TMSS10Sync

Synchronize an input file or ODBC connection data with a TapeTrack Framework Server.

Synopsis

```
TMSS10Sync [-A attributeindex] [-F recordcount] [-M] [-S logonstring] [-T] [-a] [-c configfile] [-d deffile] [-l] [-p] [-s] [-t]
```

Description

TMSS10Sync is a command line utility that allows the user to run adhoc and automated synchronization between text files, CSV files and ODBC connections to update commonly used TapeTrack database fields.

In addition to being able to synchronize data, the program can also perform complex date calculation and string manipulation procedures to edit the data that is being synchronized.

The program relies on three separate software engines that are built one on top of the other.

1. **INI Parse Engine** This component is responsible for reading the Definition File, parsing the processing directives, resolving variables and including any embedded Definition File sections.
2. **Import Engine** This component is responsible for reading data from the import file or ODBC connection, extracting data into respective fields and performing any data manipulation.
3. **Sync Engine** This component responsible for updating the TapeTrack Server with any fields which have been specified to be updated.

Options

The options which apply to the TMSS10Sync command are:

- -S The logonstring argument sets the Server logon information in the format user:password@server:port.
- -c The configfile argument specifies the path to the Batch Configuration File.
- -A This argument sets the attribute index for any field imported as the DESCRIPTION field. The value of attributeindex should be an integer between 0 and 200. The default of 200 represents the inbuilt Description field. Values of 1 through 200 represent user-defined Extended Attribute values.
- -F This argument instructs the program to simply read in the contents of STDIN and write out a formatted report on STDOUT with a ruler every recordcount records. This can be helpful forestablishing the offset and length of the fields that are being synchronized from a text file. No other processing will be preformed.
- -M This argument instructs the program to run in Dump Mode. Dump mode processes the input synchronization as it would in the regular update mode but instead of performing any updates

the program will write a list of proposed update fields for each record to the STDIN. -T This argument instructs the program to set the Do Not Slot flag for each updated Volume.

- `-a` This argument instructs the program to add any new Volumes to the TapeTrack database. In the event that this argument is not specified and newly encountered Volumes will be skipped and added to the Not-Added Count.
- `-d` This argument sets the path to the Import Definition File. The Import Definition file is responsible for instructing the program on how to interpret the input file or ODBC connection. This argument is required and defaults to the file default.ttidef.
- `-l` This argument instructs the program to list all of the supported processing directives, which will be written to STDOUT. A list of valid ODBC Drivers and DSNs will also be listed. No further processing is performed.
- `-p` This argument instructs to Sync Engine to pass the value in the the VOLUME field to TapeTrack's Barcode Parsing Engine, where it will be split into the CUSTOMER, MEDIA and VOLUME fields. Barcode Parsing can also be further managed using the barcode section of the Configuration File.
- `-s` This argument instructs the program to automatically set the SLOT field to the relative record number of the record being processed.
- `-t` This argument instructs the program to run in Test Mode. In Test Mode all three engines will be invoked, but no updates will be performed. Test Mode differs from Dump Mode in that Dump Mode will not run the Sync Engine while Test Mode will, but without performing updates.



When running in Dump Mode the program will only run the INI Parse Engine and the Import Engine. It will not run the Sync Engine. As a result, any functionality that relies on the Sync Engine, such as the Barcode Parse Option will not be invoked. Further, functions that relate to data interpretation such as date parsing will also not be run.

Exit Status

- **zero** Program has ended successfully.
- **non-zero** Program has not ended successfully.

Environment

TMSSSERVERPROXY If defined the program will route all TapeTrack TCP/IP traffic through a HTTPS proxy. The value of the variable should be in the format user:password@host:port. To debug the proxy connection use variable TMSSAPILOGDIR. TMSSAPILOGDIR If defined the program will write out a trace file to this directory. TMSSPWPATH When no password value is passed in the logon string the program will look for the password in file \etc\tapetrack\user, where user is the user value passed in the logonstring. If you wish to change this default path, you can set the path in TMSSPWPATH. TMSSNOMD5 If defined the program will not MD5 hash passwords before sending them to the TapeTrack Server. This is required when relying on Windows Active Directory authentication. It should be noted that although the password is not being hashed, it is still being encrypted during transmission.

Files

stdout Formatted output, Record Dump output or Scan File Output. stderr Diagnostic messages. stdin Sync input file.

Examples

Example 1. TMSS10Sync: List supported Processing Directives. [gnicol@z1090 ~]$ TMSS10Sync -l < syntax.txt Example 2. TMSS10Sync: Run program in Dump Mode. [gnicol@z1090 ~]$ TMSS10Sync -M < input.txt Example 3. TMSS10Sync: Produce a formatted listing with a ruler every 10 records. [gnicol@z1090 ~]$ TMSS10Sync -F 10 < input.txt > output.txt

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

Permanent link:
https://rtfm.tapetrack.com/cli/tmss10sync?rev=1497730308

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