

Base Language - Bug #5288

Some date field character comparisons have runtime incompatibilities during runtime

04/26/2021 09:19 AM - Roger Borrello

Status: New	Start date:
Priority: Normal	Due date:
Assignee:	% Done: 0%
Category:	Estimated time: 0.00 hour
Target version:	
billable: No	case_num:
vendor_id: GCD	
Description	
Related issues:	
Related to Base Language - Bug #7200: fix INIT and LABEL for LIKE defined fie... New	

History

#2 - 04/26/2021 09:27 AM - Roger Borrello

Comparisons of date fields to certain characters have problems. See testcase:

```
def temp-table ttl field fdate as date format "9999-99-99".
def var h as handle.

create ttl.
ttl.fdate = today.

h = buffer ttl:handle.
if h:buffer-field('fdate'):buffer-value <> "" then message "<>"".
if h:buffer-field('fdate'):buffer-value <> "a" then message "<>a".
if h:buffer-field('fdate'):buffer-value = "b" then message "=b".
message "Done."
```

Shows {00000001:0000000C:bogus} ** Incompatible datatypes found during runtime conversion. (5729)

#3 - 04/26/2021 09:38 AM - Roger Borrello

Constantin Asofiei wrote:

Roger Borrello wrote:

Also, still awaiting review of the update mentioned in #5215-51, which shouldn't hold up this particular issue. I can take it to a new issue, if

need be.

The fix is more complex than that. You can use any string for the comparison operand, and 4GL will not complain that is not a valid date, like: [...]

More, DATE("") must work OK in FWD, too.

And what about comparing with other literals, than string? More, try having the "" empty string as the first operand, as FWD will go down the character.compareTo path.

Just swapping the comparison operands so we go into text.compareTo instead of date.compareTo has better results.

#4 - 04/26/2021 10:55 AM - Roger Borrello

Roger Borrello wrote:

Just swapping the comparison operands so we go into text.compareTo instead of date.compareTo has better results.

I also added assign tdate = date(""). to the testcase, and what it reveals is that an empty string should be treated as Unknown in 4GL. We do handle that correctly, since the constructor for date(String d) builds Unknown properly. However, in the date.compareTo(Object obj), the constructor for date(BaseDataType value) doesn't determine to create Unknown. Instead, the empty string is used, and comparisons fail later.

I could change the constructor to distinguish the empty string, and build Unknown, or simply update the compareTo to short-circuit to return -1 comparing to empty string. I would thing constructing it properly would be better.

I see 4GL also creates Unknown with date("a"). I'll have to see if there's a more exhaustive testcase already in place for various formats (unless someone know of it already).

#5 - 04/26/2021 12:07 PM - Roger Borrello

This update to the date(BaseDataType) constructor does create Unknown when an empty string is passed in:

```
public date(BaseDataType value)
```

```

{
  if (value == null || value.isUnknown())
  {
    setUnknown();
  }
  else
  {
    if (value instanceof character)
    {
      character t = new character("");
      if (t.equals(character.valueOf(value)))
      {
        setUnknown();
        return;
      }
    }
    assign(value);
  }
}

```

However, the compareTo gets to this case:

```

/**
 * An unknown value should be always converted to a unknown date value.
 * A non-unknown value should be always converted to a non-unknown date value.
 * If an unknown value is converted to a non-unknown value the conversion fails.
 * If a non-unknown value is converted to an unknown value the conversion fails.
 */
if (((date) tmp).isUnknown() != bdt.isUnknown())
{
  incompatibleTypesOnConversion();
}

```

I don't understand what it is checking for.

#6 - 04/26/2021 01:15 PM - Greg Shah

This testcase isn't about regular date processing. It is about polymorphic type processing. All the `buffer-field():buffer-value` usage has a polymorphic value. This means its wrapper type depends upon the runtime value of the field. The same code can be used to process different fields, and those different fields can have different data types.

So you must be very careful to focus on those paths in the code. POLY cases tend to be very permissive of comparisons and assignments, converting unlike data types into like types before the comparison or assignment. Typically, we deal with this code using constructors that take a `BaseDataType`. You will also see the `assign(BaseDataType)` for this case. The date class is no exception to this rule.

Please re-evaluate your finding with this in mind.

#7 - 04/26/2021 04:33 PM - Roger Borrello

I think I understand what you are saying. This situations comes up when a new date is created in an effort to perform a comparison. The scenarios where the string being compared to the `buffer-field():buffer-value` is invalid shouldn't result in an exception, but rather just the result of the comparison. A field value could never match against a string that isn't a validly formatted field.

#8 - 04/26/2021 06:11 PM - Roger Borrello

This use case actually comes back true instead of false, as happens in 4GL:

```
def temp-table ttl field fdate as date format "9999-99-99".
def var h as handle.
create ttl.
ttl.fdate = today.
h = buffer ttl:handle.
if h:buffer-field('fdate'):buffer-value <> "today" then message "<>today".
```

The right operand actually gets evaluated to today's date, instead of the string, "today". This is because the `compareTo` is instantiating the string, and it becomes today's date (4/26/2021).

#9 - 05/14/2021 10:52 AM - Roger Borrello

The 4GL doesn't make a conversion of the "today" string to a date when performing the comparison, so `h:buffer-field('fdate'):buffer-value <> "today"`.

We have an edge case that allows this to be converted:

```
private long parseWorker(String input, byte[] dateFormat, int windowingYear)
throws ErrorConditionException
{
    if (input == null || StringHelper.safeTrimLeading(input).isEmpty() || "?".equals(input))
    {
        return INVALID_DATE;
    }
}
```

```
else if ("today".equalsIgnoreCase(input))
{
    return _today();
}
else if ("now".equalsIgnoreCase(input))
{
    invalidInitializer("NOW");
    return INVALID_DATE;
}
.
.
.
```

Because the 4GL

```
ttl.fdate = today.
```

is actually converted to a particular method

```
ttl.setFdate(date.today());
```

What is this edge case covering? I am looking through the ABL reference, and TODAY (and NOW) is not shown as a valid parameter to DATE():

A character string containing a date value to convert into a DATE data type. The string value must have the format specified by the Date Format (-d) startup parameter (the default is mdy). Note that -d sets the display format, not the date storage format, which is fixed. Furthermore, date constants entered in procedures, or as initial values in the Data Dictionary, are always specified in month/day/year format.

You do not have to specify separator characters for the month, day, and year components of the date string; however, slashes(/), periods(.), and hyphens(-) are accepted as separator characters.

I might be missing something.

#10 - 05/14/2021 10:54 AM - Constantin Asofiei

Roger, I think for the POLY case 4GL doesn't convert the right operand to the left operand's type.

We currently force this conversion, which may not be the case. If the type of the operands is not compatible (or just differs?), 4GL might fallback to comparing their string representation.

That's why we need some tests with other data types.

#11 - 05/14/2021 12:20 PM - Roger Borrello

This testcase:

```
def temp-table ttl field fdate as date format "9999-99-99".
def var h as handle.
def var tdate as date format "9999-99-99".

create ttl.
ttl.fdate = today.

h = buffer ttl:handle.
message if h:buffer-field('fdate'):buffer-value <> "" then "<>"" (success)" else "" (fail)".
message if h:buffer-field('fdate'):buffer-value <> "today" then "<>today (success)" else "=today (fail)".
message if h:buffer-field('fdate'):buffer-value = "b" then "=b (fail)" else "<>b (success)".
message if h:buffer-field('fdate'):buffer-value <> 0 then "<>0 (success)" else "=0 (fail)".
message if h:buffer-field('fdate'):buffer-value <> 0.0 then "<>0.0 (success)" else "=0.0 (fail)".
message if h:buffer-field('fdate'):buffer-value <> yes then "<>yes (fail)" else "=yes (success)".
```

Results in all the "(success)" messages on 4GL. FWD has a lot more trouble.

#12 - 05/14/2021 12:27 PM - Constantin Asofiei

Now try with int, logical, decimal, character, etc fields.

#13 - 05/14/2021 06:07 PM - Roger Borrello

Here is the completed testcase, along with the expected outcomes. Checked in as uast/compare_various_buffer-values.p.

```
def temp-table ttl
  field fdate as date format "9999-99-99"
  field fint as int
  field fch as char
  field fdec as decimal
  field flog as logical.
def var h as handle.
def var tdate as date format "9999-99-99".
```

```

create ttl.
ttl.fdate = today.
ttl.fint = 0.
ttl.fch = "".
ttl.fdec = 0.0.
ttl.flog = no.

h = buffer ttl:handle.
message if h:buffer-field('fdate'):buffer-value <> "" then "fdate<>"" (success)" else "fdate="" (fail)".
message if h:buffer-field('fint'):buffer-value <> "" then "fint<>"" (success)" else "fint="" (fail)".
message if h:buffer-field('fch'):buffer-value <> "" then "fch<>"" (fail)" else "fch="" (success)".
message if h:buffer-field('fdec'):buffer-value <> "" then "fdec<>"" (success)" else "fdec="" (fail)".
message if h:buffer-field('flog'):buffer-value <> "" then "flog<>"" (success)" else "flog="" (fail)".

message if h:buffer-field('fdate'):buffer-value <> "today" then "fdate<>today (success)" else "fdate=today (fail)".
message if h:buffer-field('fint'):buffer-value <> "today" then "fint<>today (success)" else "fint=today (fail)".
message if h:buffer-field('fch'):buffer-value <> "today" then "fch<>today (success)" else "fch=today (fail)".
message if h:buffer-field('fdec'):buffer-value <> "today" then "fdec<>today (success)" else "fdec=today (fail)".
message if h:buffer-field('flog'):buffer-value <> "today" then "flog<>today (success)" else "flog=today (fail)".

message if h:buffer-field('fdate'):buffer-value <> "b" then "fdate<>b (success)" else "fdate=b (fail)".
message if h:buffer-field('fint'):buffer-value <> "b" then "fint<>b (success)" else "fint=b (fail)".
message if h:buffer-field('fch'):buffer-value <> "b" then "fch<>b (success)" else "fch=b (fail)".
message if h:buffer-field('fdec'):buffer-value <> "b" then "fdec<>b (success)" else "fdec=b (fail)".
message if h:buffer-field('flog'):buffer-value <> "b" then "flog<>b (success)" else "flog=b (fail)".

message if h:buffer-field('fdate'):buffer-value <> 0 then "fdate<>0 (success)" else "fdate=0 (fail)".
message if h:buffer-field('fint'):buffer-value <> 0 then "fint<>0 (fail)" else "fint=0 (success)".
message if h:buffer-field('fch'):buffer-value <> 0 then "fch<>0 (success)" else "fch=0 (fail)".
message if h:buffer-field('fdec'):buffer-value <> 0 then "fdec<>0 (fail)" else "fdec=0 (success)".
message if h:buffer-field('flog'):buffer-value <> 0 then "flog<>0 (fail)" else "flog=0 (success)".

message if h:buffer-field('fdate'):buffer-value <> 0.0 then "fdate<>0.0 (success)" else "fdate=0.0 (fail)".
message if h:buffer-field('fint'):buffer-value <> 0.0 then "fint<>0.0 (fail)" else "fint=0.0 (success)".
message if h:buffer-field('fch'):buffer-value <> 0.0 then "fch<>0.0 (fail)" else "fch=0.0 (success)".
message if h:buffer-field('fdec'):buffer-value <> 0.0 then "fdec<>0.0 (fail)" else "fdec=0.0 (success)".
message if h:buffer-field('flog'):buffer-value <> 0.0 then "flog<>0.0 (fail)" else "flog=0.0 (success)".

message if h:buffer-field('fdate'):buffer-value <> yes then "fdate<>yes (fail)" else "fdate=yes (success)".
message if h:buffer-field('fint'):buffer-value <> yes then "fint<>yes (success)" else "fint=yes (fail)".
/*
** Input value: should be yes/no. (87)
Incompatible datatypes found during runtime conversion. (5729)
*/
message if h:buffer-field('fch'):buffer-value <> yes then "fch<>yes" else "fch=yes".
message if h:buffer-field('fdec'):buffer-value <> yes then "fdec<>yes (success)" else "fdec=yes (fail)".
message if h:buffer-field('flog'):buffer-value <> yes then "flog<>yes (success)" else "flog=yes (fail)".

```

There are a couple where 4GL makes a conversion that isn't a character. For example, h:buffer-field('fdate'):buffer-value <> yes turns out to be false, so today for a date equals logical yes? Very confusing.
Then the h:buffer-field('fch'):buffer-value <> yes is trying to compare "" to a logical, and complains it isn't a yes/no value.

#14 - 05/14/2021 06:18 PM - Roger Borrello

FWD complains on the very first test with Incompatible datatypes found during runtime conversion. (5729)

```
message if h:buffer-field('fdate'):buffer-value <> "" then "fdate<>"" (success) " else "fdate="" (fail)".
```

#15 - 03/22/2022 02:58 PM - Roger Borrello

- File date-5288.patch added

The attached patch is a partial fix, which hasn't been checked in. I wanted to make sure it is kept with this task.

#16 - 03/15/2023 01:26 PM - Roger Borrello

- Related to Bug #7200: fix INIT and LABEL for LIKE defined fields or variables added

Files

date-5288.patch	840 Bytes	03/22/2022	Roger Borrello
-----------------	-----------	------------	----------------