Database - Feature #5491

custom locale/collation implementations

06/25/2021 10:35 AM - Eric Faulhaber

Status:	WIP		**Start date:**	
Priority:	Normal		**Due date:**	
Assignee:	Ovidiu Maxiniuc		**% Done:**	100%
Category:			**Estimated time:**	0.00 hour
Target version:				
billable:	No		**version:**	
vendor_id:	GCD			

Description

Related issues:	
Related to Database - Support #5183: database collation	**WIP**
Related to Database - Support #4550: eliminate the need to place the p2jspi.j...	**Closed**
Related to Runtime Infrastructure - Support #5567: move FWD to Java 11	**Closed**

History

#1 - 06/25/2021 10:53 AM - Eric Faulhaber

The locale/gen-locale.p program has recently been updated and several new OS-level custom locales have been generated, to be used by PostgreSQL to provide legacy-compatible collation support for different combinations of language and character set.

We need similar implementations for use by H2, for temp-table and metadata support. Currently, we have only p2j.spi.ISO_8859_1Collator, which was designed to match the original (now deprecated) en_US@p2j_basic OS locale. This Java locale should be updated to match the rules of the newer en_US@ISO8859-1@FWD OS locale, and other, similar subclasses of java.text.RuleBasedCollator need to be created as companions of the other, new, OS-level, custom locales.

See:

- the com.goldencode.p2j.spi package
- java.text.spi.CollatorProvider and p2jspi.mf.template in the manifest directory

As new Java collators are added, we will need to update the p2jspi.jar file installed in the Java extension (jre/lib/ext/) directory of project installations.

#2 - 06/25/2021 10:53 AM - Eric Faulhaber

- Related to Support #5183: database collation added

#3 - 06/25/2021 07:22 PM - Ovidiu Maxiniuc

At this moment ISO_8859_1Collator is the only collation registered with the 'identifier' new Locale("en", "US", "P2J"). To accommodate the new locales/collations we need new naming schema. I suggest the following names for the three of them:

- new Locale("en", "US", "ISO8859-1@FWD"),
- new Locale("en", "US", "ISO8859-15@FWD") and
- new Locale("en", "US", "CP1252@FWD").

The language and country parameters are not relevant (for what I know) so we can keep it simple and use only "en", and "US", respectively. The

variant has to contain the character encoding and a reference to application. The desired locales/collations have to be set/updated in p2j.cfg.xml and directory.xml for h2 dialect.

I think we should keep the old Locale associated to same ISO_8859_1Collator although if the configuration files are not updated we might have a hard time understanding why sorting is not as expected.

#4 - 06/26/2021 05:51 PM - Ovidiu Maxiniuc

Committed revision 12576.

Added ISO_8859_15Collator.java and CP1252Collator.java identified as described above. To load them at runtime, use:

```
            <node class="string" name="embedded-collation">
              <node-attribute name="value" value="en_US_CP1252@FWD" />
```

in the container definition of your database. The static configuration is:

```
        <dialect-specific name="h2">
           <parameter name="collation" value="en_US_ISO8859-15@FWD" />
        </dialect-specific>
```

in the namespace of the respective database.

Two things to note:

- the CP1252 charset is really strict on Java. Since the codepoints 0x81, 0x8D, 0x8F, 0x90 and 0x9D are not defined for this CP and all of them will be converted to \uFFFD (REPLACEMENT-CHARACTER) causing the RuleBasedCollator c'tor to fail with java.text.ParseException at server start-up so they were disabled. However, when encountered in a order clause, these characters will be sorted last. Strangely, the other two CPs have more undefined codepoints (0x7f - 0x9f), but no exception is thrown when processing the Collator;
- three case conversion have drawn my attention because they differ from what I see in 4GL:
  - ƒ (0x83) (not lowercase F - f (0x66)) is converted to uppercase ? (0x3f, question mark),
  - µ (0xB5) is converted to same question mark,
  - ß (0xDF) is converted to SS which actually is a two character string (this looks correct from POV of German language) so ASC function returns -1.

Later edit:
I noticed some other peculiarities:

- in Java 0xd5 (Õ) converts to lowercase 0xf5 (õ) but not in 4GL. This looks correct. The inverse operation is present for both Java and 4GL. I think this is a flaw in 4GL.
- also for Ö (0xD6) to ö (0xF6)
- Ø (0xD8) to ø (0xF8) and
- Ù (0xD9) to ù (0xF9).

#5 - 06/28/2021 07:07 AM - Greg Shah

The language and country parameters are not relevant (for what I know)

Are you saying that we've never found any differences in collation which are due to just language or country? In other words, the 4GL does not honor any differences by language or country, all collation for a given codepage is the same across languages and countries?

#6 - 06/28/2021 10:10 AM - Ovidiu Maxiniuc

Indeed, at first I thought that using different CPs will automatically imply a specific sorting order since I did not see any difference on Windows when I tried launching the extraction procedure with different (input) languages set - same files were generated. I did not test on Linux.

This looked normal. If somebody wants to deploy an application for a specific language he will use the CP which covers the character of the respective language. These characters will be sorted in a specific way. For example, a customer may chosen 1252 for Dutch language. A Russian customer will chose 1251 because it contains the Cyrillic letters.

However, this is not enough. Choosing the internal/stream CP is necessary in order to have the character properly printed on screen. For files is not as mandatory as the text files do not have a BOM which explicitly configures the CP so it's more like a convention for the editor to know which to use to display the stream properly. The key here is the CPCOLL. The default one is basic. I attempted to load different languages but, for the majority the procedure refuses to start because of the error 1043 (ex: Dutch, Spanish, German, French, Romanian). However, I managed to load CPCOLL only for Russian and Finnish. I will review my recent additions to see if the output is influenced by this specific parameter and, if necessary, I will update the collation files.

#7 - 06/28/2021 01:36 PM - Ovidiu Maxiniuc

I investigated the content of the convmap.cp on my Windows test machine and I found the following collations available for CP we are interested:

en_US, en_GB, nl_NL all seem to use the default (basic) collation so no changes are necessary in the current Linux locale scripts.

#8 - 06/28/2021 02:05 PM - Greg Shah

Then perhaps we should not have en_US@CP1252@FWD, en_GB@CP1252@FWD and nl_NL@CP1252@FWD but instead just basic@CP1252@FWD?

In the future, when we add support for customer using one of the non-basic CPCOLL, we could add those variants using the language and country instead of the basic prefix.

#9 - 06/28/2021 02:20 PM - Ovidiu Maxiniuc

It is a bit more complicated than that.
We need to keep the lang/land prefix. This is used within the locale script for configuring the other LC_ categories (LC_TIME, LC_NUMERIC, LC_MONETARY, LC_MESSAGES, etc).
It seems logical to add this token of the name of the locale files. This means their names will become increasingly complex.

#10 - 06/28/2021 02:47 PM - Greg Shah

OK, understood.

Then shouldn't we mirror the same approach for the p2jspi.jar? This may avoid confusion.

#11 - 06/28/2021 03:20 PM - Ovidiu Maxiniuc

- Status changed from New to WIP

Yes. They would be identified by:

PostgreSQL	en_US@CP1252@basic@FWD, en_GB@CP1252@basic@FWD and nl_NL@CP1252@basic@FWD	same character encoder / collation but different languages / LC_MONETARY / LC_TIME / etc, specific to each country
H2	new Locale("en", "US", " ISO8859-1@basic@FWD"), new Locale("en", "US", " ISO8859-15@basic@FWD") and new Locale("en", "US", " CP1252@basic@FWD")	same language/territory/collation but different character encoding

#12 - 06/29/2021 04:14 PM - Eric Faulhaber

- Tracker changed from Bug to Feature

Ovidiu, please update the FWD documentation with the new approach to installation and configuration. Specifically, these areas:

- Install a Custom Locale
- Create a FWD-Specific Database Cluster
- any place that mentions en_US@p2j_basic

#13 - 06/29/2021 09:15 PM - Ovidiu Maxiniuc

- Assignee set to Eric Faulhaber

- vendor_id deleted (GCD)

Eric Faulhaber wrote:

> Ovidiu, please update the FWD documentation with the new approach to installation and configuration. Specifically, these areas:
>
> - Install a Custom Locale
> - Create a FWD-Specific Database Cluster
> - any place that mentions en_US@p2j_basic

Done. I think all occurrences of the old, fixed locale were changed. In some places I tried to explain shortly the naming schema of the locales. I left untouched Chapter 3: Development Environment Setup and Chapter 4: Advanced Setup from Developer Guide as these chapters expressly mention in their header that the information contained there is no longer maintained and may not be accurate. Since they are kept for historical purposes, the old locale is the right one.

#14 - 06/29/2021 09:20 PM - Ovidiu Maxiniuc

- vendor_id set to GCD

- Assignee changed from Eric Faulhaber to Ovidiu Maxiniuc

Sorry, I do not know how these task properties has changed. I did not do it on purpose :(.

#15 - 07/05/2021 04:20 PM - Ovidiu Maxiniuc

- Status changed from WIP to Review

- % Done changed from 0 to 100

Do we keep this task open for adding new collations?

#16 - 07/05/2021 04:26 PM - Eric Faulhaber

No, I'll close it after review of this current batch. As we create more locales/collators over time, we can reopen it as needed.

#17 - 07/05/2021 04:59 PM - Eric Faulhaber

Please change the title in the LC_IDENTIFICATION section of the en_GB locale definitions to say, "Great Britain" instead of "the Britain".

#18 - 07/06/2021 09:10 PM - Ovidiu Maxiniuc

Done. Committed revision 12640.
I also updated the extractor procedure to create new files with fixed header/title and to extract the SESSION:CPCOLL, too.

#19 - 07/07/2021 03:44 PM - Eric Faulhaber

Code review 3821c/12640:

The changes in this revision look good.

I took another look at gen-locale.p and I did find one thing. Even though it did not change in this revision, I didn't pick up on it earlier.

When we are generating the collation portion of the locale definition, we rely on the primary index of the tt-latin1 table being chosen through normal index selection (btw, we should probably change this table name to something less specific, since it implies ISO-8859-1 encoding is always in use, and this may be confusing to readers). The primary index is defined as:

```
    INDEX pidx AS PRIMARY f2
```

where f2 represents the case-sensitive character. The loop which generates the collation data is:

```
MESSAGE "<U0000>". // null is always first
FOR EACH tt-latin1:
    MESSAGE f(f1). // "%" f3  + ' ' + UPPER(f3) + ' ' + LOWER(f3).
END.
```

I think here we should be more explicit about the sorting. Please see my note #5183-8, specifically the last 3 paragraphs. Then review #5183-18, specifically the introduction about how the 4GL output is sorted. Note the secondary sort by codepoint. This is done to make the sort deterministic, when the 4GL weighs certain characters the same for collation purposes. I think by chance we are getting pretty consistent results with the tt-latin1 table's primary index, but I think this is only due to sequential rowids assigned at record creation. Since this is not documented/guaranteed, I think we should either add a secondary sort field to the primary index, as in:

```
    INDEX pidx AS PRIMARY f2 f1
```

...and specify USE-INDEX pidx to make the sort intent absolutely clear, OR explicitly specify the sort of the FOR EACH statement:

```
MESSAGE "<U0000>". // null is always first
FOR EACH tt-latin1 BY f2 BY f1:
    MESSAGE f(f1). // "%" f3  + ' ' + UPPER(f3) + ' ' + LOWER(f3).
END.
```

#20 - 07/14/2021 06:53 PM - Eric Faulhaber

- Status changed from Review to WIP

We need to simplify the configuration of the locale/collation in use for databases, including the _temp database. I would like to apply sensible defaults, so that these don't need to be configured explicitly at all in most cases.

The collation and code page in use for the _temp database currently are driven by the embedded-collation setting in the directory. Other than for this setting, the inclusion of a section for the temp database in the directory is unnecessary in most cases. I would prefer to derive the correct setting at server startup, but I'm not sure yet how we would do this. In the 4GL, the code page, collation, and country are settable through startup parameters and/or runtime attributes. Do we already have comparable settings in the directory (or programmatic defaults) for any of these for any other purpose? If we can detect these settings when the server starts up, we could compose a default value for the embedded-collation setting, avoiding a need for this as an explicit setting in the directory. Or perhaps we need to leave it as it is to enable an overriding setting...I am open to suggestions.

For persistent databases, we need the collation used for the primary database to be consistent with the Java Collator in use for the embedded H2 database used for metadata and for the dirty database. These must sort the same way, or we may have unpredictable behavior. When PostgreSQL is used, the collation behavior for a primary database is picked up by the database server from the OS-level locale on which the database cluster is built. It can be queried with select datcollate from pg_database where datname = '<database_name>'. For example, for a database in a cluster built on top of the en_US.iso885915 locale:

```
postgres=# select datcollate from pg_database where datname = 'postgres';
   datcollate
-----------------
 en_US.iso885915
(1 row)
```

This query (or whatever is appropriate for other dialects) would be encoded into the database-specific Dialect implementation.

The value returned would be the default for the embedded-collation setting for that database, obviating the need to set it in the directory. If we can use the same name to represent the custom locale used by PostgreSQL and the custom Collator used by Java/H2, that would make things easiest. Otherwise, we will have to implement a mapping between the information returned by the dialect-specific query and needed by the H2 set collation ... statement. The only use case I can think of where we would need to override this would be to allow a user to set a value that is not one of the collations FWD provides "out of the box"; i.e., if a user had developed their own custom locale and Collator implementation.

In any event, whether we implement all this or not, we need to update the FWD v4 documentation to reflect the new H2 Collator specifiers that are supported (as en_US_P2J is now obsolete).

#21 - 07/14/2021 06:58 PM - Eric Faulhaber

Eric Faulhaber wrote:

> ...
> Do we already have comparable settings in the directory (or programmatic defaults) for any of these for any other purpose?

From Greg via email:

EnvironmentOps:

```
   /**
    * Retrieves the session-specific current language.
    *
    * @return   The name of the current language.
    */
   public static character getCurrentLanguage()
   {
      return new character(work.obtain().currentLanguage);
   }
...
      /** Stores the current-language of the session. */
      private String currentLanguage = Utils.getDirectoryNodeString(null, "currentLanguage", "?");
```

#22 - 07/14/2021 07:21 PM - Eric Faulhaber

I assume that setting would be in the form <language>_<COUNTRY>, e.g., en_US, en_GB, etc. To compose an embedded-collation value which could be used with the set collation statement for H2, we would still need to query collation and code page. Looks like we have that in I18nOps (getCPColl() and getCPInternal(), respectively).

I just realized that we also should be forcing private-temp-dbs to create an individual _temp database for each session, since the language, code page, and collation can vary by session, and set collation works database-wide AFAIK. Currently, embedded-collation forces the same settings for all temp-tables.

We may need to modify the code which initializes databases to defer the temp-table database initialization until after a session starts. I think currently, some level of initialization still occurs at server startup.

#23 - 07/15/2021 11:01 AM - Ovidiu Maxiniuc

From #5034:

The SPI only provides a sort order. The language and country are probably irrelevant here, but they are required as parameters for collator API. We can dropped the default basic token (it was only added recently) but it might be needed if another CPCOLL is added (see #5491-7). I wonder if we can map this to country/language instead? The application specific marker can also be dropped. The only issue is to be sure we do not have collisions with the other provided collations (about 600 of them on my system). If we simplify the naming for H2 enough I think we could keep the PG naming as it is now since they are different and will not make such confusion.

#24 - 07/15/2021 11:22 AM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

> The SPI only provides a sort order. The language and country are probably irrelevant here, but they are required as parameters for collator API. We can dropped the default basic token (it was only added recently) but it might be needed if another CPCOLL is added (see #5491-7). I wonder if we can map this to country/language instead? The application specific marker can also be dropped. The only issue is to be sure we do not have collisions with the other provided collations (about 600 of them on my system). If we simplify the naming for H2 enough I think we could keep the PG naming as it is now since they are different and will not make such confusion.

As noted in my entries above, I'd like to use the same name for the OS locale and for the embedded-collation setting, if possible (well, and get rid of the embedded-collation setting externally, as noted above).

I think we should keep basic as part of the name, and I don't want to repurpose the country/language for this. I think that would just serve to confuse things more. If by "application marker", you mean the "FWD" part, I think we need to keep this as a differentiator (though I'd like to lowercase it), so we are sure not to clash with the many other names (as you say, 600 on your system).

Can we go with a convention like

```
<lang>_<COUNTRY>.<codepage>_fwd_<collation>
```

(e.g., en_US.iso88591_fwd_basic)? The names in /usr/share/i18n/SUPPORTED are all uppercase with hyphens, but these seem to go to lowercase and drop the hyphens (except C.UTF-8 for some reason) when compiled. I don't see other compiled locales on my system with underscores in the second part of the name. We could maybe go with hyphens in this part instead, however smashing it all together, like en_US.iso88591fwdbasic seems a little strange.

Would this convention work for H2 as well?

H2 identifies the locale using its name this way:

```
    static boolean compareLocaleNames(Locale locale, String name) {
        return name.equalsIgnoreCase(locale.toString()) ||
                name.equalsIgnoreCase(getName(locale));
    }
```

OTOH,
The java.util.Locale.toString() uses the underscore (_) to compose the result:

```
public final String toString() {
[...]
    StringBuilder result = new StringBuilder(baseLocale.getLanguage());
[...]
        result.append('_')
            .append(baseLocale.getRegion()); // This may just append '_'
[...]
        result.append('_')
            .append(baseLocale.getVariant());
[...]
```

So we are stuck with underscore. Beside that, the . is used with other syntactic meaning in SQL (it has a special type CHAR_DOT) so it cannot be used here.

I also tried dropping the hyphen (like iso88591) last night. It looks strange, that's why I used _ instead. Because the Locale lookup is done by checking the name additively, multiple _ are not reported as errors. This is why Eugenie successfully did the import.

I think we should stick to <lang>_<COUNTRY>_<codepage>_fwd_<4gl-collation>. This seems to fit in both worlds.

#26 - 07/15/2021 12:35 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

> This seems to fit in both worlds.

I don't think it really fits in the Linux world. There may be no technical impediment, but I've never seen underscore used to separate the country and variant in a locale name. It is usually the dot or the @ symbol. I would prefer to go with the dot (and replace this with underscore internally as needed for H2).

Is there any reason we need to expose the embedded_collation variant of this name to the outside world? I've discussed above how I think we can eliminate it from the runtime configuration (i.e., the directory). We also have this name in the dialect-specific section for H2 in p2j.cfg.xml. Do you recall what that is used for?

#27 - 07/15/2021 01:30 PM - Ovidiu Maxiniuc

Eric Faulhaber wrote:

> Do you recall what that is used for?

According to bzr, it was added in 2010, with the *"support for a permanent DB backed by H2."* A parameter at this location is only used for two reasons:

- a default value if not specified in directory, but I do not think this is the reason;
- a configuration for opening a connection to database when running in special mode:
    1. the import. However, when the create-table DDLs are generated the collation is the first thing written. So this is not the case;
    2. the **conversion**. Use this parameter to generate the first line in DDL which declares the collation. This is certainly the only usage of the parameter. This is a bit strange, nonetheless, because we can instantiate multiple database from same schema but having different collations.

> I would prefer to go with the dot (and replace this with underscore internally as needed for H2).

I did not think about it that way, but it makes sense. We can use the @, too in this case. We can replace both characters. I vote for this.

#28 - 07/15/2021 02:16 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

> I did not think about it that way, but it makes sense. We can use the @, too in this case. We can replace both characters. I vote for this.

For some reason, I really don't like the @ symbol in the name (yes, I know I added it originally), but if it is important to you or you have some standards-based reason for wanting it, we can use it :-)

If we need to keep the collation setting in p2j.cfg.xml, we can accept either form. I think people will find it more intuitive to use the same form as would be used for PostgreSQL, so maybe that's the officially supported and documented form.

Constantin, please see the previous entry about the collation setting in p2j.cfg.xml. This was added in your rev 9939 commit. I know it was a while ago, but can you shed any light on the purpose?

#29 - 07/15/2021 02:18 PM - Eric Faulhaber

Eric Faulhaber wrote:

> If we need to keep the collation setting in p2j.cfg.xml, we can accept either form. I think people will find it more intuitive to use the same form as would be used for PostgreSQL, so maybe that's the officially supported and documented form.

I should clarify: here I mean the OS locale name vs. the H2-friendly name, not underscore vs. @.

#30 - 07/15/2021 02:31 PM - Constantin Asofiei

Do you mean this in p2j.cfg.xml?

```
    <dialect-specific name="h2">
      <parameter name="collation" value="en_US_P2J" />
    </dialect-specific>
```

The reason was to emit the set collation for H2 database create table sql file (before anything else, first line in the file).

How else can you set the H2 database collation? There is no cluster to prepare it, is created 'on request' when it is first accessed.

#31 - 07/15/2021 03:38 PM - Constantin Asofiei

Constantin Asofiei wrote:

> Do you mean this in p2j.cfg.xml?
> [...]

> The reason was to emit the set collation for H2 database create table sql file (before anything else, first line in the file).

See here for the constraints of setting the H2 collation: https://www.h2database.com/html/commands.html#set_collation

#32 - 07/15/2021 04:02 PM - Eric Faulhaber

OK, Ovidiu, so it sounds like we need to keep this setting in p2j.cfg.xml. But for consistency from the user's perspective, let's document it to be the same convention as the OS-level locale name; e.g., en_US.iso88591_fwd_basic or en_US.iso88591@fwd@basic (whichever way you decide to go) and preprocess it to replace the characters H2 can't parse with underscores before executing the set collation statement.

#33 - 07/15/2021 04:03 PM - Eric Faulhaber

Eric Faulhaber wrote:

> OK, Ovidiu, so it sounds like we need to keep this setting in p2j.cfg.xml. But for consistency from the user's perspective, let's document it to be the same convention as the OS-level locale name; e.g., en_US.iso88591_fwd_basic or en_US.iso88591@fwd@basic (whichever way you decide to go) and preprocess it to replace the characters H2 can't parse with underscores before executing the set collation statement.

Is everyone OK with this approach? Speak now or don't complain later ;-)

#34 - 07/15/2021 04:36 PM - Greg Shah

I prefer the underscores.

#35 - 07/15/2021 04:46 PM - Ovidiu Maxiniuc

en_US.iso88591_fwd_basic will be then.

#36 - 07/17/2021 09:04 AM - Ovidiu Maxiniuc

The H2/Java collation names and Linux/PostgreSQL locale names were unified as described above.
Also, I think I updated all the wiki pages related to this issue. Please let me know if you see inconsistencies.

Committed revision 12690.

#37 - 07/20/2021 08:10 AM - Eugenie Lyzenko

The strange thing is happening:
1. I have both system locales installed, nl_NL@CP1252@basic@FWD and nl_NL.cp1252_fwd_basic. Both are based on the same CP1252.
2. But I can not create PG cluster for nl_NL.cp1252_fwd_basic while for nl_NL@CP1252@basic@FWD it is created without problems.

Both locales can not co-exist on system simultaneously?
PG 9.5 does not like some chars in nl_NL.cp1252_fwd_basic locale name?

Can I use nl_NL@CP1252@basic@FWD locale for internal PG needs and work with new nl_NL.cp1252_fwd_basic one defined in FWD? They should actually be the same I think.

#38 - 07/20/2021 08:21 AM - Greg Shah

Isn't the . a problem in SQL?

#39 - 07/20/2021 09:15 AM - Eugenie Lyzenko

Greg Shah wrote:

> Isn't the . a problem in SQL?

I think the pg_createcluster just drops everything after . in locale name. And because I really do not have pure nl_NL locale installed the command fails.

#40 - 07/20/2021 09:38 AM - Greg Shah

Ovidiu: The documentation seems to be incorrect. I think Eugenie is doing exact what is listed but it does not work.

#41 - 07/20/2021 09:57 AM - Ovidiu Maxiniuc

I am also investigating the issue. But I could not find a reason what is wrong.

I edited edit /etc/locale.gen and /usr/local/share/i18n/SUPPORTED and executed locale-gen.

I managed to create a cluster using:

```
sudo -u postgres pg_createcluster -p 5444 9.5 vanm2 --locale=nl_NL.cp1252fwdbasic --encoding=win1252 -d /mnt/s
ec/pd_data/9.5/vanm2/
```

Note that the DOT is accepted, instead the UNDERSCORES are dropped.

On Saturday it seemed to work for me. Something started going wrong yesterday.

#42 - 07/20/2021 10:06 AM - Eugenie Lyzenko

Do we have to use different locale name formats for PG and H2? Why not to use some simpler name, say nl_NL.cp1252? This can work for both DB types.

#43 - 07/20/2021 10:44 AM - Ovidiu Maxiniuc

Things are a bit more complicated:
The dot (.) cannot be used in H2. It is replaced in the background with _.
Then there are multiple collations for the same code page. For the moment we use only the default (basic) but other will probably be added so we need to be prepared.
And last, the fwd token is a marker for application, to avoid conflicts with other resources a user may have installed on his station.

#44 - 07/20/2021 11:57 AM - Ovidiu Maxiniuc

It works again for me:

```
om@ProDesk400G3MT:~$ psql -U fwd_user -h localhost -p 5444 -l
Password for user fwd_user:
                                    List of databases
   Name    |  Owner   | Encoding |        Collate        |         Ctype         |   Access privileges
-----------+----------+----------+-----------------------+-----------------------+----------------------
 postgres  | postgres | WIN1252  | nl_NL.cp1252_fwd_basic | nl_NL.cp1252_fwd_basic |
 template0 | postgres | WIN1252  | nl_NL.cp1252_fwd_basic | nl_NL.cp1252_fwd_basic | =c/postgres          +
           |          |          |                       |                       | postgres=CTc/postgres
 template1 | postgres | WIN1252  | nl_NL.cp1252_fwd_basic | nl_NL.cp1252_fwd_basic | =c/postgres          +
           |          |          |                       |                       | postgres=CTc/postgres
(3 rows)
```

I need to do a clean-up and recreate it all from the 1st step and post the result.

I am a bit puzzled now. Here are my installed locales:

```
om@ProDesk400G3MT:~$ locale -a
C
C.UTF-8
POSIX
[...]
en_US.cp1252
en_US.cp1252@p2j_basic.CP1252
en_US.iso88591
en_US.iso885915
en_US.iso88591@p2j_basic.latin1
en_US.utf8
en_US@p2j_basic.CP1252
en_US@p2j_basic.latin1
[...]
nl_NL
nl_NL.cp1252
nl_NL.cp1252_fwd_basic
nl_NL.cp1252fwdbasic
nl_NL.iso88591
nl_NL.iso885915
```

I had this in the morning:

```
om@ProDesk400G3MT:~$ locale -a
C
C.UTF-8
POSIX
```

```
en_CA.utf8
en_DK.utf8
en_GB
en_GB.iso88591
en_GB.iso88591_fwd_basic
en_GB.utf8
en_US.cp1252_fwd_basic
en_US.iso885915_fwd_basic
en_US.iso88591_fwd_basic
en_US.utf8
en_US.utf8_fwd_basic
en_US@p2j_basic
nl_NL.cp1252_fwd_basic
nl_NL@cp1252@fwd@basic
```

I do not know why the _fwd_basic disappeared. They are no more in my /usr/lib/locale/. They were there in the morning, I did not delete them.
I do not understand (yet) the presence of those containing @p2j_basic in their name. Must be a different location.

#45 - 07/20/2021 12:16 PM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

> It works again for me:
> [...]
>
> I need to do a clean-up and recreate it all from the 1st step and post the result.

Please clarify what do you mean by "1st step" above?

Sometimes regular system update re-generate default locales so some custom additions become deleted during the process.

And not clear what locale from your nl_NL* list was used when you started pg_createcluster to verify the locale is valid for command pg_createcluster.
It can be other than nl_NL.cp1252_fwd_basic.

Can you leave only nl_NL.cp1252_fwd_basic for locale -a output and recreate the cluster again?

#46 - 07/20/2021 12:20 PM - Eric Faulhaber

To be clear: all the FWD-related custom locales with p2j_basic or @ in the names are obsolete and should no longer be generated nor reported by locale -a when this is all working as intended. Eugenie, don't create any clusters based on these obsolete locales.

#47 - 07/20/2021 12:26 PM - Ovidiu Maxiniuc

Eugenie Lyzenko wrote:

> Please clarify what do you mean by "1st step" above?

I mean to do the entire process on a clean system, exactly like a new customer would do it. I did not reinstalled my system since 2014 and only upgraded incrementally. I might have a lot of garbage. I cannot do now a clean install, but I can try a VM.

> Sometimes regular system update re-generate default locales so some custom additions become deleted during the process.

I am thinking that this might be the problem.

> And not clear what locale from your nl_NL* list was used when you started pg_createcluster to verify the locale is valid for command pg_createcluster. It can be other than nl_NL.cp1252_fwd_basic.

Since -l command lists it, I think that is the one.

> Can you leave only nl_NL.cp1252_fwd_basic for locale -a output and recreate the cluster again?

That is exactly what I have in my /usr/lib/locale/. The other were deleted without my will. Well, beside C.UTF-8 and the locale-archive. The latter is in binary and I am a bit scared to delete because of system stability.

#48 - 07/20/2021 12:28 PM - Eric Faulhaber

Ovidiu, are the obsolete FWD locales perhaps being generated due to changes you've made in the list of supported locales, and/or to /etc/locale.gen? This is just an educated guess; I haven't tried this myself, so I don't know if that would work.

Speaking of this, if possible, we need to ensure the FWD locales are re-generated automatically at the appropriate times, so these are not constantly being lost after system updates.

#49 - 07/20/2021 12:35 PM - Eugenie Lyzenko

The only way to have cluster created for me is to add the following locale:
sudo localedef -c -f CP1252 -i nl_NL.cp1252_fwd_basic /usr/lib/locale/nl_NL.cp1252/

After this I can create cluster for nl_NL.cp1252_fwd_basic locale.

locale -a at this time:

```
...
en_ZW.utf8
nl_NL.cp1252
nl_NL.cp1252fwdbasic
nl_NL.cp1252_fwd_basic
POSIX
...
```

#50 - 07/20/2021 12:36 PM - Ovidiu Maxiniuc

Eric Faulhaber wrote:

> Ovidiu, are the obsolete FWD locales perhaps being generated due to changes you've made in the list of supported locales, and/or to /etc/locale.gen? This is just an educated guess; I haven't tried this myself, so I don't know if that would work.

My /etc/locale.gen and /usr/share/i18n/SUPPORTED do not contain any line p2j. I added a single line:

```
nl_NL.cp1252_fwd_basic CP1252
```

in /etc/locale.gen.

There is one more file: /etc/default/locale. I changed the first line to be:

```
LANG=en_US.iso88591_fwd_basic
```

#51 - 07/20/2021 12:39 PM - Eugenie Lyzenko

Eugenie Lyzenko wrote:

> The only way to have cluster created for me is to add the following locale:
> sudo localedef -c -f CP1252 -i nl_NL.cp1252_fwd_basic /usr/lib/locale/nl_NL.cp1252/
>
> After this I can create cluster for nl_NL.cp1252_fwd_basic locale.
>
> locale -a at this time:
> [...]

Well, the surprise, the cluster was created with

```
sudo -u postgres pg_createcluster -p 5433 9.5 vanm --locale=nl_NL.cp1252 -d /opt/data/9.5/vanm/
Creating new PostgreSQL cluster 9.5/vanm ...
/usr/lib/postgresql/9.5/bin/initdb -D /opt/data/9.5/vanm/ --auth-local peer --auth-host md5 --locale nl_NL.cp1
252
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with locale "nl_NL.cp1252".
The default database encoding has accordingly been set to "WIN1252".
The default text search configuration will be set to "dutch".
...
```

Sorry, I need to re-do this again.

#52 - 07/20/2021 12:54 PM - Eugenie Lyzenko

Well, now I also have the required cluster:

```
...
                                    List of databases
   Name    |  Owner   | Encoding |        Collate        |         Ctype         |   Access privileges
-----------+----------+----------+-----------------------+-----------------------+---------------------
 postgres  | postgres | WIN1252  | nl_NL.cp1252_fwd_basic | nl_NL.cp1252_fwd_basic |
 template0 | postgres | WIN1252  | nl_NL.cp1252_fwd_basic | nl_NL.cp1252_fwd_basic | =c/postgres          +
```

```
          |          |         |                        |                        | postgres=CTc/postgres
 template1 | postgres | WIN1252 | nl_NL.cp1252_fwd_basic | nl_NL.cp1252_fwd_basic | =c/postgres          +
          |          |         |                        |                        | postgres=CTc/postgres
(3 rows)
...
```

locale -a:

```
...
C
C.UTF-8
en_AG
en_AG.utf8
en_AU.utf8
en_BW.utf8
en_CA.utf8
en_DK.utf8
en_GB.utf8
en_HK.utf8
en_IE.utf8
en_IL
en_IL.utf8
en_IN
en_IN.utf8
en_NG
en_NG.utf8
en_NZ.utf8
en_PH.utf8
en_SG.utf8
en_US.utf8
en_ZA.utf8
en_ZM
en_ZM.utf8
en_ZW.utf8
nl_NL.cp1252fwdbasic
nl_NL.cp1252_fwd_basic
POSIX
ru_RU.utf8
...
```

The steps to do this:
1. Include nl_NL.cp1252_fwd_basic CP1252 into /etc/locale.gen.
2. Run sudo locale-gen.
3. Run sudo localedef -c -f CP1252 -i nl_NL.cp1252_fwd_basic /usr/lib/locale/nl_NL.cp1252_fwd_basic/.

After this I was able to make the cluster. Note I have now two similar locales with different names.

So finally I'm starting DB import for new cluster.

#53 - 07/20/2021 01:06 PM - Ovidiu Maxiniuc

Yes, but the problem is that you need to run step 3 the first. If /usr/lib/locale/nl_NL.cp1252_fwd_basic is not generated, sudo locale-gen will not be able to process it, even if it is present in /etc/locale.gen.

Also, when I run sudo locale-gen --no-archive on a clean /usr/lib/locale/, I can see nl_NL.cp1252fwdbasic directory getting created. The content is bit-by-bit identical for each file to nl_NL.cp1252_fwd_basic directory.

I think locale-gen is messing with the underscore. I noticed it also does a purge by default. And if it is executed automatically (with an system update) it might broke our locale. I think that happened to my system.

There is also another interesting file I found while googling the issue but I cannot exactly fit into the picture: /var/lib/locales/supported.d/local.

#54 - 07/20/2021 02:05 PM - Greg Shah

Why are you changing /etc/default/locale? Wouldn't this make the JVM start with LANG=en_US.iso88591_fwd_basic? We want the FWD servers and clients to have UTF-8 as the default.

#55 - 07/20/2021 05:15 PM - Ovidiu Maxiniuc

Greg Shah wrote:

> Why are you changing /etc/default/locale? Wouldn't this make the JVM start with LANG=en_US.iso88591_fwd_basic? We want the FWD servers and clients to have UTF-8 as the default.

Indeed, I reverted back.

#56 - 07/20/2021 05:27 PM - Ovidiu Maxiniuc

I used a fresh Ubuntu installation to create the PG cluster with nl_NL.cp1252_fwd_basic Collation and Ctype. However, the process is not straightforward:

1. copy nl_NL.cp1252_fwd_basic and unpack the charmap
2. sudo localedef -c -f CP1252 -i nl_NL.cp1252_fwd_basic /usr/lib/locale/nl_NL.cp1252_fwd_basic/
   At this moment the new directory (nl_NL.cp1252_fwd_basic) is visible in /usr/lib/locale and locale -a BUT the cluster cannot be created yet.
3. add nl_NL.cp1252_fwd_basic CP1252 line in /etc/locale.gen and execute sudo locale-gen.
   At this moment the nl_NL.cp1252_fwd_basic directory is gone. locale -a will list nl_NL.cp1252fwdbasic instead. The cluster cannot be created yet.
4. Run sudo localedef -c -f CP1252 -i nl_NL.cp1252_fwd_basic /usr/lib/locale/nl_NL.cp1252_fwd_basic/ again.
   The nl_NL.cp1252_fwd_basic is recreated. Both locales (nl_NL.cp1252fwdbasic and nl_NL.cp1252_fwd_basic) are visible with locale -a. The content is identical. Now the cluster CAN be created.

#57 - 07/20/2021 05:58 PM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

> I used a fresh Ubuntu installation to create the PG cluster with nl_NL.cp1252_fwd_basic Collation and Ctype. However, the process is not straightforward:
>
>   1. copy nl_NL.cp1252_fwd_basic and unpack the charmap
>   2. sudo localedef -c -f CP1252 -i nl_NL.cp1252_fwd_basic /usr/lib/locale/nl_NL.cp1252_fwd_basic/
>   At this moment the new directory (nl_NL.cp1252_fwd_basic) is visible in /usr/lib/locale and locale -a BUT the cluster cannot be created yet.
>   3. add nl_NL.cp1252_fwd_basic CP1252 line in /etc/locale.gen and execute sudo locale-gen.
>   At this moment the nl_NL.cp1252_fwd_basic directory is gone. locale -a will list nl_NL.cp1252fwdbasic instead. The cluster cannot be created yet.
>   4. Run sudo localedef -c -f CP1252 -i nl_NL.cp1252_fwd_basic /usr/lib/locale/nl_NL.cp1252_fwd_basic/ again.
>   The nl_NL.cp1252_fwd_basic is recreated. Both locales (nl_NL.cp1252fwdbasic and nl_NL.cp1252_fwd_basic) are visible with locale -a. The content is identical. Now the cluster CAN be created.

Why the step 2 above is required if it run once again on step 4? I used almost the same scenario (except not run step 2).

#58 - 07/20/2021 06:12 PM - Eugenie Lyzenko

My import completed for nl_NL.cp1252_fwd_basic based cluster.

The DB looks good.

#59 - 07/21/2021 09:44 AM - Ovidiu Maxiniuc

Eugenie Lyzenko wrote:

> Why the step 2 above is required if it run once again on step 4? I used almost the same scenario (except not run step 2).

That is because the original nl_NL.cp1252_fwd_basic locale folder created by localedef will be deleted when the locale-gen is executed at step 3. Instead they will be generated as:

```
en_GB.iso88591fwdbasic
en_US.cp1252fwdbasic
en_US.iso885915fwdbasic
en_US.iso88591fwdbasic
en_US.utf8
en_US.utf8fwdbasic
nl_NL.cp1252fwdbasic
```

The create cluster will fail with

```
initdb: invalid locale name "nl_NL.cp1252_fwd_basic"
Error: initdb failed
```

In fact, before step 4, I cannot create the cluster neither with nl_NL.cp1252fwdbasic locale.

At least this happens for me.

#60 - 07/21/2021 10:26 AM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

> Eugenie Lyzenko wrote:
>
> > Why the step 2 above is required if it run once again on step 4? I used almost the same scenario (except not run step 2).
>
> That is because the original nl_NL.cp1252_fwd_basic locale folder created by localedef will be deleted when the locale-gen is executed at step 3. Instead they will be generated as:
> [...]
>
> The create cluster will fail with
> [...]
>
> In fact, before step 4, I cannot create the cluster neither with nl_NL.cp1252fwdbasic locale.
>
> At least this happens for me.

Yes exactly. I just noted the directory created on step 2 will then be removed on step 3 by locale-gen command. That's why I think the step 2 can be bypassed. The step 4 is mandatory one.

```
initdb: invalid locale name "nl_NL.cp1252_fwd_basic"
```

#61 - 07/21/2021 12:46 PM - Roger Borrello

Are those steps only required in an upgrade scenario? Perhaps the steps should be mentioned in the Wiki in a separate section.

#62 - 07/21/2021 12:53 PM - Eugenie Lyzenko

Roger Borrello wrote:

> Are those steps only required in an upgrade scenario?

Or if you want to clean up the system from previously installed locales that are obsolete at given time. My observation the system updater runs locale-gen when there are libc related changes should be applied.

#63 - 07/21/2021 01:04 PM - Ovidiu Maxiniuc

Roger Borrello wrote:

> Are those steps only required in an upgrade scenario? Perhaps the steps should be mentioned in the Wiki in a separate section.

Yes, this is what I intend to.
But, even with all those steps, I cannot always reliable create the cluster :(. And, what is worse, lately I cannot create it back. Not in my workstation (bionic), neither in VM (focal).

The underscore is definitely a problem. When I try creating the locale and then the cluster using the old naming (@ as separator) it works each time, like a charm, no need to add the locale to locale.gen or running locale-gen.

#64 - 07/21/2021 01:19 PM - Eric Faulhaber

OK, so the underscores are a problem with Linux. The @ symbols are a problem with H2. Should we just get rid of all the separators in the second part of the name? E.g., en_US.iso88591fwdbasic? It's not ideal, but it should be unique. Thoughts?

#65 - 07/21/2021 01:49 PM - Eugenie Lyzenko

Eric Faulhaber wrote:

> OK, so the underscores are a problem with Linux. The @ symbols are a problem with H2. Should we just get rid of all the separators in the second part of the name? E.g., en_US.iso88591fwdbasic? It's not ideal, but it should be unique. Thoughts?

How about en_US.iso88591FWDBasic or en_US.iso88591FwdBasic?

#66 - 07/21/2021 01:59 PM - Eric Faulhaber

I'm ok with en_US.iso88591FWDbasic, but I have a feeling locale-gen lowercases everything after the dot, so I was trying to accommodate that. Ovidiu, please give this a try, and go with the lowercase version if that does not work.

Hopefully, this is the last round of experimentation before we can get a working version of all this.

#67 - 07/21/2021 02:08 PM - Eugenie Lyzenko

Eric Faulhaber wrote:

> I'm ok with en_US.iso88591FWDbasic, but I have a feeling locale-gen lowercases everything after the dot, so I was trying to accommodate that. Ovidiu, please give this a try, and go with the lowercase version if that does not work.
>
> Hopefully, this is the last round of experimentation before we can get a working version of all this.

Or may be: en_US.iso88591[fwd]basic?

#68 - 07/21/2021 02:58 PM - Roger Borrello

If/when there is a change, will there be another conversion required? I was about to update the customer's environment with revision 12694. This includes a change to their directory.xml to update the embedded-collation.

I just want to minimize the churn.

#69 - 07/21/2021 03:01 PM - Eric Faulhaber

Sorry for the churn. Had I anticipated these seemingly simple name changes would be so problematic, I would have had this work be in a separate branch.

#70 - 07/21/2021 03:01 PM - Greg Shah

[and] are probably not great since they have meaning in other languages including in shell script.

If the hyphen works, that would be best, but if not then the @ is probably the best choice. A separator char is pretty useful here.

#71 - 07/21/2021 03:02 PM - Ovidiu Maxiniuc

Sorry for the silence of the last hours. I continued my experiments. It looks like the underscore (_) is NOT the issue. The problem is the DOT (.). I do not know what is so special with it, but when it is used, things go wrong. Sometimes the locale is visible and usable (like I managed on Saturday and once yesterday and Eugenie with the current cluster) but most of the time, it is just visible. I really do not like this probability. It is possible that there is a secondary cause but I did a lot of experiments and I found nothing reliable.

I tried to confirm my assumption by looking at locale-gen/localedef source code but I was not able to find them in this time.

There is a good news, though. The process works each time (I've created the locale then the cluster a few times on real and virtual machines) if we use nl_NL@cp1252_fwd_basic instead. I think this is somewhere at the middle. Probably this is the reason why the @ character is used for other locales.

#72 - 07/21/2021 03:03 PM - Greg Shah

That seems fine.

#73 - 07/21/2021 03:06 PM - Eric Faulhaber

I don't understand. The dot is used in many of the standard locale names.

For example, locale -a on my system currently reports:

```
C
C.UTF-8
dutch
en_AG
en_AG.utf8
en_AU.utf8
en_BW.utf8
en_CA.utf8
en_DK.utf8
en_GB
en_GB.cp1252
en_GB.iso88591
en_GB.iso885915
en_GB.utf8
en_HK.utf8
en_IE.utf8
en_IN
en_IN.utf8
en_NG
en_NG.utf8
en_NZ.utf8
en_PH.utf8
en_SG.utf8
en_US
en_US.cp1252
en_US.iso88591
en_US.iso885915
en_US.utf8
en_ZA.utf8
en_ZM
en_ZM.utf8
en_ZW.utf8
nl_NL
nl_NL.cp1252
nl_NL.iso88591
nl_NL.iso885915
POSIX
```

#74 - 07/21/2021 03:15 PM - Ovidiu Maxiniuc

Yes, that puzzled me too. Notice that after the . there is no _ character. The _ in out locale names are dropped.

There is another thing mentioned earlier. Even if locale -a displays both nl_NL.cp1252fwdbasic and nl_NL.cp1252_fwd_basic, none of them is usable. I tried to create the cluster with each of them and in both cases I got

```
initdb: invalid locale name "nl_NL.cp1252[_]fwd[_]basic"
Error: initdb failed
```

Only inspecting the OS source code related to creating and accessing the locales can give a straight answer. Using the black-box approach I could not acquire enough information.

#75 - 07/21/2021 03:28 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

> There is a good news, though. The process works each time (I've created the locale then the cluster a few times on real and virtual machines) if we use nl_NL@cp1252_fwd_basic instead. I think this is somewhere at the middle. Probably this is the reason why the @ character is used for other locales.

Will locale-gen properly recreate this, or are we just stuck recreating the custom locale and resetting the permissions after system updates which regenerate the locales?

#76 - 07/21/2021 04:27 PM - Ovidiu Maxiniuc

Yes, if the new locale was added to /etc/locale.gen. This needs to be added in our wiki.

#77 - 07/21/2021 10:36 PM - Roger Borrello

Eric Faulhaber wrote:

> Sorry for the churn. Had I anticipated these seemingly simple name changes would be so problematic, I would have had this work be in a separate branch.

I mention "churn" only because Constantin had a change in [#5557](#5557) which requires a re-conversion. I didn't want to start one if there was going to be another one for this change.

#78 - 07/22/2021 11:08 AM - Ovidiu Maxiniuc

Roger Borrello wrote:

> If/when there is a change, will there be another conversion required? I was about to update the customer's environment with revision 12694. This includes a change to their directory.xml to update the embedded-collation.
>
> I just want to minimize the churn.

No, the conversion will NOT be affected. The only exception is the first line of the H2 DDLs for creating the tables, where the collation is declared.

The good part is that can be altered manually, to any of the collations declared in the static initializer of P2JCollatorProvider. In fact, I am a bit against this line which forces the schema to be used only with the specified collation. Momentarily FWD handles collation names like: en_US_P2J (kept for legacy) and en_US_iso88591_fwd_basic. Separately, for en_US.iso88591_fwd_basic the DOT (not accepted by H2) is silently converted to underscore so these locale names are accepted as well. I will commit today an update to support en_US@iso88591_fwd_basic (this seems to be the final solution) which is what Linux/PG also accept.

The bad part is that a database is created (both dialects) with a certain collation/locale and it becomes unusable if the collation/locale is not available any more. The only solution is to recreate the database and redo the import.

#79 - 07/22/2021 02:01 PM - Eugenie Lyzenko

Guys,

If another conversion is planned during next day please let me know within next 21 hour. This will ensure the fresh build to be ready till Monday.

#80 - 07/22/2021 10:50 PM - Eugenie Lyzenko

Ovidiu,

I have started new conversion for 12709. Just because I need to verify everything from scratch again.

Are you planning to add more conversion related changes tomorrow in this area?

#81 - 07/22/2021 10:53 PM - Eric Faulhaber

Eugenie Lyzenko wrote:

> Are you planning to add more conversion related changes tomorrow in this area?

As far as I know, there are no more conversion changes in this area planned for tomorrow.

#82 - 07/22/2021 10:54 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

> In fact, I am a bit against this line which forces the schema to be used only with the specified collation.

What is your specific concern in this regard? Is this limitation any more restrictive than what we have with a PostgreSQL database?

#83 - 07/23/2021 09:08 AM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

> Roger Borrello wrote:
>
> > If/when there is a change, will there be another conversion required? I was about to update the customer's environment with revision 12694. This includes a change to their directory.xml to update the embedded-collation.
> >
> > I just want to minimize the churn.
>
> No, the conversion will NOT be affected. The only exception is the first line of the H2 DDLs for creating the tables, where the collation is declared.
>
> The good part is that can be altered manually, to any of the collations declared in the static initializer of P2JCollatorProvider. In fact, I am a bit against this line which forces the schema to be used only with the specified collation. Momentarily FWD handles collation names like: en_US_P2J (kept for legacy) and en_US_iso88591_fwd_basic. Separately, for en_US.iso88591_fwd_basic the DOT (not accepted by H2) is silently converted to underscore so these locale names are accepted as well. I will commit today an update to support en_US@iso88591_fwd_basic (this seems to be the final solution) which is what Linux/PG also accept.
>
> The bad part is that a database is created (both dialects) with a certain collation/locale and it becomes unusable if the collation/locale is not available any more. The only solution is to recreate the database and redo the import.

Do I need to update directory.xml and p2j.cfg.xml for a project to use nl_NL@cp1252_fwd_basic now?

#84 - 07/23/2021 10:00 AM - Ovidiu Maxiniuc

Eugenie Lyzenko wrote:

> Do I need to update directory.xml and p2j.cfg.xml for a project to use nl_NL@cp1252_fwd_basic now?

The locale name at OS level is nl_NL@cp1252_fwd_basic. The Locale which is used by H2 to load the collation is nl_NL_cp1252_fwd_basic. However, using nl_NL@cp1252_fwd_basic or even nl_NL.cp1252_fwd_basic in directory.xml and p2j.cfg.xml are also acceptable since the @ and . are silently converted to _ when FWD use the value.

My recommendation is to use nl_NL@cp1252_fwd_basic in all places. Note that, after a conversion, the first line of the DDL for creating the H2 databases will contain the 'normalized' nl_NL_cp1252_fwd_basic. This is because this SQL script is used outside FWD (usually by the ant script).

#85 - 07/23/2021 10:07 AM - Ovidiu Maxiniuc

Eric Faulhaber wrote:

> Ovidiu Maxiniuc wrote:
>
> > In fact, I am a bit against this line which forces the schema to be used only with the specified collation.
>
> What is your specific concern in this regard? Is this limitation any more restrictive than what we have with a PostgreSQL database?

Yes. Here is the difference:

- with same PG DDL you can create multiple clusters using different collations. The collation/locale is not contained in DDL.
- the unchanged H2 DDLs can create only databases with the very specific collation which is hardcoded on first line. Of course, this is not a tough constraint, that line can be always changed with any text editor.

#86 - 07/23/2021 10:09 AM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

> Eugenie Lyzenko wrote:
>
> > Do I need to update directory.xml and p2j.cfg.xml for a project to use nl_NL@cp1252_fwd_basic now?

The locale name at OS level is nl_NL@cp1252_fwd_basic. The Locale which is used by H2 to load the collation is nl_NL_cp1252_fwd_basic. However, using nl_NL@cp1252_fwd_basic or even nl_NL.cp1252_fwd_basic in directory.xml and p2j.cfg.xml are also acceptable since the @ and . are silently converted to _ when FWD use the value.

My recommendation is to use nl_NL@cp1252_fwd_basic in all places. Note that, after a conversion, the first line of the DDL for creating the H2 databases will contain the 'normalized' nl_NL_cp1252_fwd_basic. This is because this SQL script is used outside FWD (usually by the ant script).

I have committed the changes to the customer project.

#87 - 07/23/2021 10:26 AM - Roger Borrello

Just confirming... making the changes in my directory.xml (from en_GB_iso88591_fwd_basic to en_GB@iso88591_fwd_basic) is all that is required. I do **not** have to recompile the locale in the OS, nor recreate the database.

#88 - 07/23/2021 10:41 AM - Ovidiu Maxiniuc

Roger Borrello wrote:

> Just confirming... making the changes in my directory.xml (from en_GB_iso88591_fwd_basic to en_GB@iso88591_fwd_basic) is all that is required. I do **not** have to recompile the locale in the OS, nor recreate the database.

The setting in directory is for _temp database which is always H2 and always recreated in memory each time the server is started. Since it is H2 you do not need to recompile/update the OS locale. If the permanent database is H2 and was created with en_GB_iso88591_fwd_basic you can keep it (no recreate/reimport) since en_GB@iso88591_fwd_basic and en_GB.iso88591_fwd_basic are synonym with (in the sense that FWD convert their name to) en_GB_iso88591_fwd_basic.
You need to update your p2jspi.jar, though.

Note: in previous revision the English locale for the Great Britain was incorrectly named en_UK instead of en_GB in P2JCollatorProvider. I corrected it in r12709.

#89 - 07/23/2021 10:48 AM - Roger Borrello

Ovidiu Maxiniuc wrote:

> Roger Borrello wrote:
>
> > Just confirming... making the changes in my directory.xml (from en_GB_iso88591_fwd_basic to en_GB@iso88591_fwd_basic) is all that is required. I do **not** have to recompile the locale in the OS, nor recreate the database.

The setting in directory is for _temp database which is always H2 and always recreated in memory each time the server is started. Since it is H2 you do not need to recompile/update the OS locale. If the permanent database is H2 and was created with en_GB_iso88591_fwd_basic you can keep it (no recreate/reimport) since en_GB@iso88591_fwd_basic and en_GB.iso88591_fwd_basic are synonym with (in the sense that FWD convert their name to) en_GB_iso88591_fwd_basic.
You need to update your p2jspi.jar, though.

Note: in previous revision the English locale for the Great Britain was incorrectly named en_UK instead of en_GB in P2JCollatorProvider. I corrected it in r12709.

This is a PostgreSQL database. I created the cluster with en_GB.iso88591_fwd_basic. I have in my directory.xml:

```
<node class="string" name="embedded-collation">
  <node-attribute name="value" value="en_GB_iso88591_fwd_basic"/>
</node>
```

Do I need to recreate the cluster with en_GB@iso88591_fwd_basic?

#90 - 07/23/2021 12:13 PM - Ovidiu Maxiniuc

If the dialect is not H2, the embedded-collation settings under respective database node is not used.
(Eric: currently _meta and dirty databases which use H2 dialect do not set the collation and default to utf-8(?). Shouldn't we use the same collation as the primary database?)

If you managed to create the cluster using en_GB.iso88591_fwd_basic you may keep it. The collation/encoding are exactly the same with en_GB@iso88591_fwd_basic, the difference is only in name. SQL statements will work just fine. The problems will occur only if the locale is deleted (by running intentionally or accidentally locale-gen [--purge]).

If you want to be future-proof, or if the database import if fast (small database) you should switch to the new naming. I have not updated the wiki yet, but the difference is minimal.

#91 - 07/23/2021 12:47 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

> If the dialect is not H2, the embedded-collation settings under respective database node is not used.
> (Eric: currently _meta and dirty databases which use H2 dialect do not set the collation and default to utf-8(?). Shouldn't we use the same collation as the primary database?)

Yes. As noted in [#5491-20](#) and follow-up notes, I want to eliminate the embedded-collation entry from the directory altogether. This setting should be deduced from the primary database and set accordingly. For the _temp database, we would deduce this from the collation and code page settings for the user's session, so that the entire _temp entry can be dropped from the directory. I just haven't gotten to this work yet, so for the moment, the embedded-collation setting is still necessary, and it should match the locale name for the primary database's cluster.

#92 - 07/23/2021 01:11 PM - Roger Borrello

I want to be *future proof*... I love that term!

So I will perform:

1. sudo cp <p2j>/locale/en_GB@iso88591_fwd_basic /usr/share/i18n/locales
2. add en_GB@iso88591_fwd_basic ISO-8859-1 line in /etc/locale.gen and execute sudo locale-gen.
3. sudo localedef -c -f ISO-8859-1 -i en_GB@iso88591_fwd_basic /usr/lib/locale/en_US@iso88591_fwd_basic/
4. drop cluster
5. create cluster
6. import data
7. Change embedded-collation as en_GB_iso88591_fwd_basic to en_GB@iso88591_fwd_basic in directory.xml

#93 - 07/23/2021 01:31 PM - Eugenie Lyzenko

Roger Borrello wrote:

> I want to be *future proof*... I love that term!
>
> So I will perform:
>
> 1. ...
> 2. add en_GB@iso88591_fwd_basic ISO-8859-1 line in /etc/locale.gen and execute sudo locale-gen.

This is not required. Moreover I think it is better to keep the file /etc/locale.gen free of any custom locales. This way you have the ability to clean the system to some "default" state for the cases we need to clean all old custom locales.

However after some system updates (that can run locale-gen) the command sudo localedef -c -f ... is mandatory.

#94 - 07/23/2021 01:36 PM - Roger Borrello

Eugenie Lyzenko wrote:

> This is not required. Moreover I think it is better to keep the file /etc/locale.gen free of any custom locales. This way you have the ability to clean the system to some "default" state for the cases we need to clean all old custom locales.
>
> However after some system updates (that can run locale-gen) the command sudo localedef -c -f ... is mandatory.

What is the best way to reset this to the default state, as I now have a couple of custom ones added? Is that locale-gen with my added line(s) removed?

#95 - 07/23/2021 01:49 PM - Eric Faulhaber

Eugenie Lyzenko wrote:

> Roger Borrello wrote:
>
>> I want to be *future proof*... I love that term!
>>
>> So I will perform:
>>
>> 1. ...
>> 2. add en_GB@iso88591_fwd_basic ISO-8859-1 line in /etc/locale.gen and execute sudo locale-gen.
>
> This is not required. Moreover I think it is better to keep the file /etc/locale.gen free of any custom locales.

No, you **should** add the custom locales to /etc/locale.gen. This is what I've been trying to achieve!

Assuming this works as expected, it will be recommended for all production environments, so we don't have to constantly answer the question, "why can't the FWD server start after a system update?".

#96 - 07/23/2021 01:52 PM - Ovidiu Maxiniuc

2. and 3. do the same thing. In this case they can be used mutually exclusive.
I would go for 2. because once our locale was added to /etc/locale.gen, any 'accidental' execution of locale-gen will regenerate the locale but will not drop it. Otherwise, you need to manually execute step 3. after such 'accidents'.

So my recommended list is:

1. Copy the locale script to OS location:
   sudo cp <p2j>/locale/en_GB@iso88591_fwd_basic /usr/share/i18n/locales
2. add en_GB@iso88591_fwd_basic ISO-8859-1 line in /etc/locale.gen and execute
   sudo locale-gen --no-purge --no-archive
   in order to keep old locales (like en_US@p2j_basic) for other clusters you may have;
3. drop old cluster:
   sudo -u postgres pg_dropcluster 9.5 db-name
4. create the new one with:
   sudo -u postgres pg_createcluster -p 543x 9.5 cluster-name --locale=en_GB@iso88591_fwd_basic --encoding=ISO8859-1 -d /path/to/data/dir
5. run import with
   ant.import.pg
6. server smoke-test

#97 - 07/23/2021 01:58 PM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

> 2. and 3. do the same thing. In this case they can be used mutually exclusive.
> I would go for 2. because once our locale was added to /etc/locale.gen, any 'accidental' execution of locale-gen will regenerate the locale but will not drop it. Otherwise, you need to manually execute step 3. after such 'accidents'.
>
> So my recommended list is:
>
> 1. Copy the locale script to OS location:
>   sudo cp <p2j>/locale/en_GB@iso88591_fwd_basic /usr/share/i18n/locales
> 2. add en_GB@iso88591_fwd_basic ISO-8859-1 line in /etc/locale.gen and execute
>   sudo locale-gen --no-purge --no-archive
>   in order to keep old locales (like en_US@p2j_basic) for other clusters you may have;
> 3. drop old cluster:
>   sudo -u postgres pg_dropcluster 9.5 db-name
> 4. create the new one with:
>   sudo -u postgres pg_createcluster -p 543x 9.5 cluster-name --locale=en_GB@iso88591_fwd_basic --encoding=ISO8859-1 -d
>   /path/to/data/dir
> 5. run import with
>   ant.import.pg
> 6. server smoke-test

So you telling if to add en_GB@iso88591_fwd_basic ISO-8859-1 line in /etc/locale.gen and then execute locale-gen - it will be possible to create new cluster for en_GB@iso88591_fwd_basic, **not** for en_GB@iso88591fwdbasic? No need to to run sudo localedef -c -f ...?

If it is true - this is really great news!

#98 - 07/23/2021 01:59 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

> sudo locale-gen --no-purge --no-archive
> in order to keep old locales (like en_US@p2j_basic) for other clusters you may have;

en_US@p2j_basic is obsolete at this point. Let's cut over to the new locales and leave this behind. I don't want to support any obsolete constructs we don't have to. We are only making it harder for ourselves in the long run leaving such things around. Purge it and don't look back :-)

#99 - 07/23/2021 02:04 PM - Ovidiu Maxiniuc

Eugenie Lyzenko wrote:

> If it is true - this is really great news!

Yes, exactly.
Here a quote from man locale-gen:

> locale-gen is a program that reads the file /etc/locale.gen and invokes localedef for the chosen localisation profiles.

#100 - 07/23/2021 02:10 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

> Eugenie Lyzenko wrote:
>
>> If it is true - this is really great news!
>
> Yes, exactly.
> Here a quote from man locale-gen:
>
>> locale-gen is a program that reads the file /etc/locale.gen and invokes localedef for the chosen localisation profiles.

So long as the locale definitions are in place in /usr/share/i18n/locales and the required character sets are available (and unzipped) in /usr/share/i18n/charmaps, I think we should be good to go. Ovidiu, have you confirmed that the permissions of the generated locale are correct when produced by locale-gen? Setting these manually is always a pain.

#101 - 07/23/2021 02:13 PM - Ovidiu Maxiniuc

Eric Faulhaber wrote:

> So long as the locale definitions are in place in /usr/share/i18n/locales and the required character sets are available (and unzipped) in /usr/share/i18n/charmaps, I think we should be good to go. Ovidiu, have you confirmed that the permissions of the generated locale are correct when produced by locale-gen? Setting these manually is always a pain.

That is correct. No more (sudo localedef ... then 4 sudo chmod ...).

#102 - 07/26/2021 01:53 PM - Ovidiu Maxiniuc

Is this a good time to rename p2jspi.jar to fwdspi.jar?

#103 - 07/26/2021 02:01 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

> Is this a good time to rename p2jspi.jar to fwdspi.jar?

I don't have an objection, but we should search and replace references to it in the documentation as well.

#104 - 07/26/2021 02:17 PM - Greg Shah

We can't just search and replace. We must leave the older docs in place and just explain the difference starting with FWD v4 (which isn't even publicly available yet).

But I have no objection. We need everyone to update anyway, right?

#105 - 07/26/2021 02:28 PM - Constantin Asofiei

Eric Faulhaber wrote:

> Ovidiu Maxiniuc wrote:
>
> > Is this a good time to rename p2jspi.jar to fwdspi.jar?
>
> I don't have an objection, but we should search and replace references to it in the documentation as well.

A problem I see is if someone forgets to remove the old p2jspi.jar and there will be both p2jspi.jar and fwdspi.jar installed.

#106 - 07/26/2021 02:32 PM - Roger Borrello

I'd like a "safe" way to update the fwdspi.jar file. Is this best scripted into install_spawner.sh (even though that name wouldn't be accurate)? I'd like to be a normal part of deployment of the application files after building is completed. What would be considered a best practice for this task?

#107 - 07/26/2021 02:32 PM - Ovidiu Maxiniuc

I do not think this is a problem: p2jspi.jar will provide en_US_P2J via P2JCollatorProvider and fwdspi.jar will provide the new collation(s) via a new FwdCollatorProvider. The former will never be updated and can assure the legacy support if kept.

#108 - 07/26/2021 02:49 PM - Greg Shah

> Is this best scripted into install_spawner.sh (even though that name wouldn't be accurate)?

No, this is too confusing and there is no relationship to the spawner.

> I'd like to be a normal part of deployment of the application files after building is completed.

I agree this would be preferred. On the other hand, I wonder if there is a way to eliminate this extension jar completely.

#109 - 07/26/2021 03:09 PM - Ovidiu Maxiniuc

Greg Shah wrote:

> I agree this would be preferred. On the other hand, I wonder if there is a way to eliminate this extension jar completely.

This is what I was thinking of, too. Since the database will not (or shouldn't be) accessed stand-alone, the collation should be already available in FWD jar/binaries. The problem is that P2JCollatorProvider is a LocaleServiceProvider and *Implementations of these locale sensitive services are packaged using the* Java Extension Mechanism *as installed extensions.* However, this is *has been deprecated and will be removed in a future release* so probably we should move to a new solution.

#110 - 07/26/2021 07:05 PM - Greg Shah

In Loading a Custom TimeZone Provider in Java 8 there is a good explanation of how the service providers load. The important point here is that **LocaleServiceProvider doesn't allow loading from classpath in Java 8 but it does allow this starting in Java 9**. So when we move to Java 11, we need to rework things to use the classpath.

But there is still hope! In [Why does the Java Extension Mechanism not check the classpath?](#), they mention that in Java 8 you can add these to the java command line:

-Djava.locale.providers=SPI,JRE -Djava.ext.dirs=/path/to/extension/jar/

The java.locale.providers tells the JVM to try to load service providers in preference to the built-in locales. The java.ext.dirs allows a directory list to be provided that overrides (or maybe extends) the JRE extension directory. Both are needed. If present, then we just need to have fwdspi.jar in the /path/to/extension/jar/ and it will be loaded. If this works, we can eliminate the copying of the spi jar to the JRE.

Ovidiu: Can you see if this works? If so, then we need to update the server.sh in all our projects to handle this.

#111 - 07/27/2021 04:01 PM - Ovidiu Maxiniuc

The -Djava.locale.providers=SPI,JRE seems to be irrelevant I did not notice any change in behaviour regardless of its value of presence.

OTOH, defining a custom java.ext.dirs have some unexpected results:

- the good part is that, indeed, our collation provider CAN be loaded from a specified directory. I tested with a different location than lib folder to make Java scan a single jar instead of hundreds we have in project, which is not a hard constraint but allows a fast start-up;
- the bad part is that FWD server is NOT starting. It needs some other classes (SunTls12MasterSecret at least) which are normally found in the ext directory. Once the ext directory is redefined, the rest of the jars from the original directory are not accessible any more.

#112 - 07/27/2021 04:05 PM - Roger Borrello

Ovidiu Maxiniuc wrote:

> The -Djava.locale.providers=SPI,JRE seems to be irrelevant I did not notice any change in behaviour regardless of its value of presence.
>
> OTOH, defining a custom java.ext.dirs have some unexpected results:
>
> - the good part is that, indeed, our collation provider CAN be loaded from a specified directory. I tested with a different location than lib folder to make Java scan a single jar instead of hundreds we have in project, which is not a hard constraint but allows a fast start-up;
> - the bad part is that FWD server is NOT starting. It needs some other classes (SunTls12MasterSecret at least) which are normally found in the ext directory. Once the ext directory is redefined, the rest of the jars from the original directory are not accessible any more.

I would think you should be able to include a colon-separated (or semi-colon separated on Windows) list of directories for the value after java.ext.dirs=

#113 - 07/27/2021 04:07 PM - Greg Shah

I guess that we need to add the JRE's ext dir into the dir list that is passed so that both the JRE ext dir and our custom dir are searched.

The -Djava.locale.providers=SPI,JRE was supposed to ensure that your custom providers and the default JRE providers were both loaded. But perhaps the JRE ext providers are not loaded unless you add that path to the dir list.

#114 - 07/27/2021 04:24 PM - Ovidiu Maxiniuc

Roger Borrello wrote:

> I would think you should be able to include a colon-separated (or semi-colon separated on Windows) list of directories for the value after java.ext.dirs=

Yes! you are right, adding the original location will fix the issue.

Greg Shah wrote:

> The -Djava.locale.providers=SPI,JRE was supposed to ensure that your custom providers and the default JRE providers were both loaded. But perhaps the JRE ext providers are not loaded unless you add that path to the dir list.

Yes, theoretically there are four locations: HOST,SPI,CLDR,JRE. If I use simply JRE it will ignore my -D and go to original ext directory. Using JRE,SPI will access the directories from command line so it will fail (SSL error) if the original dir is not appended.

So, appending the original lib/ext to java.ext.dirs seems to be the solution. But is it the correct one? We need to locate this directory first. Is there are OS level setting for it or we need to manipulate the path of java executable (JAVA_HOME if defined)?

#115 - 07/27/2021 04:28 PM - Igor Skornyakov

Roger Borrello wrote:

> I would think you should be able to include a colon-separated (or semi-colon separated on Windows) list of directories for the value after java.ext.dirs=

BTW: It seems that Java 11 doesn't support lib/ext mechanism. At least I failed to use our custom locale with H2 and Java 11.

#116 - 07/27/2021 05:08 PM - Greg Shah

Starting in Java 9, we will be able to load it from the classpath so this will "go away". We will be moving from Java 8 to Java 11 later this year. Until we move to Java 11, we need to still load the spi jar.

#117 - 07/27/2021 05:15 PM - Greg Shah

> We need to locate this directory first. Is there are OS level setting for it or we need to manipulate the path of java executable (JAVA_HOME if defined)?

We could run a Java program that prints the java.ext.dirs value from a default JVM instance. This could be captured and then appended to the value we pass to the server's java command line.

#118 - 07/28/2021 06:49 AM - Ovidiu Maxiniuc

Here is a solution:

```
echo "$(jrunscript -e 'java.lang.System.out.println(java.lang.System.getProperty("java.home"));')"
```

For me, it correctly prints

```
/usr/lib/jvm/java-8-openjdk-amd64/jre
```

#119 - 07/28/2021 07:51 AM - Greg Shah

I prefer to avoid jrunscript because it is an additional dependency which then must be configured/maintained. For example, on my system the default is the Java 11 version, so I get this:

```
Warning: Nashorn engine is planned to be removed from a future JDK release
/usr/lib/jvm/java-11-openjdk-amd64
```

The tool itself is unsupported and subject to change as can be seen by the warning.

Instead, we can simply write a class like this:

```
public class PrintSystemProp
{
   public static void main(String[] args)
   {
      if (args.length != 1)
```

```
      {
          // TODO: print syntax
          System.exit(-1);
      }

      System.out.println(System.getProperty(args[0]));
   }
}
```

When I run java PrintSystemProp java.ext.dirs it outputs:

```
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/ext:/usr/java/packages/lib/ext
```

Since this would be part of the p2j.jar, we would actually run it like this from server.sh:

```
JAVA_EXT_DIRS=$(${prog} ${cpath} com.goldencode.util.PrintSystemProp java.ext.dirs)
```

From there we can prepend to the list and add it to the command line of the FWD server. I've checked in 3821c rev 12730 with this program. By doing it this way we won't depend on jrunscript and we will use the same version of java as will be used by the server.

#120 - 07/28/2021 07:59 AM - Ovidiu Maxiniuc

jrunscript is deprecated and will be removed in a future JDK. At the same time lib/ext mechanism is already dropped since Java9. How will our custom locale be loaded in Java11? Setting the java.ext.dirs to right location(s) will not help since there will be nobody to access it. I do not have a Java11 installed yet, for testing.

#121 - 07/28/2021 08:08 AM - Greg Shah

In Java 11, the SPI will load from the classpath. When we work the move to Java 11, this will be part of the work.

#122 - 07/28/2021 09:47 AM - Greg Shah

- Related to Support #4550: eliminate the need to place the p2jspi.jar in the JVM extension directory added

#123 - 07/28/2021 09:49 AM - Greg Shah

- Related to Support #5567: move FWD to Java 11 added

#124 - 07/28/2021 03:17 PM - Ovidiu Maxiniuc

Starting with r12732, the collation provider was renamed to FwdCollatorProvider and the jar fwdspi.jar.
Starting with r447 of the customer project, the fwdspi.jar is loaded from command line using the PrintSystemProp. The server starts correctly, even if there are no more FWD jars in java's ext directory.

#125 - 07/28/2021 03:25 PM - Roger Borrello

Ovidiu Maxiniuc wrote:

> Starting with r12732, the collation provider was renamed to FwdCollatorProvider and the jar fwdspi.jar.
> Starting with r447 of the customer project, the fwdspi.jar is loaded from command line using the PrintSystemProp. The server starts correctly, even if there are no more FWD jars in java's ext directory.

Can you post some snippets (or if the wiki is going to be updated, just let me know) of the updates needed for a customer project?

#126 - 07/28/2021 03:35 PM - Greg Shah

Please send an email to the entire team to explain the changes needed to move to rev 12732.

#127 - 07/28/2021 03:36 PM - Greg Shah

Also, please apply the needed changes to Hotel GUI and Hotel ChUI.

#128 - 07/28/2021 10:17 PM - Eric Faulhaber

Ovidiu, I added the following to my /etc/locale.gen file:

```
en_GB@cp1252_fwd_basic CP1252
en_GB@iso88591_fwd_basic ISO-8859-1
en_US@cp1252_fwd_basic CP1252
en_US@iso88591_fwd_basic ISO-8859-1
nl_NL@cp1252_fwd_basic CP1252
```

These are the only uncommented entries in the file.

After running locale-gen, locale -a reports:

```
C
C.UTF-8
en_AG
en_AG.utf8
en_AU.utf8
en_BW.utf8
en_CA.utf8
en_DK.utf8
en_GB.cp1252@cp1252_fwd_basic
en_GB@cp1252_fwd_basic
en_GB@iso88591_fwd_basic
en_GB.iso88591@iso88591_fwd_basic
en_GB.utf8
en_HK.utf8
en_IE.utf8
en_IN
```

```
en_IN.utf8
en_NG
en_NG.utf8
en_NZ.utf8
en_PH.utf8
en_SG.utf8
en_US.cp1252@cp1252_fwd_basic
en_US@cp1252_fwd_basic
en_US@iso88591_fwd_basic
en_US.iso88591@iso88591_fwd_basic
en_US.utf8
en_ZA.utf8
en_ZM
en_ZM.utf8
en_ZW.utf8
nl_NL.cp1252@cp1252_fwd_basic
nl_NL@cp1252_fwd_basic
POSIX
```

Notice the strange "extra" locales:

```
en_GB.cp1252@cp1252_fwd_basic
en_GB.iso88591@iso88591_fwd_basic
en_US.cp1252@cp1252_fwd_basic
en_US.iso88591@iso88591_fwd_basic
nl_NL.cp1252@cp1252_fwd_basic
```

I tried with --purge also, but no difference.

Any idea what those are? They don't seem right.

#129 - 07/29/2021 07:38 AM - Ovidiu Maxiniuc

I do not know why are those created. The locale-gen seems a bit buggy to me; or at least the command-line help/manual do not match what it actually

does. The default values for --purge/--no-purge and --archive/--no-archive seem reversed. I guess it is the same with --lang, which I suspect causes this. I would like to see the source code for it but I could not find it (must be some -src package right?).

The en_US.cp1252@cp1252_fwd_basic and other alike do not seem right, but they should not interfere with the legit ones. Try running

```
locale-gen --no-archive
```

and see if they are still there.

#130 - 08/03/2021 11:48 AM - Roger Borrello

If I am getting Unable to read PSC footer. Failed magic check.. Heuristic search found a PSC footer candidate. The file was probably altered after being exported from database. for some tables being imported, does that render the PSC data useless? I am altering 2 tables where I add my own information. Is there a tool to regenerate or recalculate that magic check number?

#131 - 08/03/2021 12:08 PM - Ovidiu Maxiniuc

If the import process finds the PSC footer it will be able to read and process it.
The correct offsets are not so important, having the correct values will speed up the process of opening files. If they are not valid, the user will be informed about the situation - as you noticed. Don't worry, this is not critical.

However, if you want a clean import you need to use an editor which allows you to find the offset **in bytes(!)** of current character, relative to the beginning of the file. Open the .d and be sure the editor uses the correct encoding. Move the cursor on the first column of the line with PSC marker. Use this offset and overwrite the digits on the very last line (10 decimal digits, right aligned, filled with 0 at the left side). Save the file.

Note: jedit has some kind of offset indicator but it fails to count the windows EOLN as two bytes. To adjust this number add the number of EOLNs before this line (aka number of lines before the PSC marker) if the original file comes from Windows. In all cases, DO NOT mix Linux and Windows EOLNs!

#133 - 08/18/2021 03:04 PM - Roger Borrello

Will the custom locale part of the Database Server Setup wiki be updated with the new steps soon?

Also, does anyone have any experience with Red Hat, and what it might take to do this on that platform? It seems they don't have a locale-gen command, so there might be another method. If I research this, is this an OS locale customization, or a PostgreSQL locale customization?

#134 - 08/18/2021 05:34 PM - Greg Shah

is this an OS locale customization

It is at the OS level.

#135 - 08/18/2021 06:37 PM - Ovidiu Maxiniuc

Roger Borrello wrote:

> Will the custom locale part of the [Database Server Setup](#) wiki be updated with the new steps soon?

I updated that paragraph to reflect the newer, safer method for generating the locales (using locale-gen instead of localedef + multiple chmod s).

> Also, does anyone have any experience with Red Hat, and what it might take to do this on that platform? It seems they don't have a locale-gen command, so there might be another method.

I do not have much experience with Red Hat, but if there is no locale-gen command for this distribution, the paragraph will need very specific instructions.

#136 - 08/24/2021 04:28 PM - Roger Borrello

Team... I was working with my VM and moved to the new custom locale en_GB@iso88591_fwd_basic. The cluster was created, and the import.db worked fine.

I have 2 ways to start the server. The way that works is from the ./deploy/server directory, running ./server.sh. This contains the scripting to use:

```
spi="-Djava.locale.providers=SPI,JRE -Djava.ext.dirs=$(${prog} ${cpath} com.goldencode.util.PrintSystemProp ja
va.ext.dirs):../lib/spi"
```

Everything starts just fine.

The other way to start is from the /opt/<project>/server directory. This is the current working directory when I use systemctl to start it as a service, and the server.sh script is the exact same one as mentioned above. The log messages are the same whether I run interactively, or start the service:

```
[08/24/2021 16:22:29 EDT] (com.goldencode.p2j.persist.dialect.H2Helper:SEVERE) Error setting P2J-specific coll
ation;  please ensure p2jspi.jar is installed in Java extension directory
com.goldencode.p2j.cfg.ConfigurationException:  Initialization failure
    at com.goldencode.p2j.main.StandardServer.hookInitialize(StandardServer.java:2130)
    at com.goldencode.p2j.main.StandardServer.bootstrap(StandardServer.java:1030)
    at com.goldencode.p2j.main.ServerDriver.start(ServerDriver.java:485)
    at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:444)
    at com.goldencode.p2j.main.ServerDriver.process(ServerDriver.java:209)
    at com.goldencode.p2j.main.ServerDriver.main(ServerDriver.java:863)
Caused by: java.lang.RuntimeException: Error initializing persistence services
    at com.goldencode.p2j.persist.Persistence.initializeInstance(Persistence.java:2895)
    at com.goldencode.p2j.persist.PersistenceFactory.getInstance(PersistenceFactory.java:161)
```

```
    at com.goldencode.p2j.persist.DatabaseManager.initTempDb(DatabaseManager.java:1409)
    at com.goldencode.p2j.persist.DatabaseManager.initialize(DatabaseManager.java:1080)
    at com.goldencode.p2j.persist.Persistence.initialize(Persistence.java:841)
    at com.goldencode.p2j.main.StandardServer$11.initialize(StandardServer.java:1275)
    at com.goldencode.p2j.main.StandardServer.hookInitialize(StandardServer.java:2126)
    ... 5 more
Caused by: com.goldencode.p2j.persist.PersistenceException: Error initializing embedded database
Caused by: org.h2.jdbc.JdbcSQLDataException: Invalid value "en_US_iso88591_fwd_basic" for parameter "collation
"; SQL statement:
set collation "en_US_iso88591_fwd_basic" [90008-200]
    at org.h2.message.DbException.getJdbcSQLException(DbException.java:590)
```

The directory.xml is the exact same one regardless of which I'm starting. The only place collation appears in it is:

```
            <node class="string" name="embedded-collation">
              <node-attribute name="value" value="en_GB@iso88591_fwd_basic"/>
            </node>
```

Where might this check for p2jspi.jar be coming from, and why the US collation?

#137 - 08/24/2021 04:38 PM - Roger Borrello

Roger Borrello wrote:

> Where might this check for p2jspi.jar be coming from, and why the US collation?

Looks like the permissions of the /opt/<project/lib/spi were too strict... they were 644. I set to 755, and now we start up.

#138 - 08/24/2021 04:54 PM - Ovidiu Maxiniuc

Roger Borrello wrote:

why the US collation?

This is a good question. Even if the p2jspi.jar is inaccessible, the directory.xml should dictate the GB collation and FWD should complain about it instead. It seems like you are running with US collation instead. I do not think there are critical differences, yet, please check whether the right value for this setting is read from directory (in H2Helper.java:240, method prepareDatabase(Database)).

Looks like the permissions of the /opt/<project/lib/spi were too strict... they were 644. I set to 755, and now we start up.

I do not recall changing the access rights for spi dir. If this is needed, please add it to build.xml.

#139 - 08/24/2021 06:42 PM - Roger Borrello

Thanks. Have you tried setting permissions on spi/fwdspi.jar to see if it "breaks" you, too?

#140 - 08/25/2021 01:12 PM - Ovidiu Maxiniuc

I have changed manually the access mode of deploy/lib/spi to 644 and I got the exception. However, it complained about the en_GB@iso88591_fwd_basic locale I set in directory. After switching to 755, the server started normally.

However, the OS locale en_GB@iso88591_fwd_basic is used by pgSQL databases automatically, after the cluster and database were created. You do not need to configure anything in the directory for a PostgreSQL database. Adding

```
            <node class="string" name="embedded-collation">
              <node-attribute name="value" value="en_GB@iso88591_fwd_basic"/>
            </node>
```

to a pgSQL database have no effect, in fact it will not be even read.

The p2jspi.jar is used only by H2 databases. The _temp databases are always H2. The Locale for this database must be configured in the directory, otherwise the default en_US@iso88591_fwd_basic will be used. Please check if you have this settings in your directory:

```
    <node class="container" name="standard">
      <node class="container" name="database">
        <node class="container" name="_temp">
          <node class="container" name="p2j">
            <node class="string" name="schema">
              <node-attribute name="value" value="_temp"/>
            </node>
            <node class="boolean" name="load_at_startup">
              <node-attribute name="value" value="TRUE"/>
            </node>
            <node class="string" name="embedded-collation">
               <node-attribute name="value" value="en_GB@iso88591_fwd_basic"/>
            </node>
          </node>
        </node>
```

#141 - 08/25/2021 03:23 PM - Roger Borrello

Ovidiu Maxiniuc wrote:

> The p2jspi.jar is used only by H2 databases. The _temp databases are always H2. The Locale for this database <u>must</u> be configured in the directory, otherwise the default en_US@iso88591_fwd_basic will be used. Please check if you have this settings in your directory:
> [...]

I've never made reference to the _temp database in the customer's directory. I guess if we want the collation to be something other than the US version, I'll have to add it. Is load_at_startup necessary?

#142 - 08/25/2021 03:27 PM - Eric Faulhaber

Roger Borrello wrote:

> Is load_at_startup necessary?

No, not for the _temp database.

BTW, I plan to eliminate the need to configure collation for the _temp database at all in the directory. See #5491-20 and follow-up notes for details. We just haven't had a chance to implement this yet.

#143 - 08/25/2021 03:31 PM - Ovidiu Maxiniuc

Indeed, load_at_startup is not needed for _temp. It is actually skipped for this database.

It seems logical to me that the queries on permanent database to return the same result/order for the temporary database. For this, the collations must match.

I have just saw Eric's note above. I think you should keep the setting(s) for _temp database until #5491-20 is done.

#144 - 08/25/2021 03:42 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

> I think you should keep the setting(s) for _temp database until #5491-20 is done.

Yes, I did not mean to imply the other settings (besides load_at_startup) were unnecessary. I agree, those are needed until #5491-20 is implemented.

#145 - 09/01/2021 10:36 AM - Greg Shah

Can we close this task or is there something remaining to work on, review or test?

#146 - 09/01/2021 10:58 AM - Eric Faulhaber

The main work of creating the locales and collations to cover current projects is complete.

Some related work described in #5491-20 and follow-on entries remains. That is about simplifying the configuration of collation in the directory.