -D arguments at client startup (and then interrogate it using our OS-GETENV implementation)
2. configure it in the directory (but how? per each program? per P2J user?)
3. currently, the "-a apparg1 apparg2 ... appargn" arguments are not used by the ClientDriver. We can expand this and let the legacy -param value arguments be set via ClientDriver's -a appargs support.
4. we can add a new argument to ClientDriver, -param.

#14 - 09/26/2013 08:23 AM - Greg Shah

For now, just implement it in the runtime section of the directory. Use the Directory interface with ID_RELATIVE which uses Utils.getDirectoryNode*()

and has a set of search paths. This allows per-account, per-group, per-server and default configuration.

Later in the project we will handle task [#1842](#). Please look there to see how we are going to provide more flexibility.

#15 - 09/26/2013 09:43 AM - Constantin Asofiei

I've been looking at the FIRST-QUERY attribute, and I think we need to change something in the FIRST/LAST-resource implementation. Currently, HandleChain keeps context-local maps of first/last resources, where the key is the implementing class. But, for the FIRST-QUERY resource, I think we can have multiple implementations (as AbstractQuery, which extends HandleChain, is abstract).

To solve this, we could have a LegacyResource annotation to annotate the interface associated with the resource definition, but this would be redundant with the resource type returned by CommonHandle.getResourceType()

The other approach is to use the TYPE attribute, but currently at the time the HandleChain() c'tor is called, the TYPE attribute is not set; this will require changes in all resources, to call the HandleChain(String resourceType) c'tor. And we should have exposed constants with the resource type, for all resources.

#16 - 09/26/2013 09:50 AM - Greg Shah

To solve this, we could have a LegacyResource annotation to annotate the interface associated with the resource definition, but this would be redundant with the resource type returned by CommonHandle.getResourceType()

Can we change the getResourceType() implementation to use the LegacyResource annotation as its source of TYPE info?

#17 - 09/26/2013 10:02 AM - Constantin Asofiei

Can we change the getResourceType() implementation to use the LegacyResource annotation as its source of TYPE info?

Thanks, this is the way to go. We just need to remove HandleResource.setResourceType() and let getResourceType() to determine it from the LegacyResource annotation (and cache it on the first call). Note that this will be only for non-system handles; system handle implementations (like SessionUtils.getResourceType()) still need to hard-code the type, but their interface (i.e. CommonSession) should be annotated.

The LegacyResource interface should look like:

```
package com.goldencode.p2j.util;

import java.lang.annotation.*;

/**
 * Annotation definition to mark the resource type implemented by a P2J class.
 */
@Target(ElementType.TYPE)
@Retention(RetentionPolicy.RUNTIME)
public @interface LegacyResource
```

```
{  
    String resource();  
  
    /** Constant identifying the SOCKET resource. */  
    public static String SOCKET = "SOCKET";  
  
    /** TODO: constants for other resources */  
}
```

and the annotation:

```
@LegacyResource(resource = LegacyResource.SOCKET)  
public interface Socket
```

I can do this part of this task.

#18 - 09/26/2013 10:09 AM - Greg Shah

I like it. Two things:

1.
system handle implementations (like SessionUtils.getResourceType()) still need to hard-code the type

OK. But do use the LegacyResource.SESSION (or whatever is appropriate) constant instead of duplicating the "SESSION" string in SessionUtils (or other system handle imple class).

2. Please modify the javadoc for the LegacyResource like this:

```
/**  
 * Annotation definition to mark the Progress 4GL resource type implemented by a P2J  
 * class. These resources are referenced by handle in the 4GL and these constants  
 * correspond to the TYPE attribute that would be returned in the 4GL.  
 */
```

#19 - 09/26/2013 11:50 AM - Constantin Asofiei

There are some more issues:

1. for widgets, the resource type can't be set at the interface, because they all grow from `CommonWidget`. In this case, we can set it at the class. But, there are widgets implementations (like `ControlTextWidget` and `BrowseColumnWidget` classes) which can represent more than one resource.
2. `GenericFrame` extending `HandleChain` I think is wrong, because the frame resource (as widget) is implemented by `FrameWidget`
3. `WrappedResource.unknown()` is fully deprecated... a resource can't be unknown, only a handle can be; the resource can at most be invalid. This API should be removed, as it was added when the handle support did not exist in P2J, and `SELF/FOCUS` had to return a `NullWidget` instance. BTW, `NullWidget` I think is deprecated too.

#20 - 09/26/2013 12:18 PM - Greg Shah

for widgets, the resource type can't be set at the interface, because they all grow from `CommonWidget`. In this case, we can set it at the class.

OK.

But, there are widgets implementations (like `ControlTextWidget` and `BrowseColumnWidget` classes) which can represent more than one resource.

I don't fully understand this. For example, `ControlTextWidget` is extended by both `EditorWidget` and `FillInWidget`. Can't we put the annotation on the child classes?

`GenericFrame` extending `HandleChain` I think is wrong, because the frame resource (as widget) is implemented by `FrameWidget`

OK.

`WrappedResource.unknown()` is fully deprecated... a resource can't be unknown, only a handle can be; the resource can at most be invalid. This API should be removed, as it was added when the handle support did not exist in P2J

OK.

BTW, `NullWidget` I think is deprecated too.

I'm not as sure about this. We have a lot of safety code that returns this instead of null. It is not clear from the code if this can never be accessed.

But, there are widgets implementations (like `ControlTextWidget` and `BrowseColumnWidget` classes) which can represent more than one resource.

I don't fully understand this. For example, `ControlTextWidget` is extended by both `EditorWidget` and `FillInWidget`. Can't we put the annotation on the child classes?

`ControlTextWidget` is emitted in these two cases:

1. form "bar" with frame f1. - "bar" will be emitted as:

```
ControlTextWidget expr1 = new ControlTextWidget();

public void setup(CommonFrame frame)
{
    frame.setDown(1);
    expr1.setStatic(true);
    expr1.setDataTypes("character");
    expr1.setFormat("x(3)");
    ((ResourceTypeF1) frame).setExpr1(new character("bar"));
}
```

2. display "foo" with frame f3. - "foo" will be emitted as (at frame definition):

```
ControlTextWidget expr3 = new ControlTextWidget();

public void setup(CommonFrame frame)
{
    frame.setDown(1);
    expr3.setDataTypes("character");
    expr3.setFormat("x(3)");
}
```

and the display:

```
FrameElement[] elementList0 = new FrameElement[]
{
    new Element("foo", f3Frame.widgetExpr3())
};

f3Frame.display(elementList0);
```

The resource type is differentiated after the "static" flavor of the widget. For the first case, the widget type is LITERAL, while for the second is FILL-IN.

BTW, `NullWidget` I think is deprecated too.

I'm not as sure about this. We have a lot of safety code that returns this instead of null. It is not clear from the code if this can never be accessed.

`NullWidget` is used in these places:

1. `LogicalTerminal._self()` and `LogicalTerminal.focus()` - this is wrong, these two should return an unknown handle if needed, not `NullWidget`.
2. `BaseEntity.getNextSibling()` and `BaseEntity.getPrevSibling()` - these two already return handle, so I think is OK to remove `NullWidget`.
3. `FieldGroup.getNextChild()` and `FieldGroup.getPrevChild()` - is used only by the `BaseEntity.getNext/PrevSibling()` methods, so is OK to return null.
4. `FieldGroup.getFirstChild()` and `FieldGroup.getLastChild()` - these two must return handle, as the LAST-CHILD and FIRST-CHILD are handle-type attributes. If they don't return a handle, then the following code will not convert properly in P2J:


```
def var h as handle.  
h = frame f1:handle.  
h = h:first-child:first-child.
```

Looking at the NullWidget history entries, I think the goal was to protect against NPEs LogicalTerminal.focus(). Also, NullWidget.showError shows an error message, but no ERROR condition is raised; and something weird about how FOCUS works in 4GL:

5. FOCUS:something will not raise error, if FOCUS is unknown, but ERROR-STATUS:ERROR will be set, in case of NO-ERROR.
6. the following will raise an error:

```
def var h as handle.  
def var ch as char.  
  
h = focus.  
ch = h:type.
```

It's like FOCUS referencing an unknown widget is something special in 4GL.

My conclusion is that initially NullWidget was added for a FOCUS problem, and later expanded to other parts of code. I think I will use this approach:

1. I will leave NullWidget in place for now, in LogicalTerminal._self() and LogicalTerminal.focus(), but we will need to test it carefully during [#1779](#) - FOCUS implementation.
2. we should fix FIRST-/LAST-CHILD support, to allow chaining and to properly return a handle instance (possible unknown) instead of NullWidget. Also, note that when a handle is assigned a NullWidget, handle.unwrap will notice that the resource is unknown and a InvalidAttributeAccess proxy will be returned, which bypasses the NullWidget implementation. Thus, in this case, NullWidget is not used at all.

#22 - 09/26/2013 01:37 PM - Greg Shah

OK, I'm fine with your approach to all of this. Make some notes in [#1779](#) (FOCUS) and also [#2015](#) (SELF), so that we don't forget to clean that up.

The resource type is differentiated after the "static" flavor of the widget. For the first case, the widget type is LITERAL, while for the second is FILL-IN.

Is this just a quirk of Progress? In the DISPLAY case, can you actually use the field for editing? There is no lvalue and no way to read the value because the widget has no name.

Perhaps we need to create two subclasses of ControlTextWidget: LiteralTextWidget and MostlyLiteralTextWidget :)

The only thing these classes would have is different LegacyResource annotations, otherwise everything else would be implemented in ControlTextWidget. Then we change the frame defs to emit these in place of ControlTextWidget.

#23 - 09/27/2013 05:44 AM - Constantin Asofiei

Is this just a quirk of Progress? In the DISPLAY case, can you actually use the field for editing? There is no lvalue and no way to read the value because the widget has no name.

Yes, 4GL allows editing of such fields... and I think is the same as creating a dynamic widget, which is not attached to a lvalue; the following works and will enable both fill-in's:

```
def var h as handle.  
  
display "foo" with size 10 by 10 frame f3.  
  
create fill-in h  
  assign  
    row = 2  
    column = 1  
    frame = frame f3:handle  
    visible = true  
    screen-value = "bar".  
  
enable all with frame f3.  
  
wait-for close of current-window.
```

#24 - 09/27/2013 08:00 AM - Constantin Asofiei

I think I've got it how 4GL interprets string literals in DISPLAY and FORM statements. See this test:

```
def var h as handle.  
def var ch as char.  
def var i as int.  
def var l as log.  
  
function checkType returns int (i as int, h as handle, t as char).  
  if (h:type <> t)  
    then message "for case" string(i) h:type "is incorrect; expecting" t.  
end.  
  
function firstChild returns handle (h as handle).  
  h = h:first-child.
```

```
h = h:first-child.  
return h.  
end.
```

```
form "f1" with frame f1.  
h = firstChild(frame f1:handle).  
hide frame f1 no-pause.  
checkType(1, h, "LITERAL").
```

```
form "f2" view-as text with frame f2.  
h = firstChild(frame f2:handle).  
hide frame f2 no-pause.  
checkType(2, h, "LITERAL").
```

```
display "f3" with frame f3.  
h = firstChild(frame f3:handle).  
hide frame f3 no-pause.  
checkType(3, h, "FILL-IN").
```

```
display "f4" view-as text with frame f4.  
h = firstChild(frame f4:handle).  
hide frame f4 no-pause.  
checkType(4, h, "TEXT").
```

```
display "f5" view-as fill-in with frame f5.  
h = firstChild(frame f5:handle).  
hide frame f5 no-pause.  
checkType(5, h, "FILL-IN").
```

We have 3 cases:

1. a string literal with a FORM statement will always be LITERAL, regardless if VIEW-AS TEXT is used.
2. a string literal with a DISPLAY statement will always be FILL-IN, unless the VIEW-AS TEXT is used.
3. a string literal with a DISPLAY statement will be TEXT, if the VIEW-AS TEXT is used.

So, we need to make ControlTextWidget abstract and add these classes:

1. TextWidget, emitted for the case of DISPLAY <string-literal> VIEW-AS TEXT.
2. LiteralWidget, emitted for any case of FORM <string-literal>
3. also, fix the DISPLAY <string-literal> case to convert to a FillInWidget

#25 - 09/27/2013 08:25 AM - Greg Shah

Good job. This makes sense and it results in a cleaner, more understandable set of frame definitions.

#26 - 10/01/2013 11:06 AM - Constantin Asofiei

I've started working on the format-related stuff: NUMERIC-SEPARATOR, @NUMERIC-DECIMAL-POINT, NUMERIC-FORMAT, and SET-NUMERIC-FORMAT, and the problem is that in 4GL, widget's format is dynamic (i.e. if separators change, then the format is aware of this change):

```
def var d as decimal init 12345678.09 format ">>>, >>>, >>>.99".

session:numeric-format = "european".
display d with frame f1 2 down.
down with frame f1.

session:numeric-format = "american".
display d with frame f1.
down with frame f1.
```

will display this:

d
12.345.678,09
12,345,678.09

The problem in P2J is that we cache the explicit format set at the frame/widget definitions, including the separators. The good part is that in 4GL the separators for the format are always AMERICAN (, as group and . as decimal separator). I will change NumberFormat so that when parsing the format, it will use the american-style separators, but when building it for display, it will use the active format.

#27 - 10/01/2013 01:16 PM - Constantin Asofiei

- File *ca_upd20131001b.zip* added

This update is only for SESSION's format-related attributes/methods. I'm putting it through regression testing, but I have a few issues encountered during testing:

1. SCREEN-VALUE attribute doesn't seem to be fixed properly. See the GenericFrame changes - without these, this test will fail:

```
def var d as dec label "Decimal" init 12345678.09 format ">>>,>>>,>>>.99".

def var sdf as char.
def var df as char init "12,345,678.09".

form d with frame f1.
display d with frame f1.

sdf = trim(string(d, ">>>,>>>,>>>.99")).
if (sdf <> df)
  then message "expected " df " and found " sdf.

sdf = d:screen-value in frame f1.
if (sdf <> df)
  then message "expected " df " and found screen-value " sdf.
```

I used this fix just to let my numeric-format related tests to not show false negatives, but I have a feeling is not complete. I think I should make a separate task for this problem, as the trim call on line 2554 is numeric-specific. Also, my feeling is that SCREEN-VALUE must return what is actually in the frame's screen buffer for that widget, and not rely on type-specific values.

2. remember the `ErrorHandler.recordOrShowError` changes related to cases when `ERROR` flag is not set, but error message is recorded? Looks like they are not enough; I feel like I'm chasing my own tail as I should have tested this before, because the following will fail in P2J:

```
def var ch as char init ?.
session:set-numeric-format(ch, ch) no-error.
message error-status:num-messages. /* here is 1 */
session:set-numeric-format(ch, ch) no-error.
message error-status:num-messages. /* here must be 1 too */
```

This fails as the second statement will not clear the error list, and will register a new error on top of the previous one. I think Evgeny's initial fix to clear the error list when `silentErrorEnable` is called is the one to use, but I can't explain why `ErrorHandler` was initially coded to clear the list when `silentErrorDisable` is called - maybe there is a case when we don't need to clear the list? This is what bugs me.

The other update - with the `LegacyResource`-related changes - has still some runtime problems after refactoring `ControlTextWidget` into `LiteralWidget` and `TextWidget` (as the conversion rules don't emit them in proper cases), so I'll do this: tomorrow I'll extract from that update all changes not related to `LegacyResource` into a separate update and put them here. I want to release them in phases, as the `LegacyResource`-related changes touch lots of files and will take some more time to fix the problems.

#28 - 10/01/2013 07:33 PM - Greg Shah

Code Review 1001b

I am fine with the changes. It pains me to see how I got the initial group/decimal separator usage wrong. :) In particular, I am surprised that so much of the 4GL parsing is hard coded to AMERICAN format.

I used this fix just to let my numeric-format related tests to not show false negatives, but I have a feeling is not complete. I think I should make a separate task for this problem, as the trim call on line 2554 is numeric-specific.

I am not surprised that the SCREEN-VALUE is broken. We have never gotten it quite right. Yes, do create another task. Set it for milestone 12 since we should be able to get through milestones 7 and 11 without the interactive code.

Also, my feeling is that SCREEN-VALUE must return what is actually in the frame's screen buffer for that widget, and not rely on type-specific values.

This sounds right.

I think Evgeny's initial fix to clear the error list when silentErrorEnable is called is the one to use, but I can't explain why ErrorManager was initially coded to clear the list when silentErrorDisable is called - maybe there is a case when we don't need to clear the list? This is what bugs me.

No, something else is wrong. I know that clearing in the disable is correct (or mostly correct). It is possible to see the old list during the new no-error statement/expression's processing:

```
def var ch as char init ?.
session:set-numeric-format(ch, ch) no-error.
message error-status:num-messages. /* here is 1 */
if error-status:num-messages < 1 then message "There should have been an error message logged.". else session:
set-numeric-format(ch, ch) no-error. /* if we clear it in the enable, then this would fail */
message error-status:num-messages. /* here must be 1 too */
```

#29 - 10/02/2013 01:59 AM - Constantin Asofiei

- File `ca_upd20131002a.zip` added

Attached update adds runtime support for these numeric-format related attributes and methods:

```
NUMERIC-SEPARATOR  
NUMERIC-DECIMAL-POINT  
NUMERIC-FORMAT  
SET-NUMERIC-FORMAT ()
```

Passed runtime testing. Committed to bzt revision 10391.

#30 - 10/02/2013 08:09 AM - Greg Shah

Looking back at my example, I don't think I showed exactly what I was intending to show.

The key point that I recall is that it is possible to access the old error stack during no-error processing of the next expression or statement. Please look into this point and see if that helps clarify. If I am correct, then we will need to find a different fix and we also need to add some javadoc explaining this.

#31 - 10/02/2013 10:15 AM - Constantin Asofiei

- File `ca_upd20131002d.zip` added

This update adds runtime for the following:

```
CLIENT-TYPE  
PARAMETER  
GET-WAIT-STATE ()  
SET-WAIT-STATE ()
```

Note that GET-/SET-WAIT-STATE() are a no-op in CHUI/batch mode, they work only in GUI mode.

About WINDOW-SYSTEM - from the documentation:

```
• For graphical interfaces in Windows:  
- If the display is set to the Windows XP Theme, and a manifest file is used, the value  
is "MS-WINXP".  
- If the display is set to the Windows XP Theme, and a manifest file is not used, the  
value is "MS-WIN95".  
- If the display is set to the Windows Classic Theme, the value is "MS-WIN95".  
• For character interfaces, the value is "TTY".  
ABL supports an override option that enables applications that need the WINDOW-SYSTEM  
attribute to return the value of "MS-WINDOWS" for all Microsoft operating systems to do so. To  
establish this override capability, define the WindowSystem key in the Startup section in the  
current environment, which might be the registry or an initialization file. If the WindowSystem  
key is located, the WINDOW-SYSTEM attribute returns the value associated with the  
WindowSystem key on all platforms.
```

For M7, I think is enough to let it return "TTY". But, if the full implementation would mean to read this value using `Directory.getString(Directory.ID_RELATIVE, "window-system")`, I can add this too.

#32 - 10/02/2013 10:28 AM - Greg Shah

Code Review 1002d

1. LT.getClientType() returns "4GLCLIENT" but the javadoc states it would return "CLIENT" instead.
2. Add a redmine task to the UI project to implement the backing support for GET-/SET-WAIT-STATE. Assign it to milestone 12.
3. In regard to WINDOW-SYSTEM:
 - For M7, I think is enough to let it return "TTY".

Let's go ahead and implement this properly. The idea: the client should "register" the right value by either:

- a. Calling an entry point in the server with the WINDOW-SYSTEM text (TTY...). This should be called just after authentication but before the standard entry point is invoked.
- b. Enhance the standard entry point to take an object that can hold client-defined non-changing data. The WINDOW-SYSTEM value could be passed in that object. The code in StandardServer is specific to P2J (and the 4GL behavior like the transaction processing and STOP disposition). So it is not unreasonable to pass 4GL-specific client values.

But, if the full implementation would mean to read this value using Directory.getString(Directory.ID_RELATIVE, "window-system"), I can add this too.

Yes, add this override capability, exactly as you have suggested.

#33 - 10/02/2013 10:35 AM - Constantin Asofiei

2. Add a redmine task to the UI project to implement the backing support for GET-/SET-WAIT-STATE. Assign it to milestone 12.

See [#2189](#)

- b. Enhance the standard entry point to take an object that can hold client-defined non-changing data. The WINDOW-SYSTEM value could be passed in that object. The code in StandardServer is specific to P2J (and the 4GL behavior like the transaction processing and STOP disposition). So it is not unreasonable to pass 4GL-specific client values.

I like this better, but in both cases, from where does the client initially read the WINDOW-SYSTEM value?

#34 - 10/02/2013 10:46 AM - Greg Shah

from where does the client initially read the WINDOW-SYSTEM value?

At startup, the client must already know what driver is going to be used. We can base the value on that.

#35 - 10/03/2013 10:47 AM - Constantin Asofiei

- File *ca_upd20131003c.zip* added

This is for the LegacyResource and ControlTextWidget related changes (is merged with latest rev, 10392); also, it adds support for FIRST-/LAST-related attributes. Does not contain changes in 1002d.zip.

Although runtime testing has passed, I need to do some more manual testing of the conversion-related changes (I want to make sure I have tests to reach all changed paths).

#36 - 10/03/2013 11:51 AM - Greg Shah

Code Review 1003c

Really good! The only feedback I have is that I would like you to add a Redmine task to describe the fix needed for returning the proper resource type for browse columns.

If all your testing passes, you can check this in.

#37 - 10/04/2013 07:40 AM - Constantin Asofiei

- File *ca_upd20131004b.zip* added

This version was committed to bzt revision 10393. Contains a change related to unknown() API which still needs to be statically implemented by resources which are mapped via static proxies (i.e. SESSION, LAST-EVENT, ERROR-STATUS and FILE-INFO).

#38 - 10/04/2013 09:22 AM - Constantin Asofiei

Greg Shah wrote:

from where does the client initially read the WINDOW-SYSTEM value?

At startup, the client must already know what driver is going to be used. We can base the value on that.

Thinking about it, I don't think is OK to hard link the value of this attribute with the driver. We should relax this, and allow it to be configured via a "client:driver>window-system" setting. As this attribute can have 3 values for GUI mode: MS-WINXP, MS-WIN95 and MS-WINDOWS, we will be forced to have 3 different driver types, one for each possible attribute value.

#39 - 10/04/2013 09:28 AM - Greg Shah

We should relax this, and allow it to be configured via a "client:driver>window-system" setting.

I'm OK with this, but only as an override. I want there to be a reasonable default that comes from the client. If the client is running a GUI driver, it should set the most common value (I'm not sure what it is). We can check windev01 to see what it returns.

The CHUI drivers should set "TTY" as the default.

#40 - 10/04/2013 09:59 AM - Constantin Asofiei

If the client is running a GUI driver, it should set the most common value (I'm not sure what it is). We can check windev01 to see what it returns.

On devsrv01, it returns MS-WIN95. But I think this is only because of the theme used. In 4GL, the theme can be ignored when computing this attribute only if some registry key is set (see note 31 for the docs); in this case, it reports MS-WINDOWS for any theme. I guess we can use this as a default for GUI drivers.

#41 - 10/04/2013 10:14 AM - Constantin Asofiei

- File *ca_upd20131004c.zip* added

This is built on top of rev 10393 and on top of 1002d. Adds support for these attributes:

```
CLIENT-TYPE  
PARAMETER  
WINDOW-SYSTEM  
GET-WAIT-STATE ()
```

SET-WAIT-STATE ()

After this update, all the high-priority attributes/methods on note 2 are implemented.

#42 - 10/04/2013 11:28 AM - Greg Shah

Code Review 1004c

1. ClientParameters copyright date should be 2013 instead of 2005-2013.
2. The javadoc in ClientParameters states that MS-WINXP will be the default for GUI. I thought you were going to use MS-WINDOWS for that.
3. Actually, I was thinking the WINDOW-SYSTEM override would be read from the directory (on the server side, not the client) from the runtime area. I'm OK with also supporting a bootstrap config value, but I'd like to be able to control things from the directory too.

#43 - 10/04/2013 11:54 AM - Constantin Asofiei

- File *ca_upd20131004d.zip* added

1. ClientParameters copyright date should be 2013 instead of 2005-2013.

Fixed

2. The javadoc in ClientParameters states that MS-WINXP will be the default for GUI. I thought you were going to use MS-WINDOWS for that.

Correct, fixed.

3. Actually, I was thinking the WINDOW-SYSTEM override would be read from the directory (on the server side, not the client) from the runtime area. I'm OK with also supporting a bootstrap config value, but I'd like to be able to control things from the directory too.

OK, the search is done using this precedence (lowest to highest):

1. ScreenDriver.getWindowSystem
2. client:driver>window-system from client side config
3. window-system node from the directory

#44 - 10/04/2013 02:08 PM - Greg Shah

Code Review 1004d

I'm OK with this. Check it in if it passes testing.

#45 - 10/05/2013 09:15 AM - Constantin Asofiei

1004d.zip was committed to bazaar revision 10394.

#46 - 10/06/2013 09:52 AM - Greg Shah

- Target version changed from Milestone 7 to Milestone 12

The remaining work for this task is not needed until M12 so it has been re-assigned to that milestone. A sub-task to track the work can be created later. At this time, we only know of the GET-PRINTERS method that remains to be implemented for the current project. That may change when we re-do the reports for the GUI project.

#47 - 10/07/2013 05:25 AM - Constantin Asofiei

- File high_priority_attrs_tests_20131007a.zip added

Attached testcases for the high-priority attributes/methods.

#48 - 08/27/2015 05:24 PM - Greg Shah

For our next project, the following attributes need to be implemented with high priority:

APPL-ALERT-BOXES
DATA-ENTRY-RETURN
DEBUG-ALERT
IMMEDIATE-DISPLAY

#49 - 02/19/2016 08:53 AM - Greg Shah

- Assignee set to Igor Skornyakov

#50 - 03/01/2016 06:05 AM - Igor Skornyakov

created task branch 1612a

#51 - 03/01/2016 07:18 AM - Igor Skornyakov

Added conversion support and stub for the SESSION:IMMEDIATE-DISPLAY, write support for the SESSION:DATA-ENTRY-RETURN

Committed to the task branch 1612a revision 10976.

#52 - 03/01/2016 01:14 PM - Igor Skornyakov

Rebased task branch 1612a from P2J trunk revision 10976. The latest revision is now 10977.

#53 - 03/02/2016 03:22 AM - Igor Skornyakov

Rebased task branch 1612a from P2J trunk revision 10978. The latest revision is now 10979.

#54 - 03/02/2016 07:13 AM - Igor Skornyakov

- File t1612.p added

The value of the APPL-ALERT-BOXES doesn't affect MESSAGE statement with SET/UPDATE clause.

#55 - 03/02/2016 07:57 AM - Igor Skornyakov

Added state variables for the APPL-ALERT-BOXES, DATA-ENTRY-RETURN, DEBUG-ALERT and IMMEDIATE-DISPLAY SESSION attributes. Implemented runtime support for the APPL-ALERT-BOXES.

Committed to the task branch 1612a revno 10980.

#56 - 03/02/2016 08:16 AM - Greg Shah

Please document (here) the found behavior for each of the attributes.

Also, please split the testcase into separate testcases for each attribute and name the testcases descriptively, as I have previously discussed. Then check those testcases into testcases/uast/session/.

#57 - 03/02/2016 08:25 AM - Igor Skornyakov

Greg Shah wrote:

Please document (here) the found behavior for each of the attributes.

Also, please split the testcase into separate testcases for each attribute and name the testcases descriptively, as I have previously discussed. Then check those testcases into testcases/uast/session/.

OK. I planned to do it a little bit later today. Is it OK?

#58 - 03/02/2016 08:33 AM - Greg Shah

Yes, no problem.

#59 - 03/02/2016 11:36 AM - Igor Skornyakov

Semantics of the APPL-ALERT-BOXES, DATA-ENTRY-RETURN, DEBUG-ALERT and IMMEDIATE-DISPLAY SESSION attributes.

1. For the DATA-ENTRY-RETURN the assignment of the UNKNOWN values is a noop, for other attributes it is equivalent to false.
2. If APPL-ALERT-BOXES is set to true then the messages w/o SET/UPDATE clause are displayed as MESSAGE dialog boxes with a single OK button and Message title. The messages with SET/UPDATE clause are not affected.
3. The semantics of the DATA-ENTRY-RETURN seems to be as described in the Progress documentation (see uast/session/data-entry-return.p - tested both in ChUI and GUI mode):

```
DATA-ENTRY-RETURN attribute
The behavior of the RETURN key for the fill-in widgets of a frame.
Data type:
```

```
LOGICAL
Access:
```

```
Readable/Writeable
Applies to:
```

SESSION system handle

If TRUE, the RETURN key in a fill-in acts like a TAB, and if the fill-in is the last widget in the tab order of its parent frame and of all ancestor frames, the RETURN key applies a GO event to the frame (behavior prior to Version 7). This GO event, from a fill-in RETURN, propagates to all ancestor frames and their descendants, including siblings of the current frame and their descendants, all in the same frame family. If a widget is not a fill-in, the window system handles RETURN entries.

The default value is TRUE for character interfaces and FALSE for graphical interfaces. The AVM ignores this attribute if there is a default button on the frame.

The last clause means that if there is a default button on the frame the default value of the attribute is used.

4. In a simplest scenario when MULTITASKING-INTERVAL == 0 (default value) the IMMEDIATE-DISPLAY is false (default value) **some** output operation (screen updates of frames made visible with VIEW/ENABLE and messages in the message area) are on hold until it is true or a statement block for input (including PAUSE) operation. Moreover all pending updates (those which are made by the code but are not still visible) are blocked including those ones initiated before IMMEDIATE-DISPLAY was set to false. After the IMMEDIATE-DISPLAY was set to true or a statement blocks for input all the updates become visible. The error message resumes the output (all pending changes become visible). Contrary to what is said in the Progress documentation the DISPLAY statement is not affected, I've not noticed any impact of the IMMEDIATE-DISPLAY if MULTITASKING-INTERVAL != 0. See uast/session/immediate-display.p and the attached immediate-display.mkv screencast. Stream I/O is not affected by the IMMEDIATE-DISPLAY value.

At least the following should be checked:

2. Will opening of a new window (and close of the existing one) be on hold?
 4. Can you move, resize, minimize, maximize 4GL windows while this is active?
5. If SESSION:DEBUG-ALERT attribute is set to true or the -debugalert startup parameter is specified then for all message alert boxes (either with VIEW-AS ALERT=BOX clause or because SESSION:APPL-ALERT-BOXES is true) the Help button is added to the end of the button set and (Press HELP to view stack trace) is appended to the title. If the Help button is pressed then a new dialog which contains stack trace in the format like this:

```
[16/03/10@13:26:59.790+0200] P-009100 T-007732 1 4GL -- --> Q4 debug-alert.p (.\\debug-alert.p) at line 66
[16/03/10@13:26:59.790+0200] P-009100 T-007732 1 4GL --      Q3 debug-alert.p (.\\debug-alert.p) at line 61
[16/03/10@13:26:59.790+0200] P-009100 T-007732 1 4GL --      Q2 debug-alert.p (.\\debug-alert.p) at line 56
[16/03/10@13:26:59.790+0200] P-009100 T-007732 1 4GL --      Q1 debug-alert.p (.\\debug-alert.p) at line 51
[16/03/10@13:26:59.790+0200] P-009100 T-007732 1 4GL --      debug-alert.p (.\\debug-alert.p) at line 18
```

and two buttons: OK and Debug. The second one invokes debugger (doesn't work in windev01). See attached debug-alert.png. The height of the control in which the stack is shown is 13 lines for ChUI and 13 lines for GUI). It looks like non-editable EDITOR widget with two scroll bars (in GUI mode). In ChUI mode the stack trace is shown in the scrollable control without scrollbars and the stack trace is reversed comparing to the GUI mode.

If SESSION:DEBUG-ALERT attribute is set to true and the -clientlog startup parameter is specified then for all error messages and MESSAGE statements shown as ALERT-BOX a stack trace is written to the log after the message:

```
[16/03/10@13:26:59.790+0200] P-009100 T-007732 1 4GL -- 1 SESSION:DEBUG-ALERT = yes
[16/03/10@13:26:59.790+0200] P-009100 T-007732 1 4GL -- ** ABL Debug-Alert Stack Trace **
[16/03/10@13:26:59.790+0200] P-009100 T-007732 1 4GL -- --> Q4 debug-alert.p (.\\debug-alert.p) at line 66
[16/03/10@13:26:59.790+0200] P-009100 T-007732 1 4GL --      Q3 debug-alert.p (.\\debug-alert.p) at line 61
[16/03/10@13:26:59.790+0200] P-009100 T-007732 1 4GL --      Q2 debug-alert.p (.\\debug-alert.p) at line 56
[16/03/10@13:26:59.790+0200] P-009100 T-007732 1 4GL --      Q1 debug-alert.p (.\\debug-alert.p) at line 51
[16/03/10@13:26:59.790+0200] P-009100 T-007732 1 4GL --      debug-alert.p (.\\debug-alert.p) at line 18
[16/03/10@13:27:05.179+0200] P-009100 T-007732 1 4GL -- **Attribute SET-SELECTION for the FILL-IN fi has an in
valid value of 0. (4057)
[16/03/10@13:27:05.179+0200] P-009100 T-007732 1 4GL -- ** ABL Debug-Alert Stack Trace **
[16/03/10@13:27:05.179+0200] P-009100 T-007732 1 4GL -- --> P4 debug-alert.p (.\\debug-alert.p) at line 46
[16/03/10@13:27:05.179+0200] P-009100 T-007732 1 4GL --      P3 debug-alert.p (.\\debug-alert.p) at line 42
[16/03/10@13:27:05.179+0200] P-009100 T-007732 1 4GL --      P2 debug-alert.p (.\\debug-alert.p) at line 38
[16/03/10@13:27:05.179+0200] P-009100 T-007732 1 4GL --      P1 debug-alert.p (.\\debug-alert.p) at line 34
[16/03/10@13:27:05.179+0200] P-009100 T-007732 1 4GL --      debug-alert.p (.\\debug-alert.p) at line 19
```

The output described above is the same for all non-zero values of the LOG-MANAGER:LOGGING-LEVEL

The default value of the SESSION:DEBUG-ALERT is false. However the -debugalert startup parameter is specified its value is true.

#60 - 03/02/2016 11:50 AM - Igor Skornyakov

Tests added to uast/session revision 1475.

#61 - 03/02/2016 01:06 PM - Greg Shah

Semantics of the APPL-ALERT-BOXES, DATA-ENTRY-RETURN, DEBUG-ALERT and IMMEDIATE-DISPLAY SESSION attributes.

I assume you are planning to provide more details than currently posted in note 59?

#62 - 03/02/2016 01:07 PM - Igor Skornyakov

Greg Shah wrote:

I assume you are planning to provide more details than currently posted in note 59?

Of course. I will expand this note with new findings.

#63 - 03/02/2016 01:16 PM - Igor Skornyakov

Added uast/session/data-entry-return.p revision 1476. Note 59 has been updated

#64 - 03/03/2016 12:16 PM - Igor Skornyakov

I see the following code fragment in the ComboBox.processKeyEvent() method

```
else if (key == Keyboard.KA_RETURN && !dropDownActive)
{
    if (getDataEntryReturn())
    {
        // TODO: in 4GL, focus is not switched when ENTER is pressed in a COMBO-BOX!
        if (hasFocus())
            ThinClient.nextFocus(this);

        draw();
        requestSync();
    }
    return;
}
```

The getDataEntryReturn() method checks the value of the WindowConfig.sessDataEntryReturn field which has the comment that it is the value of the SESSION:DATA-ENTRY-RETURN attribute but is never assigned. A similar method (with the same implementation) exists in the FillIn class.

I understand that now the real value of the SESSION:DATA-ENTRY-RETURN attribute (and a check for the default button) should be used instead.

However according to my tests according to my tests the value of the SESSION:DATA-ENTRY-RETURN doesn't affect the COMBO-BOX. What is the reason of the code mentioned above despite this fact (and the comment)?

Thank you.

#65 - 03/03/2016 12:39 PM - Greg Shah

What is the reason of the code mentioned above despite this fact (and the comment)?

The comment was probably written before anyone tested SESSION:DATA-ENTRY-RETURN. As far as I know, you are the first to try it.

In regard to the call to `getDataEntryReturn()`, I imagine that this may have to do with support of the "simple mode" combo-boxes.

Eugenie: Please review note 64 and comment.

#66 - 03/03/2016 12:54 PM - Igor Skornyakov

Greg Shah wrote:

In regard to the call to `getDataEntryReturn()`, I imagine that this may have to do with support of the "simple mode" combo-boxes.

In my test the SIMPLE-MODE combo-boxes are not affected as well.

#67 - 03/03/2016 01:12 PM - Eugenie Lyzenko

The idea behind if when the ENTER key is pressing in combo-box with editable field (and drop-down is closed) or in fill-in the focus movement behavior is dependent on the SESSION:DATA-ENTRY-RETURN attribute. If the attribute is TRUE - the focus is moving to the next widget, if FALSE - focus does not change. The default value of the attribute according 4GL docs is TRUE for ChUI and FALSE for GUI. However like other documented features - this must be tested because 4GL documentation is far away from ideal.

#68 - 03/03/2016 01:15 PM - Igor Skornyakov

Eugenie Lyzenko wrote:

The idea behind if when the ENTER key is pressing in combo-box with editable field (and drop-down is closed) or in fill-in the focus movement behavior is dependent on the SESSION:DATA-ENTRY-RETURN attribute. If the attribute is TRUE - the focus is moving to the next widget, if FALSE - focus does not change. The default value of the attribute according 4GL docs is TRUE for ChUI and FALSE for GUI. However like other documented features - this must be tested because 4GL documentation is far away from ideal.

Thank you Eugenie. According to my tests in this case the documentation is accurate.

#69 - 03/03/2016 02:05 PM - Igor Skornyakov

It seems that `Frame.isLastWidget()` method is not exactly correct.
For the the frame definition

```
DEF VAR f1 AS CHARACTER FORMAT "X(10)" VIEW-AS FILL-IN BGCOLOR 3 LABEL "f1".
DEF VAR g1 AS CHARACTER FORMAT "X(10)" VIEW-AS FILL-IN BGCOLOR 3 LABEL "g1".
DEF VAR cb1 AS CHARACTER VIEW-AS COMBO-BOX SIMPLE
  SIZE 20 BY 4
  LIST-ITEM-PAIRS "L1", "V1", "L2", "V2", "L3", "V3", "L4", "V4", "L5", "V5".
DEF VAR cd1 AS CHARACTER VIEW-AS COMBO-BOX DROP-DOWN
  SIZE 20 BY 1
  LIST-ITEM-PAIRS "L1", "V1", "L2", "V2", "L3", "V3", "L4", "V4", "L5", "V5".
DEF VAR h1 AS CHARACTER FORMAT "X(10)" VIEW-AS FILL-IN BGCOLOR 3 LABEL "h1".
DEF button b1 LABEL "b1".
DEF FRAME fr1 cd1 f1 g1 cb1 b1 h1 WITH TITLE "Frame1" SIDE-LABELS.
```

The check for h1 returns false as the method considers as last widget `ScrollPaneGuilImpl`. I will add test for "non-virtual". Hopefully it is safe.

#70 - 03/03/2016 04:49 PM - Igor Skornyakov

Implemented runtime support for the `SESSION:DATA-ENTRY-RETURN` attribute.

Committed to the task branch 1612a revno 10981.

#71 - 03/04/2016 09:50 AM - Igor Skornyakov

There are still number of details regarding `IMMEDIATE-DISPLAY` to be checked but one problem is clear right now.

It seems that there is no problem with Swing GUI. Roughly speaking it is sufficient to check the value of the `IMMEDIATE-DISPLAY` before the flush operation and do not flush if it is on. However with Web GUI we keep pending updates in the list and this can cause a memory overflow. The simplest solution is to have a configurable limit for the `pendingBytes` and ignore the `IMMEDIATE-DISPLAY` value if `pendingBytes` exceeds the threshold. After all the `IMMEDIATE-DISPLAY` is just about an optimization. Is this approach acceptable?

Thank you.

#72 - 03/04/2016 09:52 AM - Greg Shah

Before I can evaluate your proposal, I need to understand what exactly `IMMEDIATE-DISPLAY` does. Please provide a description, as best as you know it so far.

#73 - 03/04/2016 09:53 AM - Hynek Cihlar

Igor Skornyakov wrote:

It seems that `Frame.isLastWidget()` method is not exactly correct.
For the the frame definition
[...]

The check for `h1` returns `false` as the method considers as last widget `ScrollPaneGuiImpl`. I will add test for "non-virtual". Hopefully it is safe.

The problematic code is in `ComboBoxGuiImpl.createSimpleList()`, for some reason the method adds its list to frame widgets.

```
Frame<GuiOutputManager> frCombo = (Frame<GuiOutputManager>) UiUtils.locateFrame(this);  
frCombo.addWidget(simpleListPane);
```

I don't think the list should be added among frame widgets in the first place.

Eugenie, please comment.

#74 - 03/04/2016 10:03 AM - Igor Skornyakov

Greg Shah wrote:

Before I can evaluate your proposal, I need to understand what exactly IMMEDIATE-DISPLAY does. Please provide a description, as best as you know it so far.

Section 59 has been updated.

#75 - 03/04/2016 10:04 AM - Constantin Asofiei

Hynek Cihlar wrote:

The problematic code is in `ComboBoxGuiImpl.createSimpleList()`, for some reason the method adds its list to frame widgets.

This is not the only pseudo-widget (related to COMBO-BOX) added directly into a frame... `EntryFieldGuiImpl` is added also.

#76 - 03/04/2016 10:13 AM - Igor Skornyakov

Constantin Asofiei wrote:

Hynek Cihlar wrote:

The problematic code is in `ComboBoxGuiImpl.createSimpleList()`, for some reason the method adds its list to frame widgets.

This is not the only pseudo-widget (related to COMBO-BOX) added directly into a frame... `EntryFieldGuiImpl` is added also.

This `isLastWidget` method is currently invoked for the FILL-IN only. I've added a check for non-virtual and it seem working OK.

#77 - 03/04/2016 10:15 AM - Greg Shah

The semantics of the DATA-ENTRY-RETURN seems to be as described in the Progress documentation (see `uast/session/data-entry-return.p` - tested both in ChUI and GUI mode)

I'm glad it matches the docs. Please summarize it here anyway so that future readers don't have to look somewhere else for the details.

In a simplest scenario when the IMMEDIATE-DISPLAY is on all output operation are on hold until it is off or a statement block for input (including PAUSE) operation. Moreover all pending updates are blocked including those ones initiated before IMMEDIATE-DISPLAY was set to on.

Questions:

1. What does "all output operation" mean?
2. What does "pending updates" mean?
3. You mentioned this is an optimization. Are you saying that there is no behavior difference caused by this attribute that can be observed by a user?

#78 - 03/04/2016 10:17 AM - Eugenie Lyzenko

Eugenie, please comment.

This is mandatory with the current working model(like for EntryFieldGuilmpl). These are pseudo widgets(ID is negative) so not a problem to differentiate them from other real widgets in a frame. The combo-box is a complex widget that can have inside additional pseudo-widgets tightly coupled and working with each other.

Do not change this code unless you are completely understanding what you are going to do and what additional tasks you will have to do to verify there are no GUI regressions. As general consideration - do not fix the things that is working.

#79 - 03/04/2016 10:27 AM - Igor Skornyakov

Greg Shah wrote:

Questions:

1. What does "all output operation" mean?
2. What does "pending updates" mean?

Note 59 has been updated.

3. You mentioned this is an optimization. Are you saying that there is no behavior difference caused by this attribute that can be observed by a user?

Well, it depends. When the the IMMEDIATE-DISPLAY is on for a short time then a user may not notice any difference. Otherwise it will not see any changes for some time and eventually will see just a cumulative result.

#80 - 03/04/2016 10:39 AM - Hynek Cihlar

Eugenie Lyzenko wrote:

Eugenie, please comment.

This is mandatory with the current working model (like for EntryFieldGuiImpl). These are pseudo widgets (ID is negative) so not a problem to differentiate them from other real widgets in a frame. The combo-box is a complex widget that can have inside additional pseudo-widgets tightly coupled and working with each other.

Do not change this code unless you are completely understanding what you are going to do and what additional tasks you will have to do to verify there are no GUI regressions. As general consideration - do not fix the things that is working.

I see, this is the problematic widget structure covered by [#2832](#).

#81 - 03/04/2016 12:42 PM - Greg Shah

Your description of IMMEDIATE-DISPLAY is the opposite of how it is described by Progress:

If TRUE, the AVM updates the display for every I/O operation, including DISPLAY statements. If FALSE, the AVM does not update the display until a statement blocks for input, such as an UPDATE statement. FALSE is the default setting. A TRUE setting provides more accurate screen displays during long display loops at the price of slower performance.

In Windows, this attribute has a similar effect on interactive I/O to setting the MULTITASKING-INTERVAL attribute to a low non-zero value, providing a more frequent display refresh. This attribute also provides the same functionality as the ImmediateDisplay parameter in the current environment (which might be the Registry (Windows only) or an initialization file).

Is your statement correct (that TRUE means that pushing the updates to the screen is DISABLED)?

Other questions:

- Does this affect only the screen or does it also affect stream I/O?
- Can you move, resize, minimize, maximize 4GL windows while this is active?

Also, based on this discussion this is not just an optimization, since having it on or off can change what the user will see (at least when the user will see something). Since this can be changed at any time, it has an actual visible affect on the UI.

#82 - 03/04/2016 01:07 PM - Igor Skornyakov

Greg Shah wrote:

Your description of IMMEDIATE-DISPLAY is the opposite of how it is described by Progress:

Sorry, of course

[...]

Is your statement correct (that TRUE means that pushing the updates to the screen is DISABLED)?

Sorry, of course I was wrong, fixed.

Other questions:

- Does this affect only the screen or does it also affect stream I/O?
- Can you move, resize, minimize, maximize 4GL windows while this is active?

I haven't checked this yet. Added to the list of open questions.

Also, based on this discussion this is not just an optimization, since having it on or off can change what the user will see (at least when the user will see something). Since this can be changed at any time, it has an actual visible affect on the UI.

Yes, it has the visual effect but the reason for this feature is the optimization. In any case we have to avoid the risk of the memory overflow. What I have suggested is the easiest way and is "eventually compatible" with the Progress behaviour. If it is not acceptable I will think about other solutions.

#83 - 03/04/2016 01:49 PM - Greg Shah

It seems that there is no problem with Swing GUI. Roughly speaking it is sufficient to check the value of the IMMEDIATE-DISPLAY before the flush operation and do not flush if it is on. However with Web GUI we keep pending updates in the list and this can cause a memory overflow. The simplest solution is to have a configurable limit for the pendingBytes and ignore the IMMEDIATE-DISPLAY value if pendingBytes exceeds the threshold. After all the IMMEDIATE-DISPLAY is just about an optimization. Is this approach acceptable?

Actually, I think we can exactly duplicate the behavior with no memory overflows if we just honor IMMEDIATE-DISPLAY on the JS side. The following code at the end of the Window.prototype.draw function in p2j.screen.js is the code that pushes the accumulated drawing to the screen:

```
// to blast the offscreen image on the window canvas
this.ctx.drawImage(this.offscreenCanvas, 0, 0);
```

If that is bypassed, then the screen won't be updated.

Of course, this means that the JS code needs a notification when IMMEDIATE-DISPLAY is on or off. That notification needs to occur even for the implicit "on" that occurs when shifting into edit mode. It seems like both GUI drivers can have their different implementation so long as they get the notifications properly. Do NOT "up call" from the driver to the layer above. The notifications should be down-calls into the driver.

Constantin/Sergey: any thoughts or comments?

Another question to add to the list: if IMMEDIATE-DISPLAY is false and then you go into editing mode (e.g. UPDATE statement), when you exit editing mode is IMMEDIATE-DISPLAY still false? And if so, is it back in force again?

#84 - 03/04/2016 01:59 PM - Igor Skornyakov

Greg Shah wrote:

It seems that there is no problem with Swing GUI. Roughly speaking it is sufficient to check the value of the IMMEDIATE-DISPLAY before the flush > Actually, I think we can exactly duplicate the behavior with no memory overflows if we just honor IMMEDIATE-DISPLAY on the JS side. The following code at the end of the Window.prototype.draw function in p2j.screen.js is the code that pushes the accumulated drawing to the screen:

[...]

If that is bypassed, then the screen won't be updated.

Of course, this means that the JS code needs a notification when IMMEDIATE-DISPLAY is on or off. That notification needs to occur even for the implicit "on" that occurs when shifting into edit mode. It seems like both GUI drivers can have their different implementation so long as they get the notifications properly. Do NOT "up call" from the driver to the layer above. The notifications should be down-calls into the driver.

This looks to be a solution. Thank you.

Another question to add to the list: if IMMEDIATE-DISPLAY is false and then you go into editing mode (e.g. UPDATE statement), when you exit editing mode is IMMEDIATE-DISPLAY still false? And if so, is it back in force again?

As soon as an input is expected (even on PAUSE statement) the pending outputs are flushed out but the attribute value is not changed.

#85 - 03/04/2016 05:00 PM - Igor Skornyakov

Greg Shah wrote:

Your description of IMMEDIATE-DISPLAY is the opposite of how it is described by Progress:

[...]

Is your statement correct (that TRUE means that pushing the updates to the screen is DISABLED)?

Strangely enough my initial description **was** correct. The default value IMMEDIATE-DISPLAY (false) means immediate propagation of the changes while the true value means putting them on hold.

Moreover. When I tested on a heavy loaded windev01 (as it typically is after 22:00 Moscow time) it looks that sometimes (sporadically) the updates are propagated immediately even IMMEDIATE-DISPLAY is true. I've never seen that when windev01 is not loaded. Looks like Progress bug.

#86 - 03/04/2016 05:10 PM - Greg Shah

What a surprise, you've found another Progress documentation failure.

Please note that Eugenie found exactly the same thing recently with NO-CURRENT-VALUE ([#2612-52](#)).

Moreover. When I tested on a heavy loaded windev01 (as it typically is after 22:00 Moscow time) it looks that sometimes (sporadically) the updates are propagated immediately even IMMEDIATE-DISPLAY is true. I've never seen that when windev01 is not loaded. Looks like Progress bug.

This part is disturbing. I wonder if there is some interaction with MULTITASKING-INTERVAL that will automatically set a minimum time period for a UI refresh.

OK, at this point I think we need to postpone the implementation of IMMEDIATE-DISPLAY. Please create a new task and link to the details you have found in this task. Don't do any further implementation.

#87 - 03/04/2016 05:15 PM - Igor Skornyakov

Greg Shah wrote:

What a surprise, you've found another Progress documentation failure.

Please note that Eugenie found exactly the same thing recently with NO-CURRENT-VALUE ([#2612-52](#)).

Moreover. When I tested on a heavy loaded windev01 (as it typically is after 22:00 Moscow time) it looks that sometimes (sporadically) the updates are propagated immediately even IMMEDIATE-DISPLAY is true. I've never seen that when windev01 is not loaded. Looks like Progress bug.

This part is disturbing. I wonder if there is some interaction with MULTITASKING-INTERVAL that will automatically set a minimum time period for a UI refresh.

OK, at this point I think we need to postpone the implementation of IMMEDIATE-DISPLAY. Please create a new task and link to the details you have found in this task. Don't do any further implementation.

OK. I will retest once again at the weekend and will create a task if my today's findings are correct.

#88 - 03/06/2016 02:37 PM - Igor Skornyakov

It is strange but today I do not see any effect of the IMMEDIATE-DISPLAY at all.

#89 - 03/06/2016 03:50 PM - Igor Skornyakov

- File id-0-true-false.mkv added

- File id-9999-true-false.mkv added

- File id-0-false-true.mkv added

- File *id-9999-false-true.mkv* added

- File *id.p* added

There is a correlation between IMMEDIATE-DISPLAY and MULTITASKING-INTERVAL. The attached test runs the same procedure twice - one with IMMEDIATE-DISPLAY = TRUE and another one with IMMEDIATE-DISPLAY = FALSE. The attached screencasts shows the results with the MULTITASKING-INTERVAL = 0 and 9999. Other positive values of the MULTITASKING-INTERVAL do not affect the appearance (except the obvious impact on the performance and the rate of refresh of the topmost field.

#90 - 03/07/2016 11:56 AM - Igor Skornyakov

- File *debug-alert.png* added

The note 59 has been updated.

#91 - 03/07/2016 01:37 PM - Igor Skornyakov

I cannot reproduce the effect of the IMMEDIATE-DISPLAY I've seen initially. I'm really start to think that it was a kind of hallucination. Unfortunately I started to make screencasts only yesterday. The only thing I can see is a delayed output in the message area (in case of IMMEDIATE-DISPLAY == false).

I will test other scenarios.

Regarding DEBUG-ALERT. It is possible of course to provide a stack trace of the server-side code, but it will be related to the Progress stack very indirectly. Of course it doesn't look feasible (and desirable) to invoke debugger.

BTW: Do we plan to add support for *-debugalert* and *-clientlog* *stattup* arguments?

Thank you.

#92 - 03/08/2016 02:31 PM - Igor Skornyakov

- File *immediate-display.mkv* added

I was able to see an effect of the IMMEDIATE-DISPLAY attribute (at least partially). Note 59 has been updated. See attached screencast.

The current P2J behaviour looks like a Progress one with MULTITASKING-INTERVAL == 1 (or other small positive number) - in this situation I've not noticed any effect of the IMMEDIATE-DISPLAY as mentioned in the note 59. In any case any changes regarding IMMEDIATE-DISPLAY/MULTITASKING-INTERVAL will change the existing visual behaviour (even in situations when IMMEDIATE-DISPLAY was not assigned). It is also necessary to add support for the MULTITASKING-INTERVAL at the same time.

Please note also that P2J behaviour of the converted *uast/session/immediate-display.p* is not exactly the same as with ABL - the frame *p1* is drawn a little bit differently.

#93 - 03/08/2016 02:34 PM - Greg Shah

I appreciate the updates on IMMEDIATE-DISPLAY. As I mentioned in note 86:

OK, at this point I think we need to postpone the implementation of IMMEDIATE-DISPLAY. Please create a new task and link to the details you have found in this task. Don't do any further implementation.

Create the task and link it here. Don't do anything further on IMMEDIATE-DISPLAY.

#94 - 03/08/2016 02:44 PM - Igor Skornyakov

Greg Shah wrote:

I appreciate the updates on IMMEDIATE-DISPLAY. As I mentioned in note 86:

OK, at this point I think we need to postpone the implementation of IMMEDIATE-DISPLAY. Please create a new task and link to the details you have found in this task. Don't do any further implementation.

Create the task and link it here. Don't do anything further on IMMEDIATE-DISPLAY.

Done. Task [#3020](#). What about DEBUG-ALERT (see note 59)?
Thank you.

#95 - 03/08/2016 03:21 PM - Greg Shah

Regarding DEBUG-ALERT. It is possible of course to provide a stack trace of the server-side code, but it will be related to the Progress stack very indirectly.

We already have an approach that provides the 4GL stack trace on the server side, so we could reuse that. See the `EnvironmentOps.getSourceName()` which replaces the `PROGRAM-NAME()` 4GL function. It uses `ProcedureManager.getStackEntry()`.

It will exactly duplicate the Progress procedure stack, but the program names will be relative and you won't see the Progress compiler's .ped files.

Of course it doesn't look feasible (and desirable) to invoke debugger.

Agreed.

BTW: Do we plan to add support for `-debugalert` and `-clientlog` startup arguments?

The customer does use `-debugalert`. The .pf I have does not use `-clientlog` but I will check with them.

Do not do any further development on DEBUG-ALERT until I find out how important it is.

By the way, in note 59 both DEBUG-ALERT scenarios talk about when the `-clientlog` is **not** specified. I think one of them is supposed to be the opposite case.

#96 - 03/08/2016 03:35 PM - Igor Skornyakov

Greg Shah wrote:

The customer does use -debugalert. The .pf I have does not use -clientlog but I will check with them.

BTW: in my test -debugalert changes the value of the DEBUG-ALERT to true but the error message is displayed as if it is false. See updated note 59.

Do not do any further development on DEBUG-ALERT until I find out how important it is.

OK. What should I work on tomorrow?

By the way, in note 59 both DEBUG-ALERT scenarios talk about when the -clientlog is **not** specified. I think one of them is supposed to be the opposite case.

Sorry, it was a misprint. Fixed.

#97 - 03/08/2016 04:04 PM - Greg Shah

OK. What should I work on tomorrow?

Is DATA-ENTRY-RETURN done?

Is APPL-ALERT-BOXES done?

Finish those please.

#98 - 03/08/2016 04:07 PM - Igor Skornyakov

Greg Shah wrote:

OK. What should I work on tomorrow?

Is DATA-ENTRY-RETURN done?

Is APPL-ALERT-BOXES done?

Yes, I hope that both are done. But I will perform more thorough testing.

#99 - 03/08/2016 04:17 PM - Igor Skornyakov

Rebased task branch 1612a from P2J trunk rev 10980. The latest revision is now 10983.

#100 - 03/08/2016 04:21 PM - Greg Shah

Code Review Task Branch 1612a Revision 10983

1. Are you sure that only the message(Object[] list, ColorSpec cs, handle hWin) needs to support APPL-ALERT-BOXES? That would be surprising.
2. I don't like hard coding the check of frame.config().defaultButtonId -1 in Fillin.getDataEntryReturn(). This means that part of the frame's implementation of default buttons is now exposed inside a contained widget. That is too fragile of an approach. It would be better to ask the frame to calculate the getDataEntryReturn() by moving the entire ThinClient.isDataEntryReturn() && frame.config().defaultButtonId -1 expression into Frame.
3. LogicalTerminal should not have to import these:

```
import static com.goldencode.p2j.ui.LogicalTerminal.ALERT_MESSAGE;  
import static com.goldencode.p2j.ui.LogicalTerminal.BTN_OK;
```

#101 - 03/08/2016 04:26 PM - Igor Skornyakov

Greg Shah wrote:

Code Review Task Branch 1612a Revision 10983

1. Are you sure that only the message(Object[] list, ColorSpec cs, handle hWin) needs to support APPL-ALERT-BOXES? That would be surprising.

I think so. All other message method either delegate to this one or are used for MESSAGE with SET/UPDATE clause which are not affected by APPL-ALERT-BOXES according to my tests. But I will double check.

2. I don't like hard coding the check of `frame.config().defaultButtonId -1` in `Fillin.getDataEntryReturn()`. This means that part of the frame's implementation of default buttons is now exposed inside a contained widget. That is too fragile of an approach. It would be better to ask the frame to calculate the `getDataEntryReturn()` by moving the entire `ThinClient.isDataEntryReturn() && frame.config().defaultButtonId -1` expression into `Frame`.

3. `LogicalTerminal` should not have to import these:

[...]

Thank you, I will fix it.

#102 - 03/09/2016 06:07 PM - Igor Skornyakov

My initial description of the DEBEG-ALERT semantics was not correct as I started me tests from the Procedure Editor. I will update the description tomorrow.

There is a number of attributes which are related to `-clientlog` startup parameter (`LOGFILE-NAME`, `LOGGING-LEVEL`,). Should I analyze them in the scope of this task? The same is about the `-logginglevel` and other logging-related startup parameters.
Thank you.

#103 - 03/10/2016 02:15 PM - Igor Skornyakov

The note 59 has been updated.

#104 - 03/11/2016 08:28 AM - Greg Shah

There is a number of attributes which are related to `-clientlog` startup parameter (`LOGFILE-NAME`, `LOGGING-LEVEL`,). Should I analyze them in the scope of this task? The same is about the `-logginglevel` and other logging-related startup parameters.

No. Those other attributes are not in use and we aren't going to support them at this time.

I have not investigate the details on `-clientlog`. Can you please summarize it here? Then we will make a call on how much or how little support is put in for it at this time.

#105 - 03/11/2016 08:36 AM - Igor Skornyakov

Greg Shah wrote:

No. Those other attributes are not in use and we aren't going to support them at this time.

I see. Thank you.

I have not investigate the details on -clientlog. Can you please summarize it here? Then we will make a call on how much or how little support is put in for it at this time.

I've investigated only those aspects of the -clientlog which are directly related to the DEBUG-ALERT support. I suggest to implement a minimal support of client logging (not supporting LOG-MANAGER:ENTRY-TYPES-LIST, LOG-MANAGER:LOG-ENTRY-TYPES, LOG-MANAGER:LOG-THRESHOLD and LOG-MANAGER:NUM-LOG-FILES but with support of LOG-MANAGER:LOGFILE-NAME and LOG-MANAGER:LOGGING-LEVEL as it is simple).

BTW: I understand that -clientlog, -logginglevel and -debugalert should be command-line options of the client. Is that correct?

#106 - 03/11/2016 08:50 AM - Greg Shah

What is the purpose of -clientlog? I guess it sets a log file name. But for what use?

I'm not sure that we need to support it. Please provide more details.

#107 - 03/11/2016 08:56 AM - Igor Skornyakov

Greg Shah wrote:

What is the purpose of -clientlog? I guess it sets a log file name. But for what use?

I'm not sure that we need to support it. Please provide more details.

What I see in my tests this log contains output of error messages and MESSAGE statements with stack trace if DEBUG-ALERT is true. See note 59 for a sample content.

#108 - 03/11/2016 08:57 AM - Greg Shah

What is the purpose of -clientlog? I guess it sets a log file name. But for what use?
I'm not sure that we need to support it. Please provide more details.

What I see in my tests this log contains output of error messages and MESSAGE statements with stack trace if DEBUG-ALERT is true. See note 59 for a sample content.

What do the Progress docs state about it?

#109 - 03/11/2016 08:57 AM - Greg Shah

BTW: I understand that -clientlog, -logginglabel and -debugalert should be command-line options of the client. Is that correct?

Maybe. But I'm not sure yet.

If we do have to do this, then we would add support as bootstrap config overrides on the command line. For example, the same mechanism is used to pass client:cmd-line-option:startup-procedure. See CommonDriver.processOverrides() and ClientCore.processClientParams() for how we set those values into a ClientParameters instance. That instance is passed to StandardServer.standardEntry() and from their the values are used on the server side.

#110 - 03/11/2016 09:04 AM - Igor Skornyakov

Greg Shah wrote:

What do the Progress docs state about it?

Essentially not much more than I could see. It is essentially just a logging system with a number of additional parameters to support filtering if the logged events and log files rolling over. I'm not familiar yet with batch/appserver modes so I can miss something.

Client Logging (-clientlog)

Use Client Logging (-clientlog) to allow an application to automatically write error and warning messages to the specified log file.

The pathname and filename of the log file OpenEdge uses for messages, ABL stack trace, and .NET stack trace information.

If the filename you supply is a relative pathname, then a file is accessed relative to the current working directory. If the filename is an absolute pathname, then the specified file is accessed.

Note:

Do not include a numbered sequence in the filename. This might conflict with the rolled over log files OpenEdge creates based on your Number of Log Files to Keep (-numlogfiles) and Log Threshold (-logthreshold) startup parameter settings.

Use the Log Entry Types (-logentrytypes) startup parameter to specify one or more types of log entries you want to write to the log file. Use the Logging Level (-logginglevel) startup parameter to specify the level at which log entries are written to the log file.

When you use the -clientlog startup parameter, and you also specify the Debug Alert (-debugalert) startup parameter or set SESSION:DEBUG-ALERT to TRUE, the log file includes an ABL stack trace for error messages (ABL errors and .NET Exceptions) and Alert-box messages. The top of the stack (most recent call) is displayed at the top of the trace listing.

.NET Exceptions can be thrown when working with .NET objects in ABL. When you do not handle these Exceptions—there is no CATCH or NO-ERROR logic present—and Debug Alert is on, the AVM adds the .NET stack trace to the Debug Alert information after the ABL call stack. The ABL stack trace and the .NET stack trace are added both in the Debug Alert Help dialog box and in the client log (when -clientlog is specified).

If an error message is diverted to the ERROR-STATUS system handle, and client logging is enabled, then no information is written to the log file.

In a non-interactive session, the application is configured so that the output device is associated with a file (or another device). In this configuration, when an ABL statement encounters an error, it writes the error to the output device. If client logging is enabled, then this message is also written to the specified log file. Errors are written to the output device at logging level 1 (Error) and up.

You can use the MESSAGE statement with the VIEW-AS ALERT-BOX option to write application specific information to the screen and the log file. In this case, you must specify an entry type of "4GLMessages" and a logging level of 2 (Basic), at least.

Note:

When you specify a non-zero value for the Log Threshold (-logthreshold) startup parameter, only one client process at a time can open the log file. Therefore, consider specifying a different log file for each client session.

For more information about logging levels, see the Logging Level (-logginglevel) startup parameter reference entry. For more information about specifying log entry types, see the Log Entry Types (-logentrytypes) startup parameter reference entry.

You can also use attributes on the LOG-MANAGER system handle to specify log entry types and logging levels.

For more information about the ABL elements previously referenced above, see OpenEdge Development: ABL Reference.

For more detailed information about enabling logging, see OpenEdge Development: Debugging and Troubleshooting.

#111 - 03/11/2016 09:21 AM - Greg Shah

We are not going to support -clientlog at this time. Make sure to put an `UnimplementedFeature.todo()` into the debug-alert code to highlight that this is not yet done.

At this time, you should only implement features that don't depend on -clientlog. This would include:

- the dialog changes that allow display of the stack trace
- the output of the same details to the appserver logs

Constantin can share more details on the appserver logging.

The customer stated this:

we tend to use -debugalert in GUI - whilst developing and when debugging, not to log the stack with -clientlog, but because it also has the side-affect of adding the Help button to all ALERT-BOXES, which then enables you to look at the stack. I guess you could say Progress are being sloppy - enabling 2 things with 1 option.

I think if -debugalert is set for the appserver session, you also get the stack written to the appserver log file (on any untrapped error), regardless of whether -clientlog is specified.

see <http://knowledgebase.progress.com/articles/Article/19455>

...

As mentioned before, if you have -debugalert for the appserver session, it will write the stack to the appserver log file if an untrapped error is produced.—

#112 - 03/11/2016 09:23 AM - Igor Skornyakov

Greg Shah wrote:

We are not going to support -clientlog at this time. Make sure to put an `UnimplementedFeature.todo()` into the debug-alert code to highlight that this is not yet done.

At this time, you should only implement features that don't depend on -clientlog. This would include:

- the dialog changes that allow display of the stack trace
- the output of the same details to the appserver logs

Constantin can share more details on the appserver logging.

I see. Thank you.

#113 - 03/11/2016 09:24 AM - Greg Shah

In the dialog the Debug button should be present but disabled so that it is clear that it is not functional. The customer has accepted that we are not duplicating the Progress debugger. They also accept that our stack traces will not have line numbers or the line numbers will be placeholders (since we have no cross-reference for the 4GL code).

#114 - 03/14/2016 04:37 AM - Igor Skornyakov

Rebased task branch 1612a from P2J trunk rev 10982. The latest revision is now 10986.

#115 - 03/15/2016 05:00 AM - Constantin Asofiei

Greg Shah wrote:

At this time, you should only implement features that don't depend on -clientlog. This would include:

...

- the output of the same details to the appserver logs

Constantin can share more details on the appserver logging.

Igor, to configure an appserver, on windev01 you need to use the Progress Explorer tool (there are also details in the handbook_20160303.zip about how to setup and use an appserver connection). In the appserver properties, passing -debugalert is done in the Agent/General/Server startup parameters. After this, create a remote program (on windev01) where you have some code which raises an ERROR condition, and run it remotely. The stacktrace should be logged in the file specified in the Agent/Logging settings/Server log filename setting.

In P2J, the logging file name is specified in the clientConfig/outputToFile node.

#116 - 03/15/2016 05:02 AM - Igor Skornyakov

Constantin Asofiei wrote:

Igor, to configure an appserver, on windev01 you need to use the Progress Explorer tool (there are also details in the handbook_20160303.zip about how to setup and use an appserver connection). In the appserver properties, passing -debugalert is done in the Agent/General/Server startup parameters. After this, create a remote program (on windev01) where you have some code which raises an ERROR condition, and run it remotely. The stacktrace should be logged in the file specified in the Agent/Logging settings/Server log filename setting.

In P2J, the logging file name is specified in the clientConfig/outputToFile node.

Thank you Constantin! I plan to finish with UI part today and start working on the logging part.

#117 - 03/17/2016 09:15 AM - Igor Skornyakov

We need a new widget type to display the Progress stack trace as described in the note 59 section 5. I suggest to implement a composite widget containing title, two buttons - OK and Debug (the later is disabled) and a scrollable area for the stack itself. At this moment the best candidate seems to be (a subclass of ?) the EDITOR which is read-only. I also suggest to refactor `AlertBox(Gui)Impl` to reuse buttons' processing and inherit the new dialog from it. However as the `AlertBox(Gui)Impl` code looks tricky (and looks very different for ChUI and GUI mode) it may be more practical to implement a new widget from the scratch using the similar approach.

Hynek, what do you think?
Thank you.

#118 - 03/17/2016 10:14 AM - Hynek Cihlar

Igor Skornyakov wrote:

We need a new widget type to display the Progress stack trace as described in the note 59 section 5. I suggest to implement a composite widget containing title, two buttons - OK and Debug (the later is disabled) and a scrollable area for the stack itself. At this moment the best candidate seems to be (a subclass of ?) the EDITOR which is read-only. I also suggest to refactor `AlertBox(Gui)Impl` to reuse buttons' processing and inherit the new dialog from it. However as the `AlertBox(Gui)Impl` code looks tricky (and looks very different for ChUI and GUI mode) it may be more practical to implement a new widget from the scratch using the similar approach.

I think you can use the same approach as with the update dialog in `Window.tinyInput()`, i.e. the widgets are built from their respective configs with help of window registry. The advantage of this solution is that it will work with relatively little code for both GUI and ChUI.

The new class you propose is ok, too I think. But the disadvantage is that you will have to create more code to handle both ChUI and GUI.

#119 - 03/17/2016 10:17 AM - Igor Skornyakov

Hynek Cihlar wrote:

I think you can use the same approach as with the update dialog in `Window.tinyInput()`, i.e. the widgets are built from their respective configs with help of window registry. The advantage of this solution is that it will work with relatively little code for both GUI and ChUI.

The new class you propose is ok, too I think. But the disadvantage is that you will have to create more code to handle both ChUI and GUI.

Thank you Hynek. I will take a look at the `Window.tinyInput()`.

#120 - 03/21/2016 03:03 PM - Igor Skornyakov

Hynek,
I'm working now on the dialog for the stack trace following the approach you suggested. The dialog should contain the widget for the stack trace (the read-only editor) and two buttons. At this time I'm struggling with Swing GUI version. I can see editor (albeit it doesn't look exactly as in 4GL, but can be an EditorGuiImpl issue). However I still have two problems:

1. The buttons should be aligned to the centre of the line. I understand that it can be achieved with the auxiliary container (like ButtonsPanel in the AlertBoxGuiImpl). However may be there is a simpler solution?
2. There should be spaces at the top, left and right of the editor. What is the best way to achieve this?

Thank you.

#121 - 03/21/2016 04:59 PM - Hynek Cihlar

Igor Skornyakov wrote:

Hynek,
I'm working now on the dialog for the stack trace following the approach you suggested. The dialog should contain the widget for the stack trace (the read-only editor) and two buttons. At this time I'm struggling with Swing GUI version. I can see editor (albeit it doesn't look exactly as in 4GL, but can be an EditorGuiImpl issue). However I still have two problems:

1. The buttons should be aligned to the centre of the line. I understand that it can be achieved with the auxiliary container (like ButtonsPanel in the AlertBoxGuiImpl). However may be there is a simpler solution?

Why don't you like the ButtonsPanel way :-)? This brings an extra level of decomposition to the UI design with some benefits - better widgets management, layout, etc.

2. There should be spaces at the top, left and right of the editor. What is the best way to achieve this?

IMHO the easiest is to add a root panel widget to the window, give it some nonzero location and set its width accordingly. Then add all the widgets to the root panel. The UI design will become well articulate with clear intention.

#122 - 03/21/2016 05:12 PM - Igor Skornyakov

Hynek Cihlar wrote:

Igor Skornyakov wrote:

Hynek,
I'm working now on the dialog for the stack trace following the approach you suggested. The dialog should contain the widget for the stack

trace (the read-only editor) and two buttons. At this time I'm struggling with Swing GUI version. I can see editor (albeit it doesn't look exactly as in 4GL, but can be an EditorGuiImpl issue). However I still have two problems:

1. The buttons should be aligned to the centre of the line. I understand that it can be achieved with the auxiliary container (like ButtonsPanel in the AlertBoxGuiImpl). However may be there is a simpler solution?

Why don't you like the buttonsPanel way :-)? This brings an extra level of decomposition to the UI design with some benefits - better widgets management, layout, etc.

I cannot say that I do not "like" this way. I was just thinking that may be there is a simpler way.

2. There should be spaces at the top, left and right of the editor. What is the best way to achieve this?

IMHO the easiest is to add a root panel widget to the window, give it some nonzero location and set its width accordingly. Then add all the widgets to the root panel. The UI design will become well articulate with clear intention.

Thank you Hynek. I will think about this. I wanted to avoid the position/dimension calculation which seems to depend on the type of the UI. For example to mimic a statement like this:

```
DEF FRAME fr SKIP(1) SPACE(1) editor SPACE(1) SKIP btnOk btnDebug ....
```

Unfortunately while SKIP(1) adds a space at the top the SPACE(1) has no effect (at least in GUI).

#123 - 03/21/2016 05:41 PM - Hynek Cihlar

Igor Skornyakov wrote:

Hynek Cihlar wrote:

Igor Skornyakov wrote:

Hynek,

I'm working now on the dialog for the stack trace following the approach you suggested. The dialog should contain the widget for the stack trace (the read-only editor) and two buttons. At this time I'm struggling with Swing GUI version. I can see editor (albeit it doesn't look exactly as in 4GL, but can be an EditorGuiImpl issue). However I still have two problems:

1. The buttons should be aligned to the centre of the line. I understand that it can be achieved with the auxiliary container (like ButtonsPanel in the AlertBoxGuiImpl). However maybe there is a simpler solution?

Why don't you like the buttonsPanel way :-)? This brings an extra level of decomposition to the UI design with some benefits - better widgets management, layout, etc.

I cannot say that I do not "like" this way. I was just thinking that maybe there is a simpler way.

2. There should be spaces at the top, left and right of the editor. What is the best way to achieve this?

IMHO the easiest is to add a root panel widget to the window, give it some nonzero location and set its width accordingly. Then add all the widgets to the root panel. The UI design will become well articulated with clear intention.

Thank you Hynek. I will think about this. I wanted to avoid the position/dimension calculation which seems to depend on the type of the UI. For example to mimic a statement like this:

[...]

Unfortunately while SKIP(1) adds a space at the top the SPACE(1) has no effect (at least in GUI).

Good idea, but I am afraid that 4GL internally uses two different layouts for ChUI and GUI anyway.

#124 - 03/21/2016 05:45 PM - Igor Skornyakov

Hynek Cihlar wrote:

Good idea, but I am afraid that 4GL internally uses two different layouts for ChUI and GUI anyway.

I also think so. Just trying to make differences in the implementations well-localized to simplify future changes and maintenance.

#125 - 03/22/2016 07:23 AM - Greg Shah

If you have some partially working code, please check it in and get help. We have to close this down as quickly as possible.

#126 - 03/22/2016 07:26 AM - Igor Skornyakov

Greg Shah wrote:

If you have some partially working code, please check it in and get help. We have to close this down as quickly as possible.

OK. I'm working now on the ButtonPanel code. What I did yesterday was committed.

#127 - 03/22/2016 07:27 AM - Greg Shah

In [#3033](#), you are trying to use extra widgets to get your layout to look right. Does the 4GL use a fixed location for the widgets or does it scale? If it is fixed, then you can set absolute positions, right?

#128 - 03/22/2016 07:30 AM - Igor Skornyakov

Greg Shah wrote:

In [#3033](#), you are trying to use extra widgets to get your layout to look right. Does the 4GL use a fixed location for the widgets or does it scale? If it is fixed, then you can set absolute positions, right?

This is correct and I will use this approach as a workaround to [#3033](#). I haven't checked however if it is possible to change e.g. font of the dialog and how it affects its appearance.

#129 - 03/22/2016 01:02 PM - Igor Skornyakov

The dialog for the Progress stack trace is essentially finished for the Swing GUI

Committed to the task branch 1612a revno 10990.

The remaining thing to do is adding spaces around the EDITOR widget and proper alignment of the buttons. Please note that scrolling with mouse doesn't work - seems to be the EDITOR issue.

Hynek, can you please take a look at the `AlertBoxCommons.showStackTrace()` method? Are there any obvious mistakes? Thank you.

#130 - 03/22/2016 05:20 PM - Hynek Cihlar

Igor Skornyakov wrote:

Hynek, can you please take a look at the `AlertBoxCommons.showStackTrace()` method? Are there any obvious mistakes?

Looks ok to me.

#131 - 03/22/2016 05:20 PM - Igor Skornyakov

Hynek Cihlar wrote:

Looks ok to me.

Great! Thank you very much indeed!

#132 - 03/23/2016 08:20 AM - Greg Shah

Code Review Task Branch 1612a Revision 10990

The overall approach is very good. My comments are mostly about coding standards.

1. In note 129 above, you say:

The dialog for the Progress stack trace is essentially finished for the Swing GUI.

As far as I can tell, this is not specific to Swing GUI. It seems like it should work for the web GUI and for ChUI too. Please help me understand if there are any limitations in using the current implementation in the web GUI or ChUI.

2. Direct users of `ThinClient.messageBox()` all have comments like TODO: is it always correct?. Please provide more details about your concern in these cases.

3. The following files are missing history entries: `ButtonConfig`, `EditorConfig`, `RectangleConfig`, `SkipConfig`

4. Please move the constructors into the proper order (public then protected) in `ButtonConfig`, `EditorConfig`, `RectangleConfig`, `SkipConfig`

5. The following files need copyright date updates: `ButtonConfig`, `EditorConfig`, `ErrorWriterInteractive`, `RectangleConfig`, `SkipConfig`

6. In `LogicalTerminal.message()` please format this:

```
public static void message(Object[] list,
```

```

        boolean    set,
        Accessor   var,
        boolean    auto,
        String     format,
        ColorSpec  cs,
        handle     hWin)
    {
        <---- remove this blank line which is against st
andards
        if (locate().isUsingAlertBoxes && var == null)
        {
            messageBox(list,
                set,
                var,
                ALERT_MESSAGE, BTN_OK,
                "Message",
                cs,
                hWin);
            return;
        }
    }

```

like this:

```

public static void message(Object[] list,
        boolean set,
        Accessor var,
        boolean auto,
        String format,
        ColorSpec cs,
        handle hWin)
    {
        if (locate().isUsingAlertBoxes && var == null)
        {
            messageBox(list, set, var, ALERT_MESSAGE, BTN_OK, "Message", cs, hWin);
            return;
        }
    }

```

By the way, I do appreciate that you left a blank line AFTER this if {} block. It aids readability.

7. In LogicalTerminal.messageBox(), this code:

```

List<String> stack = null;
if (SessionUtils.isDebugEnabled().booleanValue())
{
    stack = ProcedureManager.getStackTrace();
}

```

Please properly indent the line inside the if {} block. Also, please add a blank line after this to separate it from the next code.

8. AlertBoxImpl (ChUI code) should not import com.goldencode.p2j.ui.client.gui.driver.* (GUI code).

9. The following code is missing javadoc: AlertBoxImpl.createStackTraceWindow(), AlertBoxImpl.createGap(), AlertBox.createStackTraceWindow(), AlertBox.createGap(), AlertBoxCommons.showStackTrace(), AlertBoxCommons.createSkip(), AlertBoxGuiImpl.createStackTraceWindow(), AlertBoxGuiImpl.createGap().

10. In AlertBoxImpl.createGap(), please remove the extra blank line at the end of the method.

11. If it is unneeded, please remove the dead code in AlertBoxImpl.createStackTraceWindow(), AlertBoxCommons.createButton()

12. In ChuiWidgetFactory.createAlertBox(), this javadoc is mis-aligned:

```

* @param stack
* Progress stack trace

```

13. Make sure there is 1 blank line between `AlertBox.createStackTraceWindow()` and `AlertBox.createGap()`.

14. `AlertBoxCommons` (common code) should not import `com.goldencode.p2j.ui.client.gui.AlertBoxGuiImpl.*` (GUI code). I don't think that `java.util.concurrent.*` is needed either.

15. In `AlertBoxCommons.onAction()` you changed this:

```
    }  
    finished(rc);  
}
```

to this:

```
    }  
    finished(rc);  
}
```

Please put the blank line back as it aids readability.

16. Please add 1 blank line after `AlertBoxCommons.createSkip()`.

17. `AlertBoxGuiImpl.createGap()` has two formatting issues:

```
    return registry.getComponent(rcfg.id);  
    } <-- blank line that should not be there  
} <-- missing blank line after here  
/**
```

#133 - 03/23/2016 08:29 AM - Igor Skornyakov

Greg Shah wrote:

Code Review Task Branch 1612a Revision 10990

The overall approach is very good. My comments are mostly about coding standards.

1. In note 129 above, you say:

The dialog for the Progress stack trace is essentially finished for the Swing GUI.

As far as I can tell, this is not specific to Swing GUI. It seems like it should work for the web GUI and for ChUI too. Please help me understand if there are any limitations in using the current implementation in the web GUI or ChUI.

I also expected that it should work for ChUI as well (with minor changes like hard-coded offsets and the order in which the stack trace is displayed). However in ChUI the stack trace dialog is not visible. I'm trying to understand now why it happens.

Other issues will be fixed shortly.

#134 - 03/23/2016 06:54 PM - Igor Skornyakov

Greg Shah wrote:

2. Direct users of `ThinClient.messageBox()` all have comments like `TODO: is it always correct?`. Please provide more details about your concern in these cases.

I'm not sure how if the `DEBUG-ALERT` affects these cases.

The rest of the issues are fixed.

Committed to the task branch 1612a revo 10991.

I made some progress with ChUI stack trace dialog but it is still not working properly.

#135 - 03/23/2016 07:06 PM - Greg Shah

I made some progress with ChUI stack trace dialog but it is still not working properly.

It isn't necessary right now. Please open a separate task in the UI project for that. Make it a related task to this one, but do NOT put it into a milestone and do NOT make it a sub-task of 2677. Make sure I am a watcher.

#136 - 03/23/2016 07:15 PM - Greg Shah

Code Review Task Branch 1612a Revision 10991

1. There are 2 history entries in ButtonConfig which have wrong numbers:

```
** 022 EVL 20150417      Adding suffix constant for prepackaged disabled images.
** 008 IAS 20151214      MOUSE-POINTER attribute support.
** 008 IAS 20160322      Changed the constructor visibility.
```

2. LogicalTerminal.message() still has an extra blank line where there should not be one:

```
public static void message(Object[] list,
                           boolean set,
                           Accessor var,
                           boolean auto,
                           String format,
                           ColorSpec cs,
                           handle hWin)
{
    <---- REMOVE THIS
    if (locate().isUsingAlertBoxes && var == null)
    {
```

3. AlertBoxImpl.createGap() still has a blank line at the end where it should not:

```
public Widget<ChuiOutputManager> createGap(WidgetRegistry<ChuiOutputManager> registry)
{
    RectangleConfig rcfg = new RectangleConfig(WidgetId.nextID());
    rcfg.clientHeightChars = 1;
    rcfg.clientWidthChars = 1;
    rcfg.filled = false;
    rcfg.edgeChars = 0;
    registry.reconstructWidget(rcfg);
    return registry.getComponent(rcfg.id);
    <-- REMOVE THIS
}
```

4. Javadoc parameters should not be formatted like this:

```
* @param registry widget registry
```

But rather they should be like this:

```
* @param registry
* The widget registry.
```

Please fix this in `AlertBox.createGap()`, `AlertBoxGuiImpl.createGap()` and `AlertBoxImpl.createGap()`.

#137 - 03/24/2016 05:31 AM - Igor Skornyakov

Fixed the issues mentioned in the code review.

Rebased task branch 1612a from P2J trunk rev 10985. The latest revision is now 10995.

Regression testing started.

#138 - 03/24/2016 06:55 AM - Igor Skornyakov

Greg Shah wrote:

It isn't necessary right now. Please open a separate task in the UI project for that. Make it a related task to this one, but do NOT put it into a milestone and do NOT make it a sub-task of 2677. Make sure I am a watcher.

OK. Please note that the support for the `-debugalert` startup parameter and the logging is not implemented as well. Should I include them to the new task?

Thank you.

#139 - 03/24/2016 09:20 AM - Greg Shah

Please note that the support for the `-debugalert` startup parameter and the logging is not implemented as well. Should I include them to the new task?

No, we will do them here but in a new branch 1612b. I want to get the current changes merged to trunk ASAP.

It is really important for us to get these last changes done quickly. What time do you estimate is needed?

#140 - 03/24/2016 09:31 AM - Greg Shah

Code Review Task Branch 1612a Revision 10995

This review checks every change in this branch from trunk 10985. I have found 1 minor issue.

Please merge the history entries in `ClientExports`. There should only be 1 history entry for a branch.

```
** 115 IAS 20160302      Support for the APPL-ALERT-BOXES, DATA-ENTRY-RETURN,  
**                      and IMMEDIATE-DISPLAY SESSION attributes  
** 016 IAS 20160314      DEBUG-ALERT support.
```

You don't have to restart testing for this change.

#141 - 03/24/2016 09:50 AM - Igor Skornyakov

Greg Shah wrote:

No, we will do them here but in a new branch 1612b. I want to get the current changes merged to trunk ASAP.

It is really important for us to get these last changes done quickly. What time do you estimate is needed?

Well, the regression testing is running now. Actually the only change that can potentially cause regression is the change in the `Frame.isLastWidget()` method. But as you know the tests do not always pass at the first run even in the absence of bugs.

#142 - 03/24/2016 10:00 AM - Greg Shah

My question about the estimate is in regards to the `-debugalert` command line support and the addition of the needed logging.

Since you are setting up the customer's server environment (which includes appserver agents), you will have a good opportunity to get the logging going in that environment.

Please note that finishing this task (including the changes that will go into 1612b) is your top priority.

#143 - 03/24/2016 10:25 AM - Igor Skornyakov

Greg Shah wrote:

My question about the estimate is in regards to the `-debugalert` command line support and the addition of the needed logging.

Since you are setting up the customer's server environment (which includes appserver agents), you will have a good opportunity to get the logging going in that environment.

Please note that finishing this task (including the changes that will go into 1612b) is your top priority.

I understand. I think it will take a day or two. I will work on it at the weekend if the environment will be ready and I will have a chance to work with it tomorrow and ask questions (most like I will have ones).

#144 - 03/25/2016 04:14 AM - Igor Skornyakov

Multiple tests failed on the same (or similar) screen:

```
wait = true; millis = '180000'; failing screen =  
03/24/2016          SERVICE ORDER LABOR DATA COLLECTION          18:15:36  
confidential data redacted  
Enter data or press F4 to end.                                     Insert
```

After Enter. This can be a result of of my changes regarding DATA-ENTRY-RETURN. Investigating.

#145 - 03/25/2016 04:29 AM - Igor Skornyakov

Well in fact the template screen is

```
05/22/2009          SERVICE ORDER LABOR DATA COLLECTION          16:32:32  
confidential data redacted  
Enter data or press F4 to end.
```

So the difference is that in the test there is Insert at the bottom right. I cannot understand how my changes can cause it. There is Insert at the previous (not failed) screen as well. Continue investigation.

#146 - 03/25/2016 06:27 AM - Igor Skornyakov

It appears that the clause The AVM ignores this attribute if there is a default button on the frame means that in this case the default value of the attribute is used.

The code is fixed.

Committed to the task branch 1612a revno 10997.

Regression test restarted.

#147 - 03/25/2016 09:36 AM - Igor Skornyakov

I have a question regarding logging. I understand that I'm supposed to implement stack trace logging for the application server. If I understand the progress documentation correctly the parameters of the server log are the attributes of the DSLOG-MANAGER. However it seems to be unsupported (the conversion doesn't recognize even the DSLOG-MANAGER keyword).

What is wrong with my understanding?

Thank you.

#148 - 03/25/2016 09:46 AM - Greg Shah

If I understand the progress documentation correctly the parameters of the server log are the attributes of the DSLOG-MANAGER.

I don't understand what you mean by "the parameters of the server log". Please explain.

#149 - 03/25/2016 09:51 AM - Igor Skornyakov

Greg Shah wrote:

If I understand the progress documentation correctly the parameters of the server log are the attributes of the DSLOG-MANAGER.

I don't understand what you mean by "the parameters of the server log". Please explain.

I mean at least the name of the log (DSLOG-MANAGER:LOG-FILENAME). According to the Progress documentation DSLOG-MANAGER has the same parameters (LOG-LEVEL, NUM-LOG-FILES ...) as LOG_MANAGER.

#150 - 03/25/2016 09:56 AM - Greg Shah

OK, but the DSLOG-MANAGER system handle is not in use in the customer application. We don't need to worry about it.

The 4GL sets up logging automatically for each appserver agent. This is configured as part of the appserver configuration, but I think that even by default there is some kind of log. Essentially, it is like redirecting STDOUT and STDERR to this log for the given agent. In Progress, batch processes are also supposed to have their output redirected in a similar way. Both appserver agents and batch processes are non-interactive ChUI clients.

So, you can assume that this redirection is already in effect. You just need to know how to log to this facility. DSLOG-MANAGER has nothing to do with it.

Constantin: can you please provide some guidance on how to best write to the log?

#151 - 03/25/2016 10:14 AM - Constantin Asofiei

Greg Shah wrote:

Constantin: can you please provide some guidance on how to best write to the log?

For appserver agents (and also batch processes configured in the directory and started via the scheduler) the log is configured via the clientConfig/outputToFile entry. This file name (if configured) is passed to the spawner via the -O option. The spawner, in turn, will redirect all STDOUT output from the java process to this file. Thus, to write something to the client's log file:

1. from the server-side: you need to call a P2J Client API (add if one does not exist) which in turn will just write the message to STDOUT (and thus log it)
2. from the client-side: just write to STDOUT
3. also, writing directly to STDOUT must be done ONLY if the P2J client is a batch process or appserver Agent. Otherwise, for ChUI clients, writing to STDOUT will mean that the text will be directly shown on the terminal.

STDERR for web and batch P2J clients (this includes Appserver Agents) is automatically redirected to another file (via ClientDriver.setClientLog), with its name following the client_<OS-user-name>_<current-time-millis>.log pattern. AFAIK, 4GL can't write to STDERR - it always works with STDOUT.

Igor, a what kind of messages do you want to write to STDOUT? Can you provide a simple 4GL test for this?

#152 - 03/25/2016 10:32 AM - Igor Skornyakov

- File debug-alert.p added

Constantin Asofiei wrote:

For appserver agents (and also batch processes configured in the directory and started via the scheduler) the log is configured via the clientConfig/outputToFile entry. This file name (if configured) is passed to the spawner via the -O option. The spawner, in turn, will redirect all STDOUT output from the java process to this file. Thus, to write something to the client's log file:

1. from the server-side: you need to call a P2J Client API (add if one does not exist) which in turn will just write the message to STDOUT (and thus log it)
2. from the client-side: just write to STDOUT
3. also, writing directly to STDOUT must be done ONLY if the P2J client is a batch process or appserver Agent. Otherwise, for ChUI clients, writing to STDOUT will mean that the text will be directly shown on the terminal.

STDERR for web and batch P2J clients (this includes Appserver Agents) is automatically redirected to another file (via ClientDriver.setClientLog), with its name following the client_<OS-user-name>_<current-time-millis>.log pattern. AFAIK, 4GL can't write to STDERR - it always works with STDOUT.

Igor, a what kind of messages do you want to write to STDOUT? Can you provide a simple 4GL test for this?

Thank you Constantin.

The goal is to write Progress stack trace when DEBUG-ALERT is on and the -clientlog is specified (see note 59 section 5).

Guy Mills also wrote

BTW 2, -I think if -debugalert is set for the appserver session, you also get the stack written to the appserver log file (on any untrapped error), regardless of whether -clientlog is specified.

I have not tested it so far but if we are planning to use STDOUT (not ones governed by (DS)LOG-MANAGER such test seems to be not really needed.

The test program is attached,

#153 - 03/25/2016 10:44 AM - Constantin Asofiei

Igor Skornyakov wrote:

The goal is to write Progress stack trace when DEBUG-ALERT is on and the -clientlog is specified (see note 59 section 5).

In note 59 you state that stack trace is written also for MESSAGE statements shown as ALERT-BOX, but I can't make this work. Only stacktraces for ERROR conditions raised (which were not "consumed" via NO-ERROR) are logged.

The additional problem here is that for ChUI or GUI clients, we can't write to STDOUT, we need to maintain a separate log file.

#154 - 03/25/2016 10:57 AM - Igor Skornyakov

Constantin Asofiei wrote:

In note 59 you state that stack trace is written also for MESSAGE statements shown as ALERT-BOX, but I can't make this work. Only stacktraces for ERROR conditions raised (which were not "consumed" via NO-ERROR) are logged.

This is strange. I've just reproduced it.

The additional problem here is that for ChUI or GUI clients, we can't write to STDOUT, we need to maintain a separate log file.

This was my initial idea. But in this case it seems reasonable to implement (at least partially) LOG-MANAGER attributes' support.

#155 - 03/25/2016 11:04 AM - Igor Skornyakov

After the fix the following test failed in the main-regression:

tc_codes_employees_021
tc_codes_employees_024
tc_dc_slot_008
tc_dc_slot_018
tc_job_002
tc_job_clock_002
tc_pay_emp_abs_acc_001

Test restarted

#156 - 03/25/2016 11:05 AM - Constantin Asofiei

Igor Skornyakov wrote:

Constantin Asofiei wrote:

In note 59 you state that stack trace is written also for MESSAGE statements shown as ALERT-BOX, but I can't make this work. Only stacktraces for ERROR conditions raised (which were not "consumed" via NO-ERROR) are logged.

This is strange. I've just reproduced it.

I've used this test:

```
session:debug-alert = true.  
def var h as handle.  
  
message "a" view-as alert-box.
```

and after running it like this: `pro -p aa.p -clientlog bbb.txt`
the bbb.txt file contains only this:

```
[16/03/25@14:58:22.702+0000] P-028059 T-3075057408 1 4GL -- Logging level set to = 1
```

I haven't tested GUI. No other message is added to the log file if I press the Help button.

This was my initial idea. But in this case it seems reasonable to implement (at least partially) LOG-MANAGER attributes' support.

If we don't want to add support for LOG-MANAGER handle, we can use a directory configuration (maybe the same clientConfig/outputToFile node?) to simulate the LOG-MANAGER:LOGFILE-NAME attribute and -clientlog option; for GUI and ChUI clients, this option will mean that the stacktrace will be written to this file, and not STDOUT.

#157 - 03/25/2016 11:10 AM - Igor Skornyakov

Constantin Asofiei wrote:

I've used this test:

[...]

and after running it like this: `pro -p aa.p -clientlog bbb.txt`

the bbb.txt file contains only this:

[...]

I haven't tested GUI. No other message is added to the log file if I press the Help button.

I've used the debug-alert.p attached earlier. I will check with your test.

This was my initial idea. But in this case it seems reasonable to implement (at least partially) LOG-MANAGER attributes' support.

If we don't want to add support for LOG-MANAGER handle, we can use a directory configuration (maybe the same `clientConfig/outputToFile` node?) to simulate the LOG-MANAGER:LOGFILE-NAME attribute and `-clientlog` option; for GUI and ChUI clients, this option will mean that the stacktrace will be written to this file, and not STDOUT.

Looks reasonable. Thank you.

Greg, do you agree with this approach?

Thank you.

#158 - 03/25/2016 11:15 AM - Greg Shah

do you agree with this approach?

Yes. Do not spend time on LOG-MANAGER.

#159 - 03/25/2016 11:16 AM - Igor Skornyakov

Greg Shah wrote:

Yes. Do not spend time on LOG-MANAGER.

OK, thank you.

#160 - 03/25/2016 11:22 AM - Constantin Asofiei

Igor Skornyakov wrote:

Constantin Asofiei wrote:

I've used this test:

[...]

and after running it like this: `pro -p aa.p -clientlog bbb.txt`
the `bbb.txt` file contains only this:

[...]

I haven't tested GUI. No other message is added to the log file if I press the Help button.

I've used the `debug-alert.p` attached earlier. I will check with your test.

I think I understand now - the stacktrace is written if I remove the `VIEW-AS ALERT-BOX` from the `MESSAGE` statement and add the `SESSION:APPL-ALERT-BOXES = true`. Looks like this log file is tightly coupled with the simple `MESSAGE` statement and `SESSION:APPL-ALERT-BOXES` (which transforms a simple message into an alert-box).

#161 - 03/25/2016 02:56 PM - Igor Skornyakov

After a second run only the following tests from main-regression failed in both runs:

`tc_job_002`

`tc_job_clock_002`

I tried to run `tc_job_clock_002` manually but the it failed even earlier (seems to be problem with data).

Test restarted

#162 - 03/25/2016 06:26 PM - Igor Skornyakov

tc_job_clock_002 test passed.
Starting CTRL-C tests.

#163 - 03/26/2016 02:31 AM - Igor Skornyakov

All CTRL-C test passed.
The branch 1612a is ready for merge.

#164 - 03/26/2016 08:29 AM - Greg Shah

Please merge 1612a to trunk.

#165 - 03/26/2016 08:45 AM - Igor Skornyakov

Merged to the trunk revision 10986.
The task branch 1612a was archived.

#166 - 03/26/2016 08:50 AM - Igor Skornyakov

Created task branch 1612b

#167 - 03/28/2016 12:13 PM - Igor Skornyakov

Finished client logging.

Committed to the task branch 1612b revision 10988.

Ready for the code review.

#168 - 03/28/2016 01:45 PM - Greg Shah

Code Review Task Branch 1612b Revision 10988

1. My primary concern is that you are re-introducing a direct UI dependency in the ErrorManager. We originally had such dependencies, but had eliminated those by using the ErrorWriter abstraction. Please look at the 3 places we implement that interface and extend it as needed. That will allow the removal of the direct dependency. You'll also find that this mechanism is what allows client-side errors to be logged too. Look carefully at how we use ErrorWriterInteractive and ErrorWriterBatch to see the client-side part.
2. Shouldn't ThinClient.clientLog() only output the stack trace if the stack parameter is non-null? This would allow the same API to be used (later) for more generic logging.
3. I don't understand why SessionUtils.init() is being called from isDebugEnabled() and isClientlog(). Shouldn't the call from initialValue() be enough?
4. Why was the text APPL-ALERT-BOX SESSION attribute added to the last line of javadoc for LogicalTerminal.messageBox()?

#169 - 03/28/2016 01:56 PM - Igor Skornyakov

Greg Shah wrote:

Code Review Task Branch 1612b Revision 10988

1. My primary concern is that you are re-introducing a direct UI dependency in the ErrorManager. We originally had such dependencies, but had eliminated those by using the ErrorWriter abstraction. Please look at the 3 places we implement that interface and extend it as needed. That will allow the removal of the direct dependency. You'll also find that this mechanism is what allows client-side errors to be logged too. Look carefully at how we use ErrorWriterInteractive and ErrorWriterBatch to see the client-side part.

2. Shouldn't ThinClient.clientLog() only output the stack trace if the stack parameter is non-null? This would allow the same API to be used (later) for more generic logging.

OK I will fix it.

3. I don't understand why SessionUtils.init() is being called from isDebugEnabled() and isClientlog(). Shouldn't the call from initialValue() be enough?

Initially I expected that adding code to the initialValue() is enough. However it appeared to be not the case.

4. Why was the text APPL-ALERT-BOX SESSION attribute added to the last line of javadoc for LogicalTerminal.messageBox()?

This is just a typo. Fixed.

#170 - 03/28/2016 01:59 PM - Greg Shah

3. I don't understand why SessionUtils.init() is being called from isDebugEnabled() and isClientlog(). Shouldn't the call from initialValue() be enough?

Initially I expected that adding code to the initialValue() is enough. However it appeared to be not the case.

You should never be able to call get() without initialValue() having been executed. This is a pre-existing bug in our system which needs to be fixed. See if you can figure it out.

#171 - 03/28/2016 02:02 PM - Igor Skornyakov

Greg Shah wrote:

You should never be able to call `get()` without `initialValue()` having been executed. This is a pre-existing bug in our system which needs to be fixed. See if you can figure it out.

The problem is that sometimes the `StandardServer.getClientParameters()` returns null when called from the `initialValue()`. If this is unexpected should I fix it?

#172 - 03/28/2016 02:12 PM - Igor Skornyakov

The issues mentioned in the code review have been fixed.

Committed to the task branch 1612b revision 10989.

#173 - 03/28/2016 03:10 PM - Greg Shah

Code Review Task Branch 1612b Revision 10989

1. In `ErrorHandler.recordOrShowError()`, I think the direct use of the `ProcedureManager` will be a problem when that code is called from the client. The `ProcedureManager` is not accessible on the client side today.

Constantin: what do you think?

2. `ErrorWriterBatch.java`, `ErrorWriterInteractive.java` `ErrorWriterServer.java` and `ErrorWriter.java` need history entries.

#174 - 03/28/2016 03:11 PM - Greg Shah

You should never be able to call `get()` without `initialValue()` having been executed. This is a pre-existing bug in our system which needs to be fixed. See if you can figure it out.

The problem is that sometimes the `StandardServer.getClientParameters()` returns null when called from the `initialValue()`. If this is unexpected should I fix it?

What is call stack when this happens?

#175 - 03/28/2016 03:22 PM - Igor Skorniyakov

Greg Shah wrote:

The problem is that sometimes the `StandardServer.getClientParameters()` returns null when called from the `initialValue()`. If this is unexpected should I fix it?

What is call stack when this happens?

```
ServerDriver [Java Application]
  Daemon Thread [Conversation [00000001:bogus]] (Suspended)
    SessionUtils.init(SessionUtils$WorkArea) line: 854
    SessionUtils.access$200(SessionUtils$WorkArea) line: 63
    SessionUtils$1.initialValue() line: 88
    SessionUtils$1.initialValue() line: 67
    SessionUtils$1(ContextLocal<T>).get(boolean) line: 417
    SessionUtils$1(ContextLocal<T>).get() line: 374
    SessionUtils.getSessionTooltips() line: 804
    LogicalTerminal.getSessionTooltips() line: 13744
    NativeMethodAccessorImpl.invoke0(Method, Object, Object[]) line: not available [native method]
    NativeMethodAccessorImpl.invoke(Object, Object[]) line: 62
    DelegatingMethodAccessorImpl.invoke(Object, Object[]) line: 43
    Method.invoke(Object, Object...) line: 498
    MethodInvoker.invoke(Object[]) line: 76
    Dispatcher.processInbound(InboundRequest, boolean, NetResource) line: 705
    Conversation.block() line: 319
    Conversation.run() line: 163
    Thread.run() line: 745
```

#176 - 03/28/2016 03:46 PM - Greg Shah

Strange. If this call occurs after the `StandardServer.standardEntry()` is called, then how can the `ClientParameters` be null?

Please track this down.

#177 - 03/28/2016 03:47 PM - Igor Skornyakov

Greg Shah wrote:

Strange. If this call occurs after the `StandardServer.standardEntry()` is called, then how can the `ClientParameters` be null?

Please track this down.

OK.

#178 - 03/28/2016 04:03 PM - Igor Skornyakov

Greg Shah wrote:

Strange. If this call occurs after the `StandardServer.standardEntry()` is called, then how can the `ClientParameters` be null?

Please track this

The `StandardServer.standardEntry()` is called after `LogicalTerminal.getSessionToolTips()` is called. Trying to understand why.

#179 - 03/28/2016 04:13 PM - Igor Skornyakov

This happens because of the `ToolTip.setSessionToolTips(tc.server.getSessionToolTips())` call at the end of the `ThinClient.loadEnvironment()` method.

#180 - 03/28/2016 06:44 PM - Greg Shah

Code Review Task Branch 1612a Revision 10990

History number 802 in `ErrorWriterServer` should be 002.

#181 - 03/28/2016 06:44 PM - Greg Shah

This happens because of the `ToolTip.setSessionToolTips(tc.server.getSessionToolTips())` call at the end of the `ThinClient.loadEnvironment()` method.

Good find. Please do fix it.

#182 - 03/29/2016 05:21 AM - Igor Skornyakov

Greg Shah wrote:

Good find. Please do fix it.

Done.

Committed to the task branch 1612b revision 10991.

To make the change as local as possible I decided to change `SessionUtils.getSessionTooltips()` only. This approach is similar to one in the `SessionUtils.getParameter()`.

The history record in the `ErrorWriterServer` is fixed as well.

#183 - 03/29/2016 09:09 AM - Greg Shah

To make the change as local as possible I decided to change `SessionUtils.getSessionTooltips()` only. This approach is similar to one in the `SessionUtils.getParameter()`.

The reason I asked for the change in the first place was because the logic of all this is incorrect. I don't want to leave any of this broken logic in place. It is confusing and makes the code hard to understand. And it is all unnecessary.

1. Remove the `init()` method and any calls to it.
2. Move the directory lookups for `param` to the `WorkArea.param` initializer OR to the `initialValue()` method.
3. Remove any directory lookup code from `getParameter()` and `getSessionTooltips()`. These should be simple lookups in the `workarea`.
4. Add directory lookup code for `isDebugEnabled` and `isClientLog` so that these can be set to a known default from the directory. The lookup can be in the initializer or in the `initialValue()` method.
5. Move the `ThinClient.loadEnvironment()` call to `getSessionTooltips()` to the `ThinClient.initializePost()` which is done after the call to `StandardServer.standardEntry`. This is safe because there can be no user interface usage until after both `StandardServer.standardEntry` and `ThinClient.initializePost()` is called.

#184 - 03/29/2016 09:21 AM - Constantin Asofiei

Greg Shah wrote:

1. In `ErrorManager.recordOrShowError()`, I think the direct use of the `ProcedureManager` will be a problem when that code is called from the client. The `ProcedureManager` is not accessible on the client side today.

Constantin: what do you think?

You are correct, `ErrorManager` is used both from client and server-side. And when called from client-side, `ProcedureManager` will not be accessible - in this case, an API which invokes the server-side is needed.

#185 - 03/29/2016 09:47 AM - Igor Skornyakov

Greg Shah wrote:

5. Move the `ThinClient.loadEnvironment()` call to `getSessionTooltips()` to the `ThinClient.initializePost()` which is done after the call to `StandardServer.standardEntry`. This is safe because there can be no user interface usage until after both `StandardServer.standardEntry` and `ThinClient.initializePost()` is called.

Sorry Greg,

As I can see the `ThinClient.initializePost()` is called even before `ThinClient.loadEnvironment()`

#186 - 03/29/2016 10:15 AM - Greg Shah

As I can see the `ThinClient.initializePost()` is called even before `ThinClient.loadEnvironment()`

You're right. Propose an alternative that does not call `WorkArea.get()` before the `ClientParameters` are available.

#187 - 03/29/2016 10:29 AM - Igor Skornyakov

Constantin Asofiei wrote:

Greg Shah wrote:

1. In `ErrorManager.recordOrShowError()`, I think the direct use of the `ProcedureManager` will be a problem when that code is called from the client. The `ProcedureManager` is not accessible on the client side today.

Constantin: what do you think?

You are correct, `ErrorManager` is used both from client and server-side. And when called from client-side, `ProcedureManager` will not be accessible - in this case, an API which invokes the server-side is needed.

Thank you Constantin.

I've fixed this.

Committed to the task branch 1612b revision 10992.

#188 - 03/29/2016 11:02 AM - Greg Shah

Code Review Task Branch 1612b Revision 10992

Generally the changes are OK.

However, I think this code in `ErrorManager.recordOrShowError()` is incorrect:

```
if (SessionUtils.isClientlog())
{
    if (errorWriter != null)
    {
        errorWriter.logStackTrace(masterMsg);
    }
}
```

Note 59 says this:

If `SESSION:DEBUG-ALERT` attribute is set to true and the `-clientlog` startup parameter is specified then for all error messages and MESSAGE statements shown as ALERT-BOX a stack trace is written to the log after the message:

This suggests the code should be this:

```
if (SessionUtils.isDebugEnabled() && SessionUtils.isClientlog())
{
    if (errorWriter != null)
    {
        errorWriter.logStackTrace(masterMsg);
    }
}
```

#189 - 03/29/2016 11:06 AM - Igor Skornyakov

Greg Shah wrote:

Code Review Task Branch 1612b Revision 10992

Generally the changes are OK.

However, I think this code in `ErrorManager.recordOrShowError()` is incorrect:

[...]

Note 59 says this:

If `SESSION:DEBUG-ALERT` attribute is set to true and the `-clientlog` startup parameter is specified then for all error messages and `MESSAGE` statements shown as `ALERT-BOX` a stack trace is written to the log after the message:

This suggests the code should be this:

[...]

You're right. Please note however that `SessionUtils.isClientlog()` returns true iff both client log is specified **and** `DEBUG-ALERT` is true. My initial version was exactly like you suggested but I've changed it as there are no places (so far) where we need to check for the client log only.

#190 - 03/29/2016 11:08 AM - Greg Shah

My initial version was exactly like you suggested but I've changed it as there are no places (so far) where we need to check for the client log only.

Later we are going to have to split this out anyway, so it might as well be done now. That also has the advantage of making it easier to understand for the reader. Others could make the same mistake I did. I didn't check the called code since it is unexpected for `isClientLog()` to check `isDebugEnabled()`.

#191 - 03/29/2016 11:13 AM - Igor Skornyakov

Greg Shah wrote:

Later we are going to have to split this out anyway, so it might as well be done now. That also has the advantage of making it easier to understand for the reader. Others could make the same mistake I did. I didn't check the called code since it is unexpected for isClientLog() to check isDebugEnabled().

I see. I've fixed it. (Not committed yet).

#192 - 03/29/2016 11:22 AM - Igor Skornyakov

Greg Shah wrote:

You're right. Propose an alternative that does not call WorkArea.get() before the ClientParameters are available.

I understand that not much useful can be done after the invocation of the standardEntry which is the first place where ClientParameters are known at the server side. So the only solution can be with using directory service. I'm not familiar with its functionality and looking on it now.

#193 - 03/29/2016 12:19 PM - Greg Shah

We don't want to expose the directory configuration for SessionUtils to the client. The directory node that is checked is something private to SessionUtils.

Instead, why not send the ClientParameters to the server before ThinClient.loadEnvironment() is called?

#194 - 03/29/2016 12:24 PM - Igor Skornyakov

Greg Shah wrote:

We don't want to expose the directory configuration for SessionUtils to the client. The directory node that is checked is something private to SessionUtils.

Instead, why not send the ClientParameters to the server before ThinClient.loadEnvironment() is called?

I was thinking about it. My concern is this opens the door for submitting two (possibly different) versions of ClientParameters to the server. It is possible of course just ignore the one from the standardEntry() if another instance is already found. Is it OK?
Thank you.

#195 - 03/29/2016 12:26 PM - Greg Shah

My concern is this opens the door for submitting two (possibly different) versions of ClientParameters to the server. It is possible of course just ignore the one from the standardEntry() if another instance is already found. Is it OK?

No. Just remove the ClientParameters instance from standardEntry(). I don't ever see a reason to send it twice. The values are not going to change in between the calls. It will only be sent on its own.

#196 - 03/29/2016 12:27 PM - Igor Skornyakov

Greg Shah wrote:

No. Just remove the ClientParameters instance from standardEntry(). I don't ever see a reason to send it twice. The values are not going to change in between the calls. It will only be sent on its own.

I see. Thank you.

#197 - 03/29/2016 01:01 PM - Igor Skornyakov

Client command line parameters' processing refactored.

Committed to the task branch 1612b revision 10993.

#198 - 03/29/2016 01:15 PM - Greg Shah

Code Review Task Branch 1612b Revision 10993

1. ClientCore should not import com.goldencode.p2j.ui.client.gui.*.
2. ThinClient.setSessionToolips() should be named ThinClient.setSessionTooltips().
3. MainEntry, StandardServer are missing history entries.
4. This code in MainEntry has 3 coding standards issues:

```
/**
 * Defines the only method which serves as the main remote entry point.
 *          <--- BLANK LINE SHOULD NOT BE HERE
 */
public boolean standardEntry();
```

```
/**
 * Set the client parameters.
 *
 * @param params
 *          A container with any client-related parameters.
 *          <--- BLANK LINE SHOULD NOT BE HERE
```

```
*/  
public void setClientParams(ClientParameters params);  
    <--- BLANK LINE SHOULD NOT BE HERE  
}
```

5. There should be a blank line after `ThinClient.setSessionToolips()`.

#199 - 03/29/2016 01:26 PM - Igor Skornyakov

Greg Shah wrote:

Code Review Task Branch 1612b Revision 10993

Fixed.

Committed to the task branch 1612b revision 10995.

#200 - 03/29/2016 01:29 PM - Greg Shah

Code Review Task Branch 1612b Revision 10995

Everything looks good except you forgot about this one:

```
public void setClientParams(ClientParameters params);  
    <---- HERE  
}
```

#201 - 03/29/2016 01:32 PM - Igor Skornyakov

Greg Shah wrote:

Code Review Task Branch 1612b Revision 10995

Everything looks good except you forgot about this one:

[...]

Sorry. Fixed.

Committed to the task branch 1612b revision 10996.

#202 - 03/29/2016 01:34 PM - Greg Shah

It looks good. Please rebase to the latest trunk level. After that you can regression test.

#203 - 03/29/2016 01:39 PM - Igor Skornyakov

Rebased task branch 1612b from P2J trunk rev 10987. The latest revision is now 10997.

Starting regression testing.

#204 - 03/30/2016 03:25 AM - Igor Skornyakov

All regression tests except gso_ctrlc_3way_tests and tc_job_002 passed.

#205 - 03/30/2016 06:53 AM - Igor Skornyakov

ctrl-c-3way tests passed.

#206 - 03/30/2016 08:50 AM - Greg Shah

Great! Please merge 1612b to trunk.

#207 - 03/30/2016 09:57 AM - Igor Skornyakov

The task branch merged to the trunk revision 10988 and archived.

#208 - 03/30/2016 12:23 PM - Greg Shah

In our gap analysis rules, I was planning to set the following:

```
<rule>attrs.put(prog.kw_dbg_alrt, rw.cvt_lvl_full | rw.rt_lvl_full_restr)</rule> <!-- debug-alert; the d
ebug button in the dialog doesn't bring up a debugger and the stack trace line numbers are not valid -->
<rule>attrs.put(prog.kw_imm_disp, rw.cvt_lvl_full | rw.rt_lvl_partial)</rule> <!-- immediate-display, cl
ient-side implementation doesn't change behavior based on the flag -->
```

Is this correct?

#209 - 03/30/2016 12:28 PM - Igor Skornyakov

Greg Shah wrote:

In our gap analysis rules, I was planning to set the following:

[...]

Is this correct?

Yes, this is correct (in fact we just do not display line numbers). Please note also the the stack trace dialog for ChUI is not implemented yet.

#210 - 03/30/2016 12:47 PM - Greg Shah

Thanks, those are useful clarifications.

#211 - 03/31/2016 08:42 AM - Greg Shah

- Status changed from New to Closed

#212 - 04/01/2016 11:58 AM - Constantin Asofiei

Ovidiu reports a regression when building a SESSION as a handle via `SessionUtils.asHandle()` - `isDataEntryReturn` and `isUsingAlertBoxes` exist in both `SessionUtils` and `LogicalTerminal`, thus the static proxy can not be built.

```
Caused by: java.lang.RuntimeException: Error creating static proxy for interface com.goldencode.p2j.util.CommonSession: ambiguous static method com.goldencode.p2j.util.logical.isDataEntryReturn() (found in com.goldencode.p2j.ui.LogicalTerminal and com.goldencode.p2j.util.SessionUtils).
    at com.goldencode.proxy.StaticProxy.init(StaticProxy.java:285)
    at com.goldencode.proxy.StaticProxy.<init>(StaticProxy.java:174)
    at com.goldencode.proxy.StaticProxy.obtain(StaticProxy.java:151)
    at com.goldencode.p2j.util.SessionUtils.asHandle(SessionUtils.java:128)
    [...]
```

The `LogicalTerminal` versions need to be renamed to something else (maybe append an `_` to them).

#213 - 04/01/2016 11:58 AM - Constantin Asofiei

Constantin Asofiei wrote:

The `LogicalTerminal` versions need to be renamed to something else (maybe append an `_` to them).

I meant prefix them with a `_`...

#214 - 04/01/2016 03:50 PM - Greg Shah

The idea is reasonable.

Ovidiu: please put this into 2181a.

#215 - 04/01/2016 04:14 PM - Ovidiu Maxiniuc

Greg Shah wrote:

Ovidiu: please put this into 2181a.

Done. Revision is 11004.

#216 - 04/04/2016 10:55 AM - Constantin Asofiei

The fix for regression in note 212 was moved from branch 2181a to branch 3012a rev 10996.

#217 - 04/07/2016 07:09 AM - Paul E

Not really sure which issue to put this comment on, or whether to raise a new issue.

We've not been running our test pipeline for the last few days due to using the assigned infrastructure for other things. On running it again last night we're seeing the stacktrace from note 2 of redmine #3050.

This is in P2J bzz rev 10991.

Can this fix be pushed our way please?

#218 - 04/07/2016 08:17 AM - Paul E

p.s. alternatively if a fix can't be pushed our way soon then can you let me know and we'll reconvert with an older P2J version?

#219 - 04/07/2016 09:02 AM - Eric Faulhaber

I'm pushing trunk revision 10996 your way now, which will get you this fix. Sorry for the delay.

#220 - 11/16/2016 12:12 PM - Greg Shah

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

Files

ca_upd20130221d.zip	130 KB	02/21/2013	Constantin Asofiei
ca_upd20130222b.zip	152 KB	02/22/2013	Constantin Asofiei
ca_upd20130222c.zip	1.74 KB	02/22/2013	Constantin Asofiei
ca_upd20130222g.zip	153 KB	02/22/2013	Constantin Asofiei
ca_upd20131001b.zip	307 KB	10/01/2013	Constantin Asofiei
ca_upd20131002a.zip	246 KB	10/02/2013	Constantin Asofiei
ca_upd20131002d.zip	53.9 KB	10/02/2013	Constantin Asofiei
ca_upd20131003c.zip	498 KB	10/03/2013	Constantin Asofiei
ca_upd20131004b.zip	498 KB	10/04/2013	Constantin Asofiei
ca_upd20131004c.zip	105 KB	10/04/2013	Constantin Asofiei
ca_upd20131004d.zip	103 KB	10/04/2013	Constantin Asofiei
high_priority_attrs_tests_20131007a.zip	9.49 KB	10/07/2013	Constantin Asofiei
t1612.p	2.96 KB	03/02/2016	Igor Skornyakov
id.p	864 Bytes	03/06/2016	Igor Skornyakov
id-0-true-false.mkv	384 KB	03/06/2016	Igor Skornyakov
id-9999-true-false.mkv	456 KB	03/06/2016	Igor Skornyakov
id-0-false-true.mkv	331 KB	03/06/2016	Igor Skornyakov

id-9999-false-true.mkv	387 KB	03/06/2016	Igor Skornyakov
debug-alert.png	32.4 KB	03/07/2016	Igor Skornyakov
immediate-display.mkv	302 KB	03/08/2016	Igor Skornyakov
debug-alert.p	4.17 KB	03/25/2016	Igor Skornyakov