

## Base Language - Bug #7200

### fix INIT and LABEL for LIKE defined fields or variables

03/15/2023 01:19 PM - Roger Borrello

<b>Status:</b> New	<b>Start date:</b>
<b>Priority:</b> Normal	<b>Due date:</b>
<b>Assignee:</b>	<b>% Done:</b> 0%
<b>Category:</b>	<b>Estimated time:</b> 0.00 hour
<b>Target version:</b>	
<b>billable:</b> No	<b>case_num:</b>
<b>vendor_id:</b> GCD	
<b>Description</b>	
<b>Related issues:</b>	
Related to Base Language - Bug #5288: Some date field character comparisons h... <b>New</b>	

### History

#### #2 - 03/15/2023 01:26 PM - Roger Borrello

- Related to Bug #5288: Some date field character comparisons have runtime incompatibilities during runtime added

#### #3 - 03/15/2023 01:32 PM - Roger Borrello

- File `initializer_var_like_field.p` added
- File `initializer_field_like_var.p` added
- File `7190_both-side-by-side.png` added

Testcases are checked into old testcases and attached.

Interestingly, the code using the fields as the "model" (`initializer_var_like_field.p`) properly initializes the field holding the date given "1.1" as the initial value:

```
field f5b as date init 1.1
...
/*def var vf5a like tt1.f5a.*/
def var vf5b like tt1.f5b.
```

and becomes:

```
@LegacySignature(type = Type.VARIABLE, name = "vf5b")
/*def var vf5a like tt1.f5a.*/ <-- BTW, this comment should be above the Legacy
Signature
date vf5b = UndoableFactory.date();
```

Whereas the variable being the "model" for the field (`initializer_var_like_field.p`):

```
def var vf5b as date init 1.1.
..
/*field f5a like vf5a*/
field f5b like vf5b
```

becomes:

```
@LegacySignature(type = Type.VARIABLE, name = "vf5b")
/*def var vf5a as date init 0.*/
date vf5b = UndoableFactory.date(date.fromLiteral("1.1"));
```

The 4GL has "1.1" become the data "1/1/yy" where yy is the current year (23).

In walking through the code, the fromLiteral ends up constructing using date() since the format isn't recognized:

```
public static date fromLiteral(String str)
{
    Matcher matcher = literalPattern.matcher(str);
    if (!matcher.matches())
    {
        return new date(); // format not recognized
    }
}
...

```

That ends up just returning a date containing unknown, and then that ends up going through the deepAssign:

```
Daemon Thread [Conversation [00000005:bogus]] (Suspended)
date.deepAssign(date) line: 2567
date.<init>(date) line: 668
TypeFactory.date(date) line: 575
UndoableFactory.date(date) line: 486
InitializerLikeTest.<init>() line: 84
InitializerLikeTestConstructorAccess.newInstance() line: not available
...

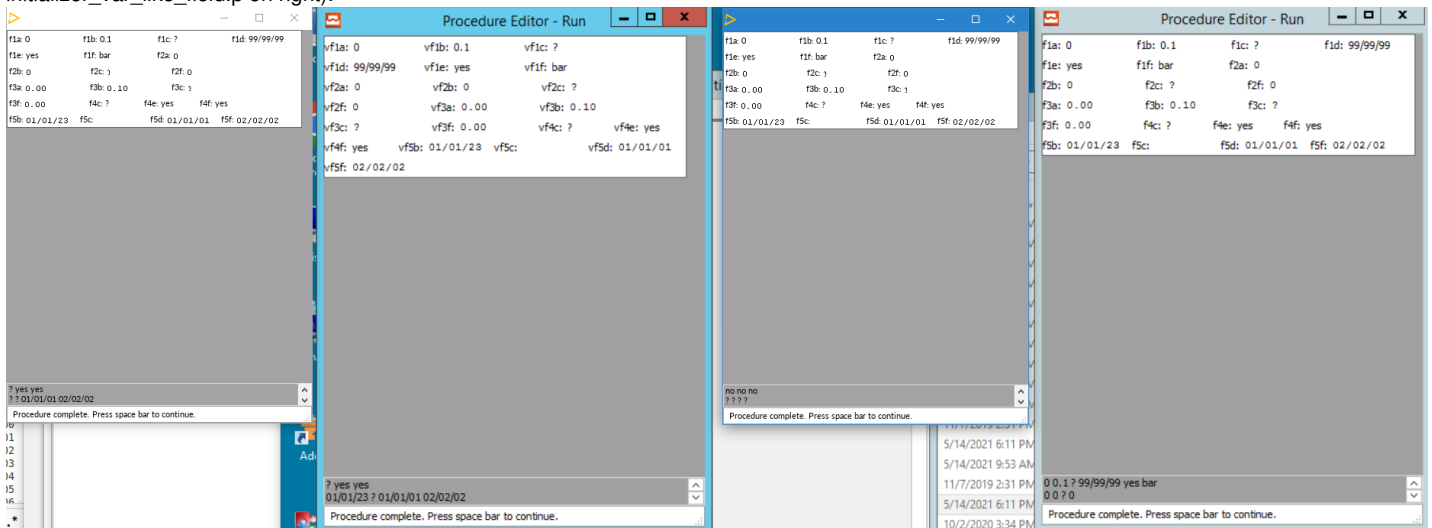
```

For the case where the field is the model, we don't go through the same path and the output of the variable is the correct "01/01/23":

```
Daemon Thread [Conversation [00000006:bogus]] (Suspended)
TypeFactory.date() line: 521
UndoableFactory.date() line: 422
InitializerLikeTestReverse.<init>() line: 84
InitializerLikeTestReverseConstructorAccess.newInstance() line: not available
...

```

In any case, what we seem to have are fields in the table that are formatted more correctly than in the variables (initializer\_field\_like\_var.p on left initializer\_var\_like\_field.p on right):



**Files**

---

7190_both-side-by-side.png	137 KB	03/15/2023	Roger Borrello
initializer_field_like_var.p	1.92 KB	03/15/2023	Roger Borrello
initializer_var_like_field.p	2.01 KB	03/15/2023	Roger Borrello