Database - Feature #6442

missing encoding/collation features

05/27/2022 01:06 AM - Eric Faulhaber

Status:	WIP	**Start date:**	
Priority:	High	**Due date:**	
Assignee:	Ovidiu Maxiniuc	**% Done:**	70%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**vendor_id:**	GCD

Description

History

#1 - 05/27/2022 01:07 AM - Eric Faulhaber

The following need to be implemented:

- DBCODEPAGE() built-in function
- DBCOLLATION() built-in function
- temp-table field option COLUMN-CODEPAGE

#2 - 11/28/2022 01:33 PM - Eric Faulhaber

- Assignee set to Igor Skornyakov

#3 - 04/05/2023 03:44 AM - Eric Faulhaber

- Assignee changed from Igor Skornyakov to Boris Schegolev

#5 - 05/16/2023 05:34 PM - Boris Schegolev

- Status changed from New to WIP

#6 - 05/17/2023 05:01 PM - Boris Schegolev

I pushed the latest version as revision 6442a/14572:

- fixed the rules so that the conversion works with the new keywords
- stubbed implementation for DBCODEPAGE and DBCOLLATION

#7 - 05/18/2023 10:03 AM - Boris Schegolev

Just a quick question about design: are collation/codepage available as some database config (like in ConnectionManager.ConnectInfo) or should I just deduce/read this information from JDBC?

#8 - 05/24/2023 03:23 PM - Eric Faulhaber

Boris Schegolev wrote:

> Just a quick question about design: are collation/codepage available as some database config (like in ConnectionManager.ConnectInfo) or should I just deduce/read this information from JDBC?

Ovidiu, I have not had a chance to dig into this. Do you have any suggestions? I suspect JDBC will not provide enough information.

#9 - 05/24/2023 08:57 PM - Ovidiu Maxiniuc

Me either.

Currently the customer is free to pick the encoding/collation he desires for SQL database, based on the set the respective dialect supports. There is currently no method to extract these information from SQL. The problem is that each SQL engine has its own way of setting it and the resources for doing this are different: H2 uses the FwdCollatorProvider (we have complete control here), PSQL uses the locale we provide when the cluster is created, and in MSSQL and MariaDb case the user must chose from a large but predefined list.

I also am circumspect about the chances of getting the information using JDBC, exactly because of this diversity. Talking about encoding, FWD communicates with the JDBC driver using Java native String objects, but on the other side is really implementation specific. For collation, I am really curious, too.

#10 - 05/26/2023 01:56 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

> There is currently no method to extract these information from SQL. The problem is that each SQL engine has its own way of setting it and the resources for doing this are different: H2 uses the FwdCollatorProvider (we have complete control here), PSQL uses the locale we provide when the cluster is created, and in MSSQL and MariaDb case the user must chose from a large but predefined list.

This suggests we need to leverage the Dialect hierarchy to help determine this information in a dialect-specific and OE-compatible way.

#11 - 05/26/2023 02:32 PM - Boris Schegolev

That's doable. I will look into that.

But maybe first a more general question: where does the collation information come from? Is it something that can be changed in runtime (I know it can be done with SQL, but I am not sure whether 4GL allows that) or is it defined beforehand?

#12 - 05/26/2023 04:15 PM - Eric Faulhaber

AFAIK, in OE, collation tables are referenced from a configuration file, convmap.dat, which the DBA sets up (I don't know how). This file refers to names of predefined collation tables. I think a default collation is set up for a given database, and there is an AVM startup parameter (-cpcoll) which can override this default.

The OE doc for DBCOLLATION() states that the name returned...

> ...corresponds to the definition of the collating sequence contained in the convmap.dat file, which usually resides in the $DLC directory.

There is no such file in the FWD implementation. As Ovidiu notes, we use different means to support collation, depending on the database dialect. Currently, collation is set for the entire database; we don't support overrides at the table or column level.

We do support using a specific code page for a temp-table CLOB field (see RecordBuffer.getCodePage()). This is set when the temp-table is defined

(see the doc for the DEFINE TEMP-TABLE statement). As I dug into the code to see how we would implement the temp-table COLUMN-CODEPAGE option, it seems we already have done that, but the gap markings were not updated to reflect the support. Igor, Ovidiu, do you know of anything left to implement for the COLUMN-CODEPAGE option?

I'm not sure what the implementation of the DBCODEPAGE() function needs to look like. AFAIK, the name returned by this function corresponds with another file set up by an OE DBA, convmap.cp. You'll need to dig into the OE documentation and do some experimenting to figure this out.

#13 - 05/26/2023 04:58 PM - Greg Shah

The AVM -cpcoll sets the 4GL language runtime collation. It may affect temp-tables but not the database.

My understanding is that when you initialize the database, you do it by copying one of the empty databases provided with OE, each one of those has a very specific encoding. I don't recall if this can be changed using a utility.

Marian?

#14 - 05/26/2023 06:18 PM - Ovidiu Maxiniuc

Eric Faulhaber wrote:

> As I dug into the code to see how we would implement the temp-table COLUMN-CODEPAGE option, it seems we already have done that, but the gap markings were not updated to reflect the support. Igor, Ovidiu, do you know of anything left to implement for the COLUMN-CODEPAGE option?

Yes, I remember I added implementation of IS-COLUMN-CODEPAGE and some related methods when working with LOBs and CPs and probably I forgot to update the gap markings. I wrote a short test to check that out:

```
DEFINE TEMP-TABLE cp-test NO-UNDO
   FIELD f1 AS CLOB COLUMN-CODEPAGE "utf-16"
   FIELD f2 AS CLOB TTCODEPAGE
   FIELD f4 AS CHARACTER COLUMN-CODEPAGE "utf-16"
   FIELD f5 AS CHARACTER TTCODEPAGE.

CREATE cp-test.

MESSAGE IS-COLUMN-CODEPAGE(cp-test.f1)
        IS-COLUMN-CODEPAGE(cp-test.f2)
        IS-COLUMN-CODEPAGE(cp-test.f4)  // err
        IS-COLUMN-CODEPAGE(cp-test.f5). // err

MESSAGE GET-CODEPAGE(cp-test.f1)
        GET-CODEPAGE(cp-test.f2)
        GET-CODEPAGE(cp-test.f4)  // err
        GET-CODEPAGE(cp-test.f5). // err
```

There is a problem I identified with this occasion: when the IS-COLUMN-CODEPAGE is invoked statically (as in my example above) the compiler knows the type and will report the lines marked with err with error 223 and 196. Apparently, I worked only with dynamic case where 5729 error is raised. I guess this must be fixed as part of this task.

IIRC, Marian has provided some tests for these features on xfer server.

Boris, is there a branch for this task where I could update the gap markings?

#15 - 05/27/2023 08:52 AM - Marian Edu

Greg Shah wrote:

> The AVM -cpcoll sets the 4GL language runtime collation. It may affect temp-tables but not the database.

It will definitively not affect the database per-se, as in the indexes will remain as is and all queries that uses the indexes will render the same results as before... however if the order requested (using BY) isn't resolved by the server but locally by the client (aka, there is no index that can be used) then the order of records might be different based on the client -cpcoll startup parameter.

It worth nothing this will also affect the result of COMPARE and the use of COLLATE option in queries - the latter can be used to override the order in database queries that does not need client sort because the index potentially have the records in the proper order. For TEMP-TABLE the indexes use the runtime client connection, for databases the index uses the database collation so the result might be different, by using COLLATE the sort will be done on the client-side after record selection based on specified collation. I guess this can also be used for temp-tables to sort using a different collation than that of -cpcoll but I've never tried something like that :(

> My understanding is that when you initialize the database, you do it by copying one of the empty databases provided with OE, each one of those has a very specific encoding.

True, unless you start from a 'template' one has to load the collation table.

> I don't recall if this can be changed using a utility.

No, there is no utility for this - the only way around is reload the new collation table and then rebuild the indexes.

#16 - 05/29/2023 11:13 AM - Ovidiu Maxiniuc

Marian,

Let's assume we have a field t1.f1 in a permanent table/database with a configuration where b sorts higher than a. At the same time, we launch the client using -cpcoll with a collation where a sorts higher than b. (Note that a and b are generic here; in real life, they might be variant of accented a, or other special symbols). How will the following queries behave?

- for each t1 where t1.f1 > 'a'
- for each t1 where t1.f1 > 'a' COLLATE 'collation1'
- for each t1 where t1.f1 > 'a' COLLATE 'collation2'

#17 - 05/29/2023 04:42 PM - Boris Schegolev

Ovidiu Maxiniuc wrote:

> Boris, is there a branch for this task where I could update the gap markings?

There's a 6442a branch for this task. I rebased it to be in sync with current trunk.

#18 - 05/29/2023 06:43 PM - Ovidiu Maxiniuc

Thank you. I committed the gap markings for CODEPAGE-related options/functions as revision 14593.

#19 - 05/30/2023 02:39 AM - Marian Edu

Ovidiu Maxiniuc wrote:

> Marian,
>
> Let's assume we have a field t1.f1 in a permanent table/database with a configuration where b sorts higher than a. At the same time, we launch the client using -cpcoll with a collation where a sorts higher than b. (Note that a and b are generic here; in real life, they might be variant of accented a, or other special symbols). How will the following queries behave?

We need to write tests to confirm that of course but the expected results, at least my expectation, will be:

- for each t1 where t1.f1 > 'a'

Rows with b values will be filtered out wether or not there is an index for that field since the record selections is done on the server side (since version 7, not sure if there are any clients left on Progress V6 where records were sent to the client for selection). There are cases still when using some functions will still force the server to send the records to the client for selection but this is not the case (CAN-FIND).

- for each t1 where t1.f1 > 'a' COLLATE 'collation1'

This is not valid syntax, COLLATE only works with BY so for sorting. Assuming we add the BY keyword it still doesn't change a thing since sorting comes after record selection so rows with b will be filtered out.

- for each t1 where t1.f1 > 'a' COLLATE 'collation2'

Same as before, however say we have b and c that both ranks higher that a in both collations but their order is different in those two collations then the order of records will be different depending on the situation:
- no BY option, no index for that field then will come out in the order of primary index for the table
- no BY option with an index on the field, the records will be sorted using the database collation so b and c will be sorted based on their ranking in that collation used by the database
- BY option used on the field, same as previously
- BY option with COLLATE the records will be sorted based on the ranking provided by the specified collation (client-side)

#20 - 05/30/2023 02:41 AM - Marian Edu

Ovidiu Maxiniuc wrote:

- for each t1 where t1.f1 > 'a'

Just realised > is used for where so both a and b will be filtered out :)

#21 - 09/13/2023 08:12 AM - Greg Shah

- Assignee deleted (Boris Schegolev)

#22 - 12/12/2023 03:07 PM - Eric Faulhaber

- Priority changed from Normal to High

- Assignee set to Ovidiu Maxiniuc

#23 - 12/21/2023 12:39 PM - Ovidiu Maxiniuc

- % Done changed from 0 to 70

The DBCODEPAGE() and DBCOLLATION() built-in function were implemented in #7326 and available in trunk starting with r14896.

I will investigate the existing support for remaining temp-table field option COLUMN-CODEPAGE.