Table of Contents

[**osFinancials5.1.0.233 - Known issues** 1](#_Toc187898373)

[**Customise languages - Codepage setting/Default Character Set - Firebird databases** 1](#_Toc187898374)

[**Default Character Set : NONE - Database properties in FlameRobin** 2](#_Toc187898375)

[**Options explored** 3](#_Toc187898376)

[***Datapump*** 3](#_Toc187898377)

[**Changing the CHARACTER SET** 5](#_Toc187898378)

[**IBEXPERT tests** 6](#_Toc187898379)

[***Execute ALTER DATABASE Script error*** 6](#_Toc187898380)

[***Steps to Create a New Database with UTF8:*** 6](#_Toc187898381)

[***Step-by-Step Migration:*** 6](#_Toc187898382)

[***Tips:*** 6](#_Toc187898383)

**osFinancials5.1.0.233 - Known issues**

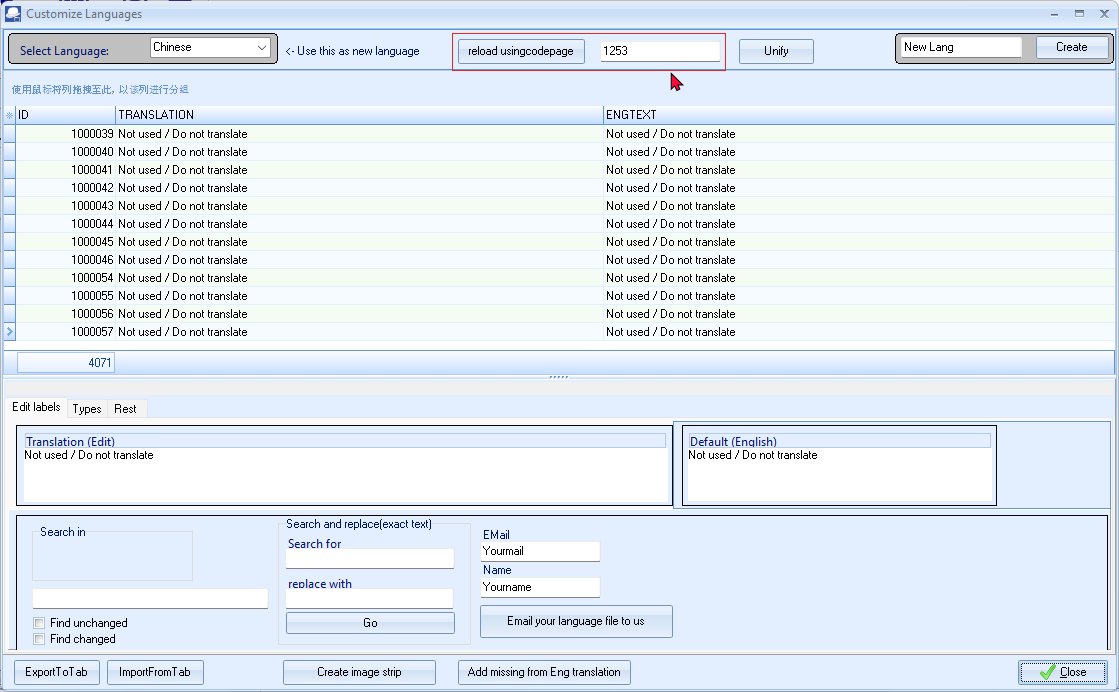
**Customise languages - Codepage setting/Default Character Set - Firebird databases**

Since osFinancials5.0.0.453 up to osFinancials5.1.0.233 the code settings in **Tools → Customise languages** (Setup ribbon) cannot be changed.

This is replicated in all Sets of Books in Firebird and MSSQL databases.

The default Codepage setting is 1253 which **is specifically designed for the Greek language.** It's a character encoding standard used to represent text in the modern Greek alphabet, including accented letters, punctuation marks, and special symbols.

In basically all languages **Switch language** (**Start** ribbon) (.../bin/languages folder) the Codepage is set to 1253.



Entering a valid Codepage code and click on the **reload usingcodepage** button does not save the Codepage code and reverts back to the original setting (1253).

This is replicated with:

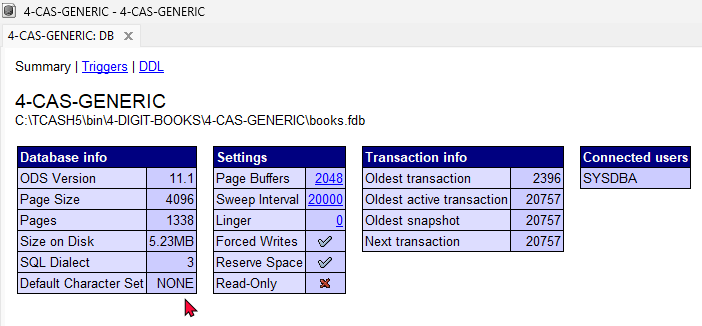
* Unicode enabled setting (n**ounicode=FALSE**)
* Unicode disabled setting (n**ounicode=TRUE**)

**Default Character Set : NONE - Database properties in FlameRobin**

All Sets of Books linked to languages **Default Character Set : NONE** - In Database properties of FlameRobin.

The Default Character Set : NONE for all existing Firebird databases.

When creating a new custom Set of Books, the Default Character Set : NONE remains NONE.



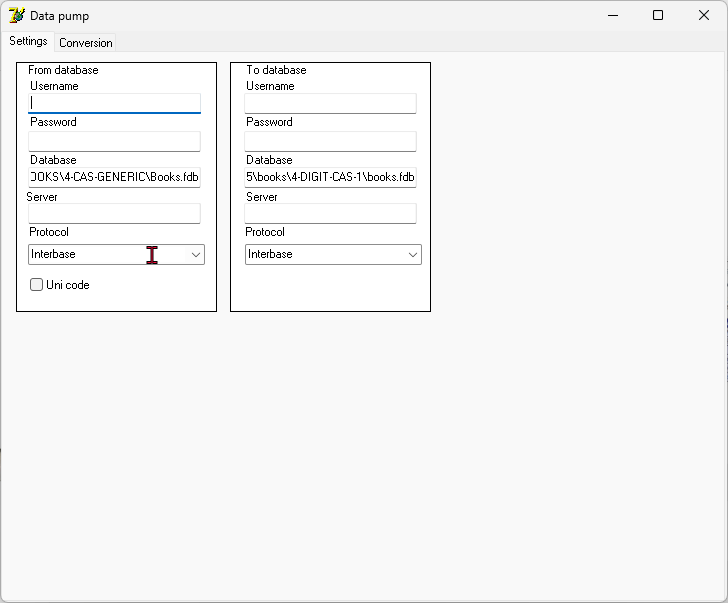
When creating a Set of Books (Customise your own Set of Books (Advanced)?) using the Firebird (default) option, the Default Character Set : NONE for all existing and new Firebird databases.

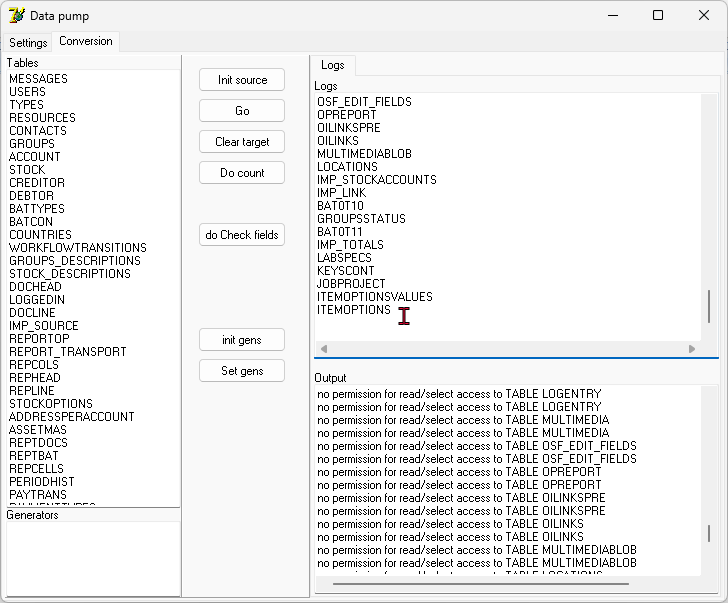
**Options explored**

***Datapump***

Should try to make database with Chinese Codepage - then pump data into that using the Datapump

Protocol = Interbase - Cannot set the language to Chinese language codepage in Create Set of Books - also no premissions in datapump.





**Changing the CHARACTER SET**

**Resource**: <https://www.firebirdsql.org> -

We are on Firebird2.1 = Added in: 2.5

Is this possible to do this in Firebird version 2.1 and what tools do we need for this?

Yes, it is possible to set the default character set and use Unicode in Firebird version 2.1. Here are the steps and tools required to configure it:

**Steps to Set Default Character Set in Firebird 2.1**:

**Create Database with Default Character Set**: When creating a new database, specify the default character set. For example, to create a database with UTF8 as the default character set:

sql

*CREATE DATABASE 'path\_to\_your\_db' DEFAULT CHARACTER SET UTF8;*

**Alter Existing Database**: If you need to change the character set for an existing database, you can alter it:

sql

*ALTER DATABASE SET DEFAULT CHARACTER SET UTF8;*

**Specify Character Set for Columns**: When creating or altering tables, specify the required character set for columns:

sql

*CREATE TABLE your\_table (  
column\_name VARCHAR(255) CHARACTER SET UTF8  
);*

**Tools Needed**:

* **IBExpert**: A comprehensive tool for Firebird databases that provides a user-friendly interface to set and manage character sets.
* **ISQL**: Firebird’s Interactive SQL tool. You can execute DDL commands to create or alter databases and tables to set character sets.
* **Database Service Utility Tools**: Tools like gf[5^]ix, gbak, and others, which are part of the Firebird installation, can assist with various database management tasks.

The tools and steps outlined above should help you set the default character set in your Firebird 2.1 database.

By following these steps, you can set and ensure the use of the appropriate character set and use Unicode effectively for your Firebird database in osFinancials/TurboCASH.

**Resources**:

* IBExpert - [01-documentation:01-08-firebird-documentation:firebird-2-administration-handbook:service-configuration](https://ibexpert.com/docu/doku.php?id=01-documentation:01-08-firebird-documentation:firebird-2-administration-handbook:service-configuration) -
* Firebird documentation - [Character Data Types](https://www.firebirdsql.org/file/documentation/chunk/en/refdocs/fblangref40/fblangref40-datatypes-chartypes.html) -'

**IBEXPERT tests**

***Execute ALTER DATABASE Script error***

*Invalid token. Dynamic SQL Error. SQL error code = -104. Token unknown - line 1, column 16. SET. ---------------------------------- SQLCODE: -104 GDSCODE: 335544569*

It seems Firebird 2.1 does not support using the ALTER DATABASE command with the SET clause as initially suggested. Let's take an alternative approach by creating a new database with the desired character set and migrating your data.

***Steps to Create a New Database with UTF8:***

1. **Backup the Existing Database**: Use gbak to backup your current database:

sql

*gbak -backup old\_database.fdb old\_database.fbk*

1. **Restore the Backup to a New Database with UTF8**: Restore the database from the backup and set the character set to UTF8:

sql

*gbak -create -user SYSDBA -password masterkey old\_database.fbk new\_database.fdb -fix\_fss\_data UTF8 -fix\_fss\_metadata UTF8*

***Step-by-Step Migration:***

1. **Generate SQL Scripts**:

In IBExpert, generate SQL scripts for your current database schema and data. This is done via the Extract Metadata option.

1. **Edit the Scripts**:

Open the extracted script and ensure all tables and columns use the UTF8 character set.

1. **Execute the Scripts**:

Run the edited scripts in IBExpert to create and populate your new database.

***Tips:***

* **Backup Before Proceeding**: Always ensure you have a backup of your current database before making significant changes.
* **Validate Data**: After migration, validate your data in the new database to ensure it has been accurately transferred.
* **Test Thoroughly**: Test all functionalities of your application with the new database to ensure everything works as expected.

Taking these steps should help you transition to using UTF8 without encountering dynamic SQL errors.

If you run into any issues or need further assistance with any of these steps, feel free to ask!