**Creating the Import Environment**

*1. Create the import database on the server*

In order to create a DB for practical purposes with no unneeded data (doesn't matter if the data is already in the TestDB or not), create an empty DB

Fields for creating the new DB

・DB Name : xxx\_import where xxx is the name of the school (ex: swu\_import)

・Character Set : utf8\_general\_ci

・MySQL Collation : utf8\_general\_ci

*2. Setting the DB User*

Server Menu of PHPMyadmin

Privileges － Add a new user. Once the user exists, select the said user and grant privilege to ImportDB

Username: xxx\_sms

Host: localhost

Password: Create one using "Generate Password" or match with /test, and match with the config file

Database for user: None

Global Privileges:

 Data - Check all items

 Structure - Check all items

 Administration - Check only LOCK TABLES



*3. Export the sampledb\_sms from 236 Server. If it's in the same server, then it's fine to copy.*

*4. Import the exported SQL to importDB*

*5. Create the import environment from the ZIP in the TechnoSMS server (Refer to upload\_guide.txt)*

*6. Create and upload Import.config*

*7. Update the DB*

 Login to the Admin and update the DB

 Run the DBKanri and the DB will be created from the Vardef

 Run db\_update

 \*There's no need to run the following tool since it has already been done in sampledb\_sms

-- For MySQL5.6, run the following tool

-- (If this hasn't been run on the server, run it on the copy in your local. Then, re-align using DBKanri)

-- http://localhost/KAIHATSU/GENEL/db\_utilities/mysql\_5\_6\_NULL\_converter.php

*8. Ask the Support to configure the schools, authorizations, etc*

**Data Transfer**

*To be prepared in advance:* The DBs to be used for the transfer in both the server and the local has to be kept the same (if the columns of the DB are different, the CSV import will be slightly off)

Always update using DBKanri and db\_update after using the tool.

*1. Create xxx\_import in Local. Export the ImportDB from the server and import it to the Local*

*2. Create the ID to be used for the school in csv-maker (This is done so that it would also be applied in the working folder).*

 Add the ID and password to the following file

 \csv-maker\data\accounts.json

*3. Access the URL - http://localhost/csv-maker/public/ (There’s no need to configure the browser)*

 Username – Password: The ones that has been created in the previous step

*7. Saving the set fields of the table*

 Click the 保存 (Save) under the title

 Save the created 設定 (Setting) and reload the important points.

＊Save File

 Save Destination: C:\wamp\www\csv-maker\SCHOOL\_FILES\xxx where xxx is school name (Ex: swu)

 \*Since the generated table name is based on the filename, please be careful when transferring multiple CSVs into the same table since it will become the same file name.

 ＊Save DB

 Add table to DB

 Run the following SQL in the Local

 http://localhost/csv-maker/lib/techno-csv/resources/tbs\_format\_config.sql

*4. Setting the loading source table*

DB Connection Settings： Assign to the table where the destination of the import will be

Ex: Localhost/root/ /swu\_import

Set the ホスト名 (Host Name), Username (ユーザ名) and Password (パスワード).

Upon click on the 更新 (Update) button, all the DB in the local would be displayed in the pulldown

Select the xxx\_import and click on the 更新 (Update) button

CSV File： Select the loading source CSV file

読み込むテーブル (Table to be loaded)： Select the loading source CSV file

作成するテーブル (Table to be created)： Select the table where the the destination of the import will be

＊In order for the CSV file (the one to be to loaded) to be in UTF8, it is needed to change its character encoding using an editor

＊In order for the loaded CSV to be copied in 「C:\wamp\www\csv-maker\data\user\(username)\tables」,

it is possible for the loaded file to be directly change rather than deleting it on the browser and be added



*5. Creating the function for the conversion of data based on the mapping table*

C:\wamp\www\csv-maker\functions\field

Create using the same name for the file name and class name

$name – Name that shows the process functions

$argument\_count – Number of arguments to be set

$CODE\_MAP – Used for character conversion mapping

apply function – Conversion function

is\_applicable – Sets the table name and columns to be imported

What has been set here will be shown in the process functions.





*6. Configuring the fields of the table*

処理関数 (Process Function)

結合 (Association) – Character string association

数値加算 (Numerical Value Addition) – Adds specific numerical value　Disregard letters

Null値 (Null Value) – Output as null

対象値の存在 (Conversion of Target Values) – Convert blank and other characters

 引数 (Argument) = Original Data, Value when blank, Value if not blank

 Ex: The blank will be converted to 1 while the others remain the same

 The引数 (argument) will be: Original Value, 1, Original Value

数値の変換 (Numerical Conversion) – Conversion of small numbers

引数 (Argument) = Original Value, Value for the ones that hasn't been set, if 0 is inputted, if 1 is inputted, if 2 is inputted...

 Ex：0 to 1, 1 to 2

 The引数 (argument) will be: Original Value, Original Value, 1, 2

空文字列は NULL (Empty Character is NULL) – When empty character, output NULL. Others will remain the same.

文字列取得 (Fetching of Character String) – Same function as Substring

printf – Same function as Sprintf　Used in zero padding

Custom Function – Process function will be shown in the specific field （function made in 5）

引数 (Argument)

固定文字列 (Fixed Character String) – Date, etc are fixed

列参照 (Column Reference) – Refer to the title names of the loading source table

行番号 (Line Number) – Line number of the CSV

＊Be careful since while the Custom Function will be shown when refreshed, 設定 (Setting) will disappear!

Once the 設定 (Setting) has been saved, the Custom Function will be shown for the reloading

＊If you bookmark the URL of the field setting of the table, the same table will be loaded.

*8. Output CSV*

The 作成する (Create) button on the lower right of the screen will output the CSV.



It will be successful if the number of lines of the original data has the same number with that of the outputted CSV

If there has been error, it will output all the data up to the one before the error occurred and the contents of the erroneous line will be a message containing an array

**Importing the DB on PHPMyadmin**

1. Select the DB and table to be imported, then click on the import tab

File to be imported: The csv generated by the csv-maker

Number of rows to skip, starting from the first row: 1 (Skip the title row)

Format of the file to be imported: CSV for the LOAD DATA

Character separating the fields: Comma “,”

End-of-line Character: \r\n

Column Name: Set this when the table to imported and the table organization used by the DB of the csv-maker is different.

While this is usually not set, it'll be good if you copy the first row of the generated CSV

2. Run Import

If the number of imported rowers and the number of rows in the CSV generated by the csv-maker are the same, it means the process is successful

3. Export the local that has been imported and perform ImportDB of the server using the exported SQL

If there are so many data, the import directly the CSV to the server

In that case, please be mindful of the column names when importing