

Base Language - Feature #2123

map the converted method/attribute names to their legacy name

04/12/2013 05:21 AM - Constantin Asofiei

| | |
|---|----------------------------------|
| Status: New | Start date: |
| Priority: Normal | Due date: |
| Assignee: | % Done: 0% |
| Category: | Estimated time: 0.00 hour |
| Target version: | version: |
| billable: No | |
| vendor_id: GCD | |
| Description | |
| Related issues: | |
| Related to Base Language - Feature #1920: implement persistent procedures | Closed |

History

#1 - 04/12/2013 05:28 AM - Constantin Asofiei

Note 54 from [#1920](#):

About the invalid API calls (when the invoked API is not valid for the current referent). 4GL shows these kind of messages:

```
**<API> is not a queryable attribute for <referent-type>. (4052)
```

where:

- <referent-type> is i.e. "PSEUDO-WIDGET" for FILE-INFO, SESSION, ERROR-STATUS, LAST-EVENT handles or i.e. "PROCEDURE" for PROCEDURE handles
- <API> is the actual 4GL API name being invoked. Is not that easy to fully support each 4GL API name, as the name is lost during conversion. For now, is best to let <API> be the Java name of the actual invoked method, and in the future produce a map which has as key the Java API name and as value its 4GL equivalent (I will make it so that this will be pretty easy to implement, or better I will add support for a few APIs and will complete them later, is just tedious work).

Currently, the handle.apisToAttrOrMthd map is not populated. As the list of attributes/methods is pretty large, we should move this to another helper class, which can be maintained easier. We have two ways of solving this:

1. solve each case "on the way", as runtime regression testing exposes "invalid error messages"
2. populate the entire map up-front, and go through each implemented attribute/method and map it. It gets tiresome when working with such large lists, so I estimate ~20-30 hours of work if we need to do it upfront.

#2 - 04/12/2013 06:04 AM - Constantin Asofiei

- *Parent task deleted (#1920)*