Base Language - Feature #2002

refactor some "shared" attributes

02/18/2013 03:04 PM - Greg Shah

Status:	Closed	**Start date:**	02/18/2013
Priority:	Normal	**Due date:**	
Assignee:	Constantin Asofiei	**% Done:**	100%
Category:		**Estimated time:**	8.00 hours
Target version:	Conversion Support for Server Features		
billable:	No	**version:**	
vendor_id:	GCD		

Description

Related issues:

Related to Base Language - Feature #2041: implement runtime support for "shar...	**Closed**	**10/23/2013**	**11/01/2013**

History

#1 - 02/18/2013 03:13 PM - Greg Shah

Please implement appropriate interfaces for the following attributes which are shared between multiple resources:

ADM-DATA
COLUMN-LABEL
DATA-TYPE
FORMAT
LABEL
LOCALNAME
NAME
NAMESPACE-URI
PRIVATE-DATA
SENSITIVE
TYPE
UNIQUE-ID

Please create the minimum number of proper interfaces. This would be an interface that can be implemented by resources that share the same attributes such that they have no extra attributes (that would not normally be there).

Once the interfaces are there, please implement the backing runtime stubs in every resource that:

1. Already is supported by P2J.
2. Needs to support those attributes.

Some of the above may already be implemented in a proper interface. Even if a proper interface already exists, please make sure it is implemented in the runtime in the right places.

Finally, refactor/implement stubs for any of the chaining related attributes (NEXT-SIBLING/PREV-SIBLING and FIRST-CHILD/LAST-CHILD for SESSION handles as well as widgets) which still need to be reworked.

#2 - 02/19/2013 08:51 AM - Constantin Asofiei

- File attr_grouping.ods added

Except DATA-TYPE and FORMAT (which are common for BUFFER-FIELD and widgets), I don't see any way of grouping these attributes "by resource". More, looks like NEXT-SIBLING is supported by BUFFER, while PREV-SIBLING is not supported by BUFFER handle.

Spliting these in many interfaces will not help us in the implementation classes... what I suggest is to be a little more relaxed in grouping the attributes, following this approach:

- an interface groups all attributes which apply to as many resources as possible, following a reasoable approach (i.e. attribute/method semantic). Idea is, if we have a set of attributes S1 which applies to N resources and there is another attribute A1 which applies to N-1 resources, then we put them all in the same interface.
- when a certain resource has to implement a not-supported attribute, its implementation will call the handle.invalidAttribute API which, depending on the state of the resource, will show the appropriate error. This API will contain the code from the handle.InvalidAttributeAccess.invoke.

This way, we don't make a lot of interfaces and it may be possible to structure the implementation classes so that we don't duplicate code for these attributes.
I'll let you know my further analysis.

#3 - 02/19/2013 01:32 PM - Constantin Asofiei

I think the best grouping should be this:

1. COLUMN-LABEL (BUFFER-FIELD), DATA-TYPE, FORMAT, LABEL - all in an interface named CommonField
2. ADM-DATA, UNIQUE-ID - all in a single interface
3. LOCAL-NAME - its own interface
4. NAMESPACE-URI - its own interface
5. NAME, PRIVATE-DATA, NEXT-SIBLING, PREV-SIBLING - leave them in CommonHandleChain
6. SENSITIVE - already has its own interface
7. TYPE - its correct location is the current location, CommonHandle, as it applies to all.

I see that the latest update from Vadim on #1668 restructures some of these attributes... I think is best to continue looking at this when all updates are at least in testing and no one else is working on these attributes.

#4 - 02/19/2013 04:33 PM - Greg Shah

Please go ahead with this task in combination with #1974 (#1668). Prepare a single update that cleans all of this up.

#5 - 02/21/2013 05:33 AM - Constantin Asofiei

- File ca_upd20130221b.zip added

This update cleans #1668 errors + refactors attributes. I'm running conversion now, I'll let you know how it goes.

LE: WriteableName needs to be removed.

#6 - 02/21/2013 05:40 AM - Constantin Asofiei

- Status changed from New to WIP

#7 - 02/21/2013 10:40 AM - Constantin Asofiei

The only change in generated code is the one related to label: now it unwraps using handle.unwrapCommonField, instead of handle.unwrapWidget. The hierarchy is OK (CommonWidget extends CommonField), so there should be no runtime problems related to this.

#8 - 02/21/2013 10:45 AM - Constantin Asofiei

There is one MAJIC-side change needed, as GenericWidget.getFormat now returns character instead of String (I chose to rename the old API which returned String to _getFormat). There is another case of rename: String GenericFrame.getName is now String GenericFrame._getName.

#9 - 02/21/2013 12:07 PM - Greg Shah

There are 2 check-ins that will occur shortly. Be prepared to merge up to the lastest level. Also, please prepare any Majic changes that need to be made. I plan to put this update in next.

I will provide code feedback shortly (I'm working my way through the update).

#10 - 02/21/2013 12:11 PM - Constantin Asofiei

> Also, please prepare any Majic changes that need to be made.

See #2023

#11 - 02/21/2013 12:50 PM - Greg Shah

Code Review Feedback:

1. Please put a TODO comment into BrowseColumnWidget that it should not always return "FILL-IN" as its type. From the 4GL docs:

```
For browse columns, the TYPE attribute returns "COMBO-BOX", "FILL-IN", or
"TOGGLE-BOX". If you specify the Browse Column Type (-browcoltype) startup
parameter, the TYPE attribute for browse columns returns "BROWSE-COLUMN"
regardless of the actual column type.
```

2. Your FieldReference history entry is incomplete.

3. The XNodeRefImpl history number is wrong.

4. handle.invalidAttribute() is missing javadoc.

5. It seems like SESSION:FIRST-BUFFER should be in the persist package, not in EnvironmentOps.

6. Wasn't the idea behind the BufferField design was to avoid get/set methods that could conflict with DMO getters and setters? Is this still an issue with CommonField or am I mis-remembering?

7. Shouldn't the Sizeable attrs (width-chars, width-pixels, height-chars, height-pixels) be calling an interface method instead of LogicalTerminal.whatever()?

If you can merge and make the changes quickly, we will apply this to staging next and we will run conversion there. We will batch up runtime regression testing with some other changes, so the harness will be deferred for now.

#12 - 02/21/2013 01:13 PM - Constantin Asofiei

> 1. Please put a TODO comment into BrowseColumnWidget that it should not always return "FILL-IN" as its type. From the 4GL docs:

Fixed.

> 2. Your FieldReference history entry is incomplete.

Fixed.

> 3. The XNodeRefImpl history number is wrong.

Fixed.

> 4. handle.invalidAttribute() is missing javadoc.

Fixed.

> 5. It seems like SESSION:FIRST-BUFFER should be in the persist package, not in EnvironmentOps.

I agree, but which class? I don't think BufferImpl is OK, as FIRST-BUFFER can be for a temp-table too. We can leave this for #1612 work (my approach there was to let it convert to a SessionUtils API which in turn will call the real implementation from the persist package). In #1612, I used the same approach for some appserver-related attributes, as converting them to ServerImpl.whatever didn't look clean to me.

> 6. Wasn't the idea behind the BufferField design was to avoid get/set methods that could conflict with DMO getters and setters? Is this still an issue with CommonField or am I mis-remembering?

I'm not sure what you mean here, couse I'm not up-to-date with all the info about DB-related handles/attrs/methods. Also, is BufferField a new to-be-released interface? Because I can't see how CommonField could conflict with DMO properties, as currently CommonField is used only by FieldReference, on the persist side.

> 7. Shouldn't the Sizeable attrs (width-chars, width-pixels, height-chars, height-pixels) be calling an interface method instead of LogicalTerminal.whatever()?

The approach in the methods_attributes.rules SESSION section is to qualify the API using the class which statically implements it (as these are emitted in more than one class), and get rid of the class qualifier if a VAR_HANDLE is used to access the attribute (see lines 789 and later).

> If you can merge and make the changes quickly, we will apply this to staging next and we will run conversion there. We will batch up runtime regression testing with some other changes, so the harness will be deferred for now.

I will provide a merged update tonight.

#13 - 02/21/2013 01:24 PM - Eric Faulhaber

Greg Shah wrote:

> 6. Wasn't the idea behind the BufferField design was to avoid get/set methods that could conflict with DMO getters and setters? Is this still an issue with CommonField or am I mis-remembering?

This is a non-issue. The conflict is only with methods in the Buffer interface, because each DMO interface has an inner class, <DMO>.Buf which extends both the generated DMO interface AND Buffer. Since the generated DMO interface uses JavaBean naming conventions of get/is/set method name prefixes, we cannot use this convention in the Buffer interface. No other interfaces are affected by this restriction.

#14 - 02/21/2013 01:33 PM - Constantin Asofiei

- File ca_upd20130221f.zip added

Fixed issues 1 to 4, merged with 10181.

#15 - 02/21/2013 02:55 PM - Greg Shah

FYI:

```
  [javadoc] /mnt/san/sata/gc/20121029/p2j/src/com/goldencode/p2j/util/KeyReader.java:436: warning - @param arg
ument "format" is not a parameter name.
  [javadoc] /mnt/san/sata/gc/20121029/p2j/src/com/goldencode/p2j/util/KeyReader.java:447: warning - @param arg
ument "format" is not a parameter name.
  [javadoc] /mnt/san/sata/gc/20121029/p2j/src/com/goldencode/p2j/util/KeyReader.java:470: warning - @param arg
ument "format" is not a parameter name.
  [javadoc] /mnt/san/sata/gc/20121029/p2j/src/com/goldencode/p2j/util/KeyReader.java:481: warning - @param arg
ument "format" is not a parameter name.
```

I am still going ahead with running conversion on staging using the version already posted here. Assuming it passes, I will be checking it in as well. I will probably fix this javadoc error just before check in.

#16 - 02/22/2013 03:02 AM - Constantin Asofiei

Committed to bzr revision 10182.

#17 - 02/23/2013 11:50 AM - Greg Shah

- % Done changed from 0 to 100

- Status changed from WIP to Closed

#18 - 11/16/2016 11:07 AM - Greg Shah

- Target version changed from Milestone 4 to Conversion Support for Server Features

Files

attr_grouping.ods	9.76 KB	02/19/2013	Constantin Asofiei
ca_upd20130221b.zip	584 KB	02/21/2013	Constantin Asofiei
ca_upd20130221f.zip	585 KB	02/21/2013	Constantin Asofiei