TMSS10Sync Cookbook

TMSS10Sync Implementation

The TapeTrack Sync uses, in its minimum format, uses an executable TMSS10Sync, an input file containing the information to syncronise and a definition file (ttidef) containg the information on how to read, or translate, the information in the input file.

TTIDEF Definition File Explained

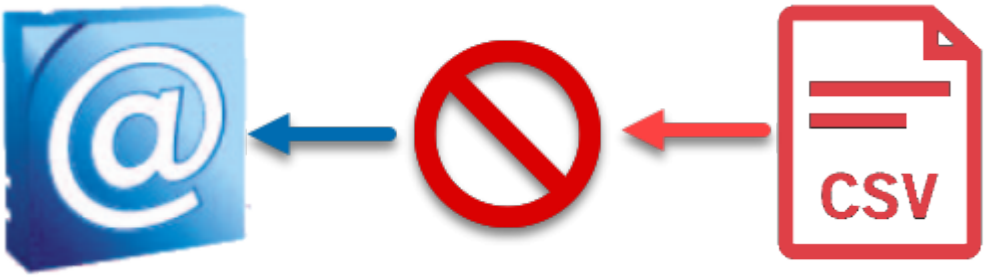
Imagine a Customer going to a bank to deposit cash into their account, a common task for most people. The problem for this customer is they only speak Spanish and the bank staff only speak English.



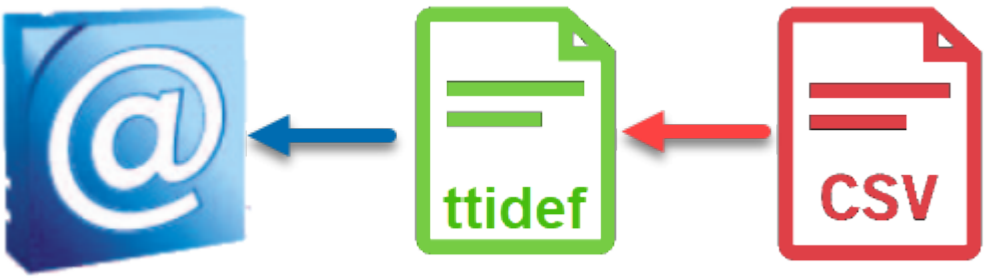
The solution to this problem is the implementation of a translator who speaks both languages. The customer talks to the translator, who passes the information to the teller in English so the money can be paced into the correct account.



This is similar the situation faced by TapeTrack Sync when importing information into the database. A text or CSV input file can contain Volume information in many formats so direct import is not possible in the raw format.



Introduce a translator between the input file and TapeTrack Sync and the information can be read from the input file and passed to TapeTrack in the correct format. This TapeTrack translator is called a definition file, a text file with a .ttidef extension.



Constructing a Definition File

Using a text editor, such as Notepad, create a definition file. This file:

- Has a file extension of .ttidef
- Should have a descriptive name, default.ttidef will load into Sync process if no parameter is set when TMSS10Sync is called.

The first step to writing the definition file is to decide what information you wish to synchronize, or import, into the Tapetrack database.

CSV Input File Format

Sample CSV File - `volumes.csv`

<sxh> GZB0001,LTO5,Full System Backup GZB0002,LTO4,GL Backup GZB0003,LTO2,Exchange Server Backup GZB0004,LTO5,Full System Backup GZB0005,LTO4,GL Backup GZB0006,LTO2,Exchange Server Backup GZB0007,LTO5,Full System Backup GZB0008,LTO4,GL Backup GZB0009,LTO2,Exchange Server Backup GZB0010,LTO5,Full System Backup </sxh>

In this comma delimited CSV file we can see the fields that we need to extract:

- Volume-ID in column 1
- Media-ID in column 2
- Volume Description in column 3

Missing from the CSV file is the Customer-ID, which will set directly from within the definition file.

Lines in the definition file that start with # are comments for explanation and are not read by the Sync engine.

Sample Definition File - default.ttidef

<sxh> # # Object Type: TapeTrack Sync Definition File # Description: Sample minimal template # System: TapeTrack # SecLevel: Propriatory # Copyright: © 2016 GazillaByte LLC # SetLiteral(CUSTOMER, “0001”); # Set the Customer-ID to a literal value of 0001 SetLiteral(MEDIA, “LTO”); # Set the Media-ID to a literal value of LTO SetCSVDelimiter(“,”); # Instruct the parser that we have a CSV file and the file is delimited with commas Extract(VOLUME, 1, 6, 0); # Extract the Volume-ID from the 1st column of the CSV with a length of 6 bytes Extract(USER1, 2, 10, 0); # Extract a user field for translation Extract(DESCRIPTION, 3, 40, 0); # Extract the Description field RemoveSpaces(DESCRIPTION); # Remove any white space from the LHS of the description # # Add L suffixes to the Volume-ID based on Media Type and Generation # AddTranslation2(VOLUME, USER1, “LTO1”, “*L1”); AddTranslation2(VOLUME, USER1, “LTO2”, “*L2”); AddTranslation2(VOLUME, USER1, “LTO3”, “*L3”); AddTranslation2(VOLUME, USER1, “LTO4”, “*L4”); AddTranslation2(VOLUME, USER1, “LTO5”, “*L5”); AddTranslation2(VOLUME, USER1, “LTO6”, “*L6”); AddTranslation2(VOLUME, USER1, “LTO7”, “*L7”); </sxh>

From:
https://docs.tapetrack.com/ - **TapeTrack Documentation**

Permanent link:
https://docs.tapetrack.com/cookbook/sync?rev=1560394460

Last update: **2025/01/21 22:07**