

## Base Language - Bug #6640

### parsing bug for initial ",0000" for decimal variables or fields

07/27/2022 03:53 AM - Constantin Asofiei

<b>Status:</b>	New	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Greg Shah	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>case_num:</b>	
<b>billable:</b>	No		
<b>vendor_id:</b>	GCD		
<b>Description</b>			

#### History

##### #2 - 07/27/2022 03:57 AM - Constantin Asofiei

- Subject changed from parsing bug for initial ",0000" for decimal values or fields to parsing bug for initial ",0000" for decimal variables or fields

There is a problem with parsing an initial value of ",0000" for decimal variables or fields.

For example:

```
def var d as dec initial ",0000".
message d.
```

and:

```
ADD FIELD "cost" OF "Book" AS decimal
  FORMAT "->>, >>9.99"
  INITIAL ",00000"
```

This needs to be fixed both at schema and program parser.

I assume the leading comma is for the group separator, but some tests need to be done to ensure this. Also, we need to check if this must be configured in p2j.cfg.xml or not (i.e. if it must consider -numsep at the 4GL client startup parameter or some other configuration).

**#4 - 07/29/2022 09:14 AM - Greg Shah**

- Assignee set to *Greg Shah*

**#5 - 08/22/2022 04:11 PM - Greg Shah**

Also, we need to check if this must be configured in p2j.cfg.xml or not (i.e. if it must consider -numsep at the 4GL client startup parameter or some other configuration).

At parsing time the number separator is not configurable. At least, in all my previous testing that has been true. And we have never seen customer code that shows otherwise.

On the other hand, this particular case requires that the initializer be a string literal. This form is something we do support but it is a kind of data type conversion which we implement at parsing time. Since it is a literal, there is no reason to defer it to runtime, but there is a conversion involved. I will note this seems to be the same conversion used at runtime:

```
def var d as decimal init ",001".
```

```
/*  
output:  
1 1  
*/  
message d decimal(",001").
```

I'm looking deeper now.