wd

**MMSLOADER Report**

**Requirements Specification**

**Revision: 1.0**

**Prepared by**

**MUHAMMAD ASIM NAEEM**

**17 October 2017**

REVISION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| Version Number | Items/Description of change | Date Updated | Updated By |
| 1.0 | Initial Release | 17 July 2017 | Shahid Kochak |
| 1.0 | Enhancements | 06 Oct 2017 | M. Asim Naeem |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

CONTENTS

1 OBJECTIVE 4

2 Process Flow 5

3 Specifications 6

3.1 Requirements 6

**3.2** **Dispatcher** 8

3.2.1 Configuration 8

3.3 **LOADER** 9

3.3.1 PRE-REQUISITE 9

3.3.2 CONFIGURATIONS 10

3.3.3 TEMPLATE CONFIGURATION 11

**3.4** **Current Setup** 22

3.4.1 System 22

3.4.2 Database 22

3.4.3 Configuration 22

3.4.4 Capabilities 23

**3.5** **Current Table List** 23

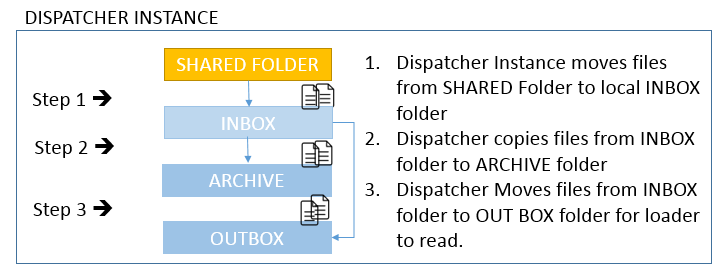
**3.6** **Conditions and Rules:** 24

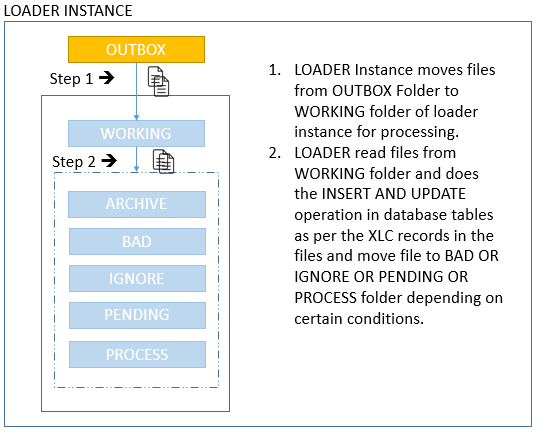
**3.6.1** **Sequential / Serial Processing** 24

**3.6.2** **Parallel Processing** 25

# OBJECTIVE

The main objective of this application is it to read **XLC RAW** files (CSV,TXT etc) and load data to respective tables depending on the configured templates. By considering the volume of XLC RAW files, multiple instances of the application can be executed in parallel.





# Process Flow

The process flow of the XLC Loader application is as follows:

File Repository

Config DB

Files

Insert data

Read CSV

Processed

Load Templates

Pending

Bad

Map data to template

Ignore

Template Not found

Successful

Working

Data DB

Corrupt

Connection Issue

Following are the process steps:

* Move files from file repository to working folder.
* Read the CSV data from files
* Load template configuration from Database.
* Map the CSV data to the template configuration.
* If required XLC template not found
  + Move the file to Ignor folder
* If template found proceed to data insertion
* Insert the data to Database according to the template configurations.
* If the data insertion successful
* Move the file to processed.
* If the data insertion not successful due to unavailability of database connection
  + Move the file to pending folder
* If the data insertion into database is not successful due to any other reason
  + Move the file to Bad folder

# Specifications

## Requirements

Following are the basic requirements for Loader application

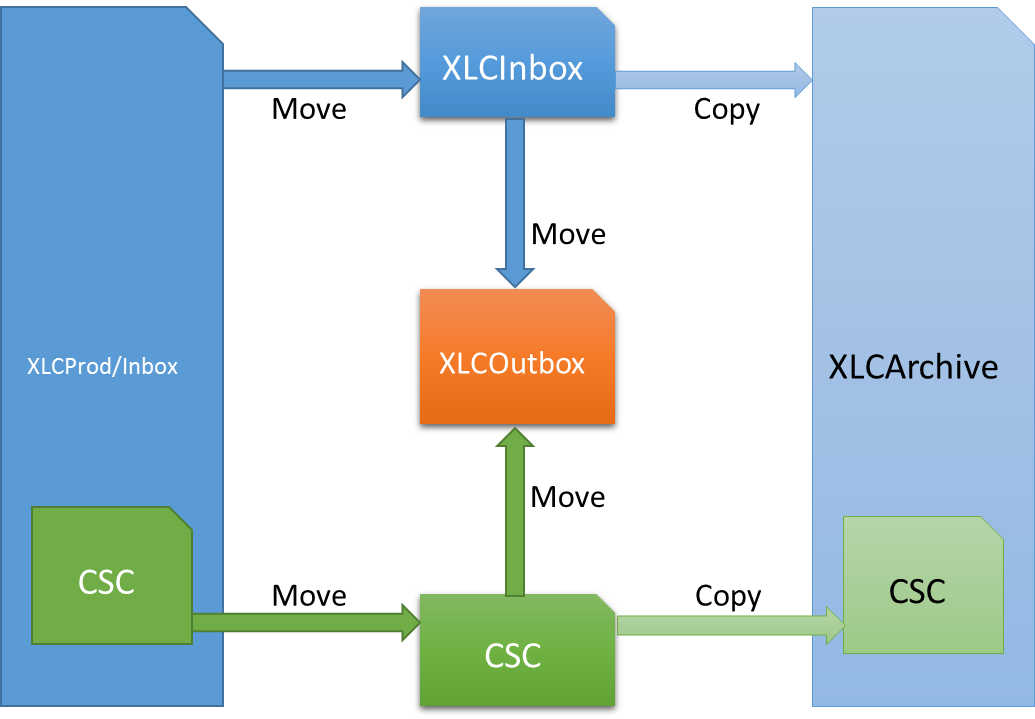
1. Loader should be able to do below task
   1. Able to create folder structure per process as follows.
      1. **WORKING**
      2. **PENDING**
      3. **PROCESS**
      4. **BAD**
      5. **IGNORE**
      6. **ARCHIVE**
   2. Able to save configuration data e.g. templates, server connections etc. to different database.
   3. Able to configure templates per **XLC** **RAW** (Csv,txt etc.)files and table.
   4. Each template must have specific associated server connection.
   5. Able to save connections of different database servers.
   6. Able to perform file processing in parallel.
   7. Able to run multiple instance of application simultaneously.
   8. Able to log all activities performed by the application.
   9. Able to read **XLC RAW** files (CSV,txt etc..) and do below operations
      1. Read files and insert or update data to respective tables as per template configuration
      2. After reading file successfully it should move files to certain folders as per below conditions.
         1. If loader is able to read file without any issues file should be moved from **WORKING** to **PROCESS** folder.
         2. If there is some connection failure while reading a file, File should be moved from **WORKING** to **PENDING** folder.
         3. If XLC is not configured in loader database then file should be moved from **WORKING** to **IGNORE** folder.
         4. If there are some records which are **corrupt or mismatch** (**colums not matching with template columns or datatype of columns not same as defined in template**) then those records should be saved as file and moved to BAD folder.
   10. Able to log all exceptions thrown by application.

## **Dispatcher**

Dispatcher is intended to move the files from main shared folder to the OUTBOX folder. So that the Loader can read the files and process these.

### Configuration

Following are the configurations for different dispatchers



* Dispatcher 1
  + **Step 1**:

Move files from XLCProd/Inbox folder to XLCInbox folder

* + **Step 2**:

Copy files from XLCInbox folder to XLCArchive

* + **Step 3**:

Move files from XLCInbox folder to XLCOutbox

* Dispatcher 2
  + **Step 1**:

Move files from XLCProd/Inbox/CSC folder to CSC folder

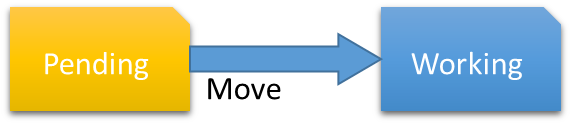
* + **Step 2**:

Copy files from CSC folder to XLCArchive/CSC

* + **Step 3**:

Move files from CSC folder to XLCOutbox

* Dispatcher 3
  + Move files from Pending folder to Working folder. So the files can be processed again.



## **LOADER**

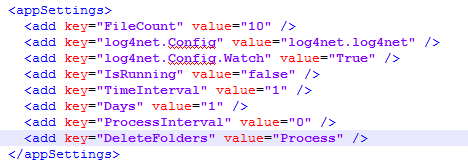
### PRE-REQUISITE

* + 64 BIT OR 32 BIT WINDOWS OPERATING SYSTEM
  + DOTNET FRAMEWORK 4.5 or above.
  + MICROSOFT ACCESS DATABASE ENGINE **(included in Setup)**
  + WD.DATACCESS DLL **(included in Setup)**
  + WD.DOMAIN DLL **(included in Setup)**
  + Log4net DLL **(included in Setup)**
  + ORACLE MANAGED DATA ACCESS DLL **(included in Setup)**
  + METRO FRAMEWORK DESIGN DLL **(included in Setup)**
  + METRO FRAMEWORK DLL **(included in Setup)**
  + METRO FRAMEWORK FONTS DLL **(included in Setup)**
  + NEWTONSOFT JSON DLL **(included in Setup)**

### CONFIGURATIONS

* + App Settings

Following are the required settings in AppSetting portion of configuration file:



* + - FileCount

Its the integer value which represents the maximum number of files processed by application simultaneously.

* + - Inbox

It is the main repository of files to be processed. Set the value as absolute path of inbox folder. The application will move files from Inbox folder to its working folder for further processing.

* + - Log4net.Config

It is the configuration file name for Log4Net.

* + - IsRunning

If set as true, on application execution, the application will directly start the processing of CSV files.

* + - TimeInterval

It is the time between two cycles of simultaneous file processings. The value is specified in seconds.

* + - Days

Its is the value in number of days to delete the files from specified folders. All the files older than (in days) this value are deleted.

* + - ProcessInterval

Hours from mid-night to start the deletion process. During this process, all the old files are deleted from specified folders. The value represents number of hours.

* + - DeleteFolders

It’s a comma separated list of folder names from which the old files need to be deleted. Deletion process is executed by midnight.

* + Log4net Configuration

Following are the settings required in log4net configuration file.

* + - Specify the path of error log file and activity log file.
    - User can select any of the folder on the machine.
    - Make sure the folder must exist. However, the files can be created by application if not already exist.



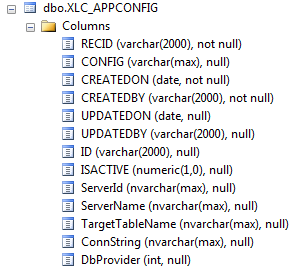
### TEMPLATE CONFIGURATION

* + Testing environment is having below credentials.
    - **Server Name** : 172.21.12.73\INPXMLD2MY
    - **DBName** : MMSDB
    - **Table Name**: XLC\_AppConfig
    - **User Id** : XXXX
    - **Password** :XXXXX
  + XLC RAW file sample Data

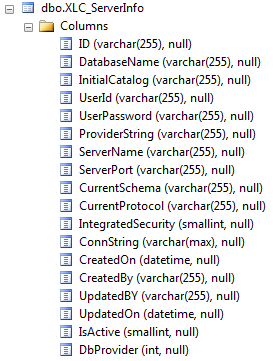
**In below table each column header is represented as F1,F2…FN**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| XSU01 | 6/19/2017 8:08 | 6/17/2017 3:08 | 1/1/1970 8:00 | EM6A1 | NEPTUNE | 13 | 58 | 0 | WXR1A478PZ0T | 10.0.49.0 | **…………** |
| XSCC01 | 6/19/2017 8:09 | 6/17/2017 3:08 |  | EM6A1 | NEPTUNE | 13 | 58 | 0 | WXR1A478PZ0T | XCFGS2 | **………..** |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

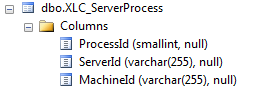
* + Each column in raw file is named as F1, F2, F3 and so on. And then map to the database table fields.
  + To configure templates for XLC Raw file(s) we have below table structure and JSON format for storing MAPPING information per XLC.
    - TABLE STRUCTURE
      * XLC\_AppConfig



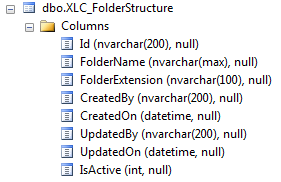
* + - * XLC\_ServerInfo



* + - * XLC\_ServerProcess



* + - * XLC\_FolderStructure



* + - MAPPING JSON

Example

{

"Columns": [

{

"Index": 1,

"ColumnName": "F1",

"DataType": "string",

"DefaultValue": "",

"FieldName": "REC\_ID",

"Format": "",

"Length": "8",

"IsPrimary": false,

"iLength": 8

},

{

"Index": 2,

"ColumnName": "F2",

"DataType": "datetime",

"DefaultValue": "",

"FieldName": "P\_DATE",

"Format": "",

"Length": "16",

"IsPrimary": false,

"iLength": 16

},

{

"Index": 3,

"ColumnName": "F3",

"DataType": "datetime",

"DefaultValue": "",

"FieldName": "ORA\_CREATE\_DATE\_TIME",

"Format": "",

"Length": "16",

"IsPrimary": false,

"iLength": 16

}

],

"Schema": null

}

* + - File Deletion

Loader application is able to delete the unwanted files from specied folders like Processed, Bad or Ignor. On average the loader application is able to delete 13000 files per second. The deletion process requires 3 informations to be executed successfully

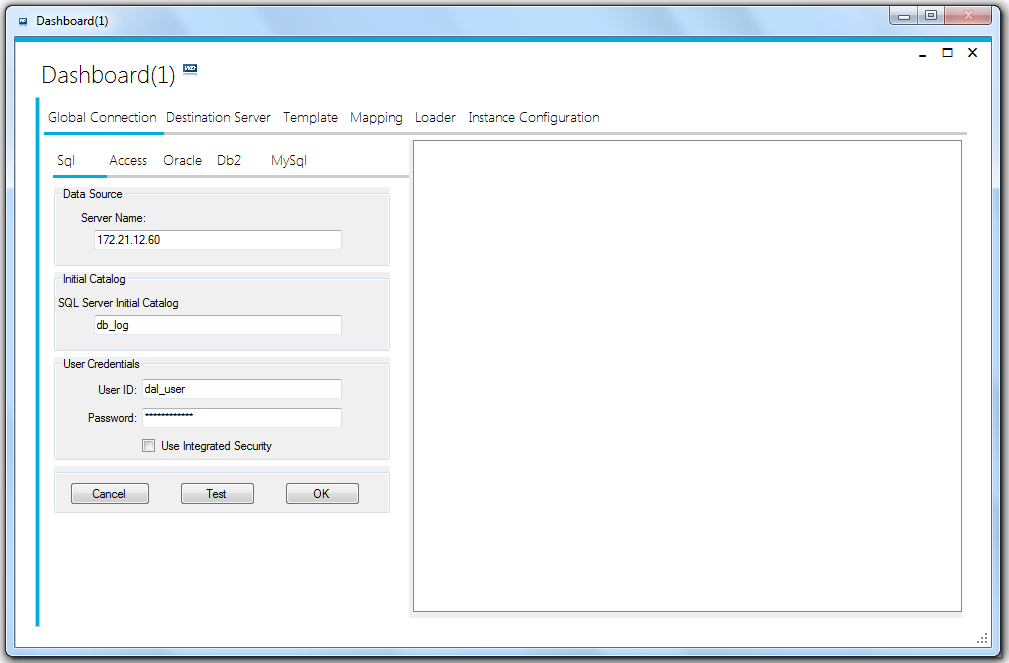
**WHAT**: It is important to know which files to be deleted or how older files to be deleted. It can be configured in configuration “Day” in app.config which may have integer value. The files older than (in number of days) the specified value are to be deleted.

**WHEN**: It is important to know when the deletion activity should be performed. By default it is performed at midnight. However, there is a configuration property “ProcessInterval” in app.config which is an integer value. It is the number of hours from midnight. This property may have either positive value or negative value. 0 menas exactly at midnight.

**WHERE**: It is also important to provide the location from where the files to be deleted. For this purpose there is a configuration property “DeleteFolders” in app.config. This property have string value which contains the comma separated list of folders from where the files to be deleted.

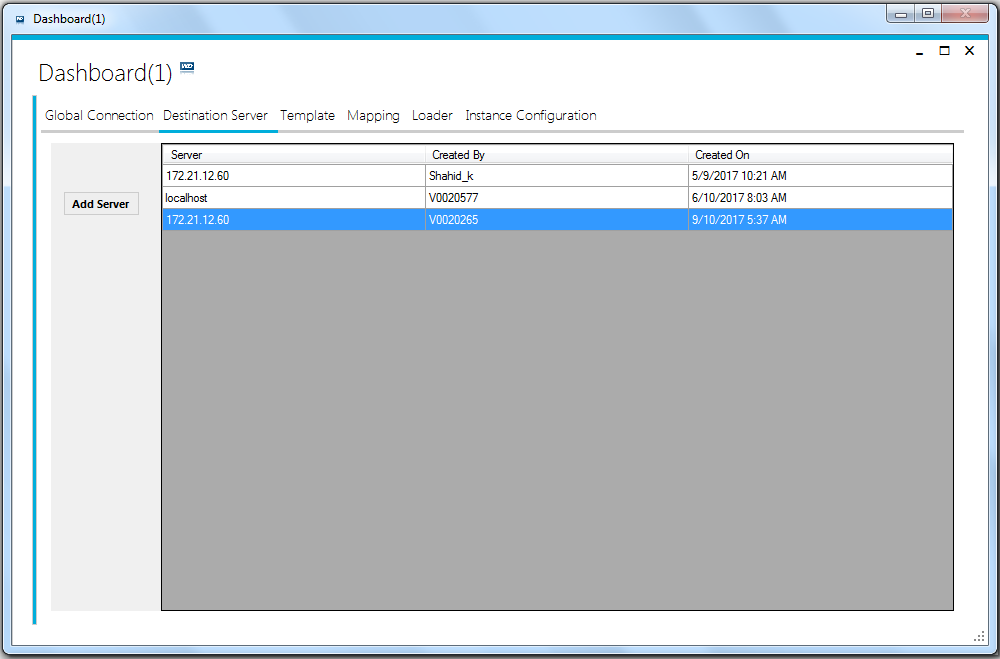
* + - SCREEN SHOTS TO SHOW UI FOR TEMPLATE CREATION, MAPPING AND EXECUTION.
      * GLOBAL CONNECTION

Global connection screen is used to save the connection for Config DB, where all the configuration data will be saved; like template information, DB server connections etc.

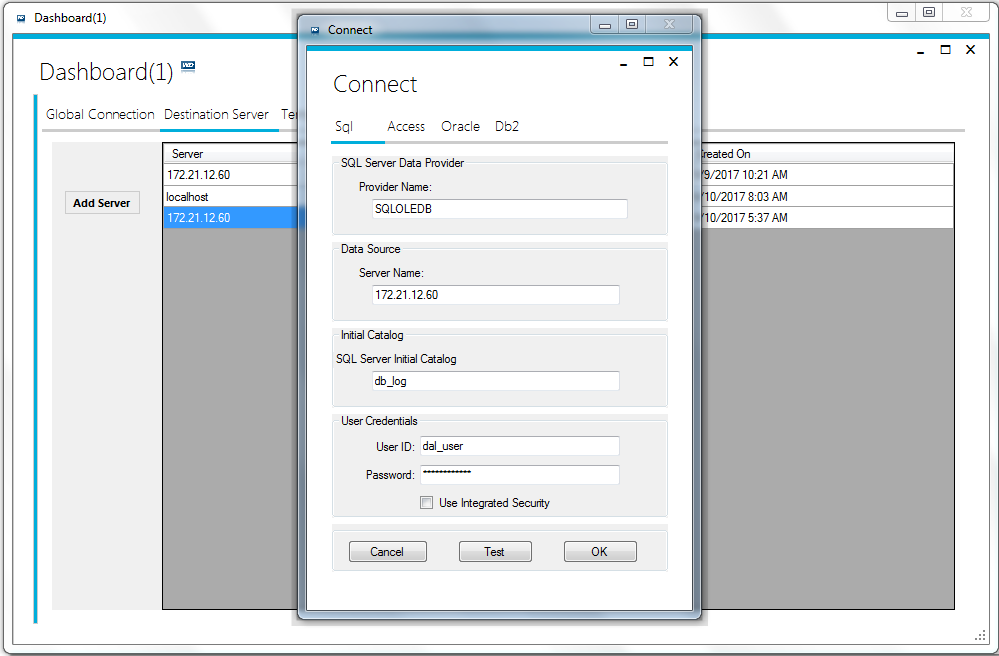


*Global connections*

* + - * DESTINATION SERVER

In the below screen you can create new database server connection, which can be used while creating templates. We can create multiple database connections.

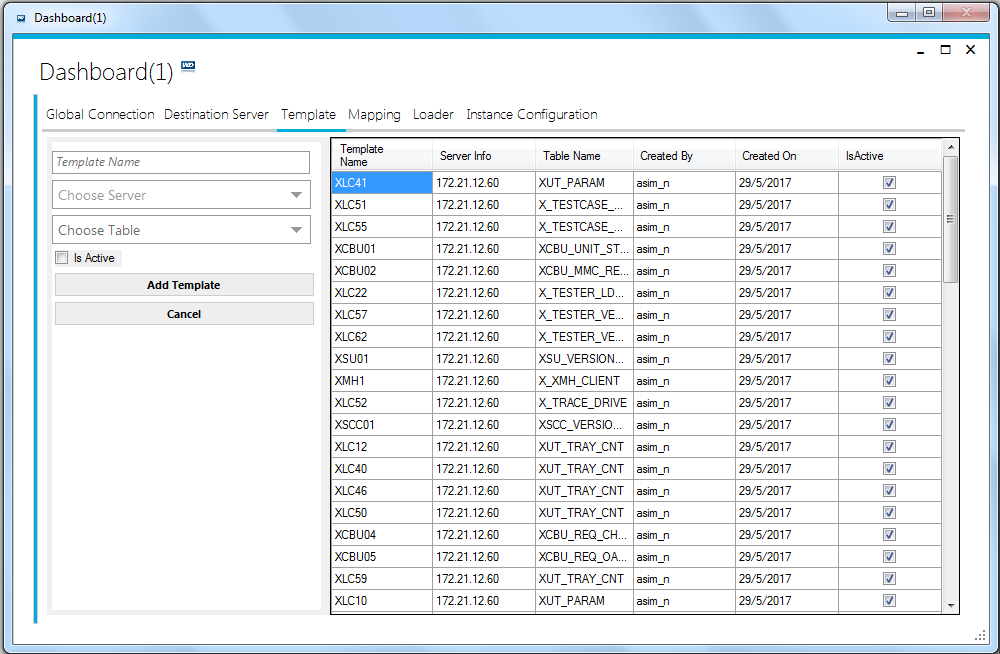
*List of destination databse servers*

We can add new database connection by clicking on “Add Server” button.

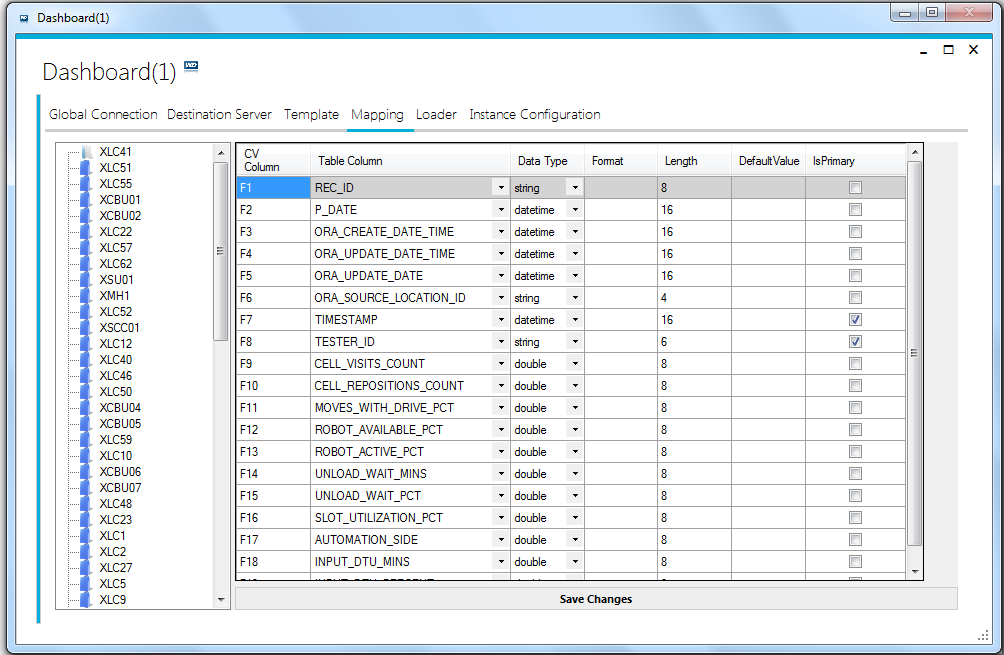
Fill the appropriate information and click “OK” to save the connection.

* + - * TEMPLATE CREATION UPDATION AND LIST VIEW

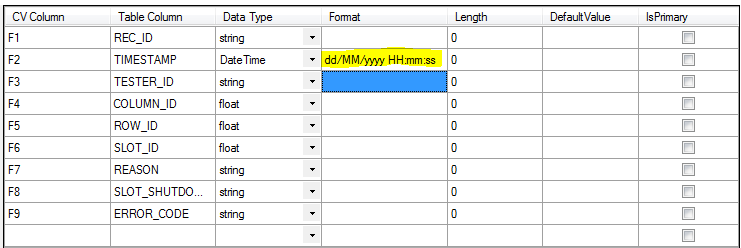
**In below screen you can create / edit update template and view list of templates created. (More information in User Guide)**



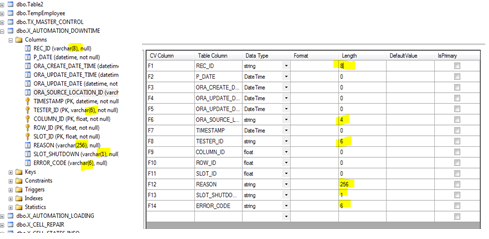
* + - * TEMPLATE MAPPING

**In below screen left side of screen shows list of templates and right side shows mapping attributes as per template selected.(More information in User Guide)**

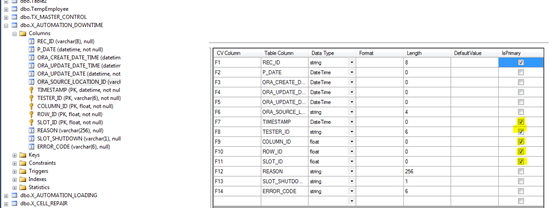
1. “CV Column” is the order of values in CSV file been read by application.(Mandatory)
   1. Each value must have prefix “F”.
   2. After “F”, the number is the location of the value in row of CSV file.
   3. The number must be unique for each row in this column. Duplicate values may result error during file processing.
   4. “F1” must always be associated with “REC\_ID” column of database table.
   5. “F1” cannot be change or deleted.
2. “Table column” is the column name of associated table. That column is attached to that particular value of CSV file. (Mandatory)
3. “Data Type” Column is to define the data type of the values.(Mandatory)
4. “Format” column is to define the format of the value read from CSV file.



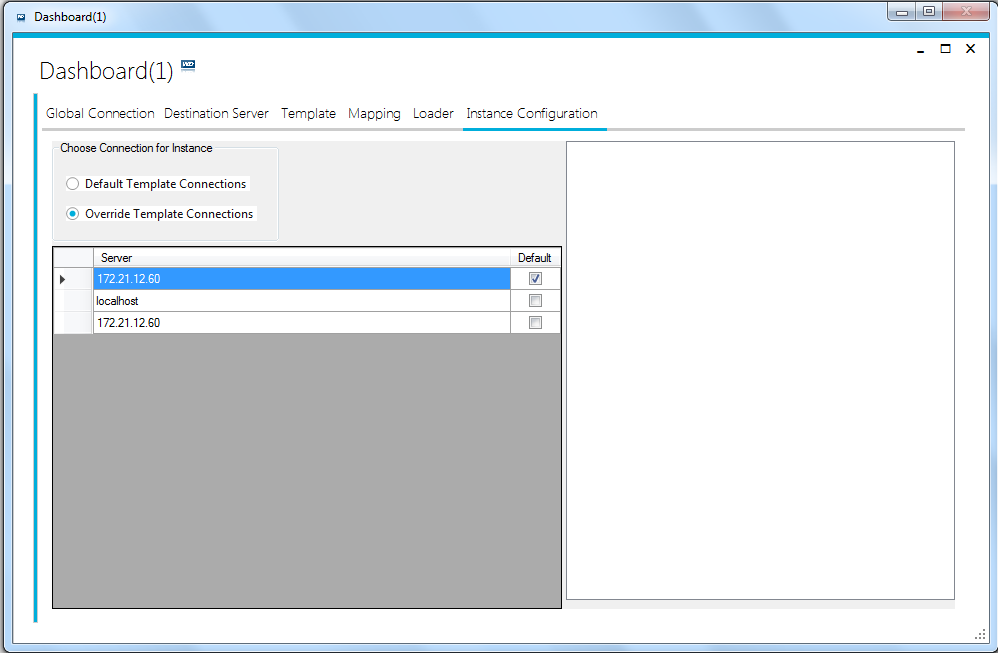
1. “Length” column is to define the character length of the value to be inserted in the database. It is mandatory for columns with string datatype. The value must be exactly the same as the length of the associated column of table defined in the database.



1. “Default Value” column is to define the default value if any. This value will be inserted into database in case if there is no value exist in CSV file for this column.
   1. In case, if there is any default value defined for that particular column in database, it will be over-written with the default value set in XLC loader application.
2. “Is Primary” column is to set if the column has primary key or part of unique key index in database.
   1. Mandatory if the column has primary key or unique index key.
   2. “REC\_ID” column will always be part of primary key.
   3. These values are essential to identify & update the existing row in database.



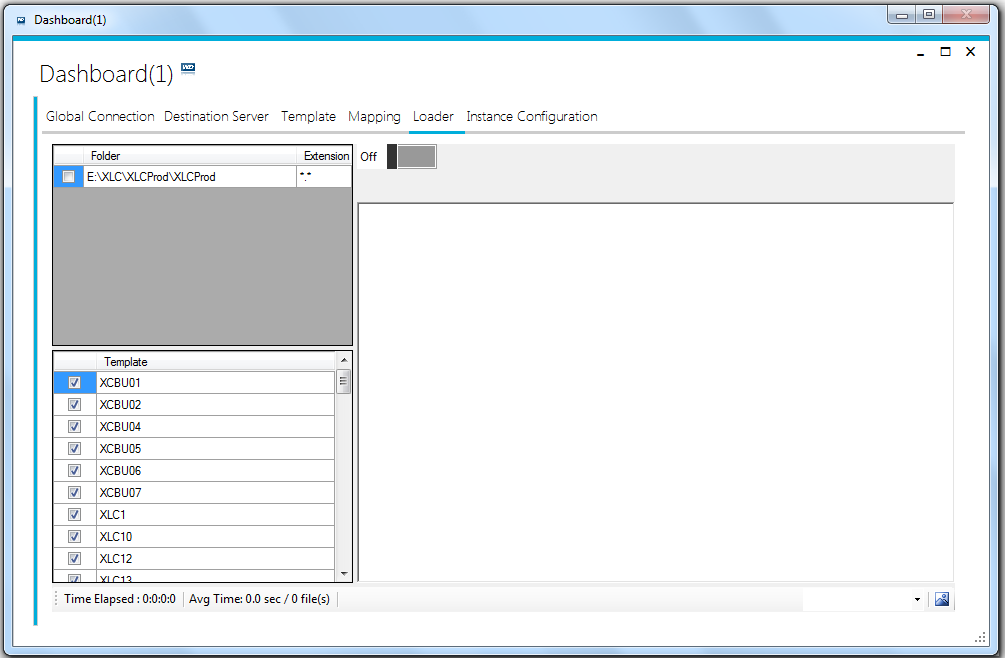
* + - * INSTANCE CONFIGURATION

In below screen there are two options to chose for CSV file data saving. Either we can use the default connection associated with each template or we can override the default connection and chose some different connection for data saving. This setting is only valid for current instance of application.

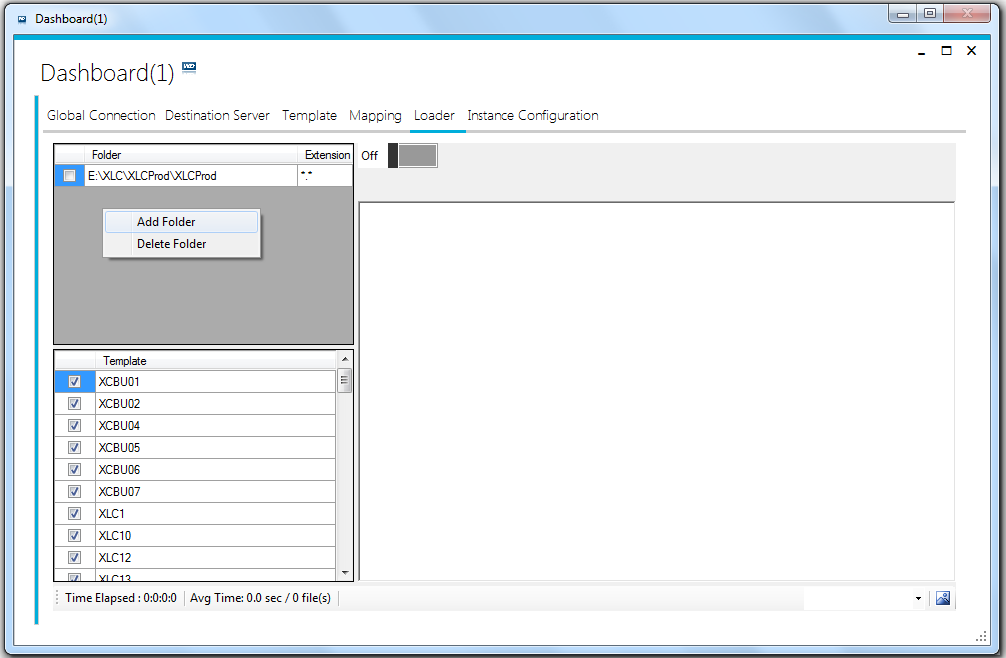
The selected database server connection will over the default connection associated with each template. And all the CSV file data will be saved as per selected database server conection.

* + - * LOADER EXECUTAION SCREEN

**In below screen left side of screen shows list of folders and list of templates.You need to select template and folders and press toggle on to process files. (More information in User Guide)**



* XLC\_FolderStructure



* + Right click on
* **XLC\_AppConfig table** is main table where we store XLC template configurations.
* **RecID** column stores XLC template RecIDs eg XLC53,XLC54
* **Config** column stores the mapping data information as json string.

**Mapping information details contains below information**

* + Table information for Template where data will be inserted or updated.(**TargetTableName**)
  + Mandory **(CV COLUMN AND TABLE COLUMN)** Columns mappings of **XLC RAW** file columns with table columns.**(ColumnName, FieldName)**
  + Madatory **Data type** as per database table column.**(DataType)**
  + Optional **Length** of column as per database table column.**(Length)**
  + Optional **Data Format** of column as per database table column.**(Format)**

Eg (DateTime : YYMMDD or yyyy-MM-dd).

* + Optional **Default Value** as per database table column.**(DefaultValue)**
  + Unique or **Primary** information as **Boolean.(IsPrimary)**
* **CreatedOn** column stores date time, when Template has been created.
* **CreatedBy** column stores user id.
* **UpdatedOn** column stores date time, when Template get updated
* **UpdatedBy** column stores user id.
* **IsActive** column reperesents active templates.
* **ID** columnis unique identifier for AppConfig table.
* **ServerId:** column stores the server id string
* **ServerName:** Unique name or IP of the database server
* **TargetTableName:** Name of the target table where the data for this template will be stored.
* **ConnString:** complete connection string of database server
* **ProviderType:** value of DbProvider. Like SQL, ORACLE, DB2 etc.

## **Current Setup**

The following are the details of current loader setup

### System

* Server IP: 172.21.12.167
* OS: Windows server 2008 R2 standard (64bit)
* Installed memory: 8GB
* Active Directory\User : MY\MitecsRpt

### Database

* DBType: Microsoft SQL Server
* Server\Instance: 172.21.12.73\INPXMLD2MY
* Database: mmsdb
* User Id: mmlad

### Configuration

* Loader Instances running: 4
* File dispatcher Instances Running: 1
* File Count: 10
* Inbox Folder: D:\XLCOUTBOX

### Capabilities

The loader capabilities can be measured as

* **Files/Hour:**

Loader can process on average 900 files per hour. However these figures may vary on the basis of file size.

* **File Deletion:**

Loader application is capable of deleting 13000 files per second.

* **Instance Count:** 4,

Can start more instances of Loader applications. It depends if files coming to outbox folder in one hour are more than the files processed by running instances per hour.

* **Max Instances:**

As many as required instances can be executed simultaneously. However, it is strongly recommended to only execute the instances as per requirement. So that there should not be extra overhead on resources for idle instances of loader.

## **Current Table List**

* + - * X\_AUTOMATION\_DOWNTIME
      * X\_AUTOMATION\_LOADING
      * X\_CELL\_STATES\_INFO
      * X\_CELL\_STATUS
      * X\_DRIVE\_MOVEMENT
      * X\_DTU\_STATUS
      * X\_I\_OPTIMUS
      * X\_I\_RADAR
      * X\_MITECS\_STAT
      * XSCC\_VERSION\_INFO
      * X\_STAT\_TESTER
      * X\_SYSTEM\_STATUS
      * X\_XMH\_CLIENT
      * XSU\_VERSION\_INFO
      * XUT\_TRAY\_CNT
      * X\_TESTER\_VERSION\_INFO
      * X\_TRACE\_DRIVE
      * XCBU\_REQ\_CHANNEL\_DIAG
      * XCBU\_MMC\_REQUEST\_STATE
      * XCBU\_REQ\_ALL\_UNIT\_STATUS
      * XCBU\_REQ\_OALL\_STAT\_29M
      * XCBU\_REQ\_OALL\_STAT\_33M
      * XCBU\_REQ\_OALL\_STAT\_37M
      * XCBU\_UNIT\_STATUS
      * X\_TESTER\_LD\_DIST
      * X\_IDLE\_DRIVE
      * X\_ROBOT\_IDLE\_TIME
      * X\_SLOT\_DOWNTIME
      * X\_CELL\_REPAIR
      * X\_DRIVE\_MODEL\_QTY
      * X\_DTU\_CYCLE\_TIME
      * X\_TEST\_UTILIZATION
      * XUT\_PARAM
      * X\_TESTCASE\_TIME

## **Conditions and Rules:**

The following are the filter conditions for data mapping and manipulation :

1. Create Templates for XLCs.
2. Map columns for XLC with Prefix as F and Suffix interger e.g F1,F2

**Note**

**F1 with Table Column as Rec\_ID is mandatory.**

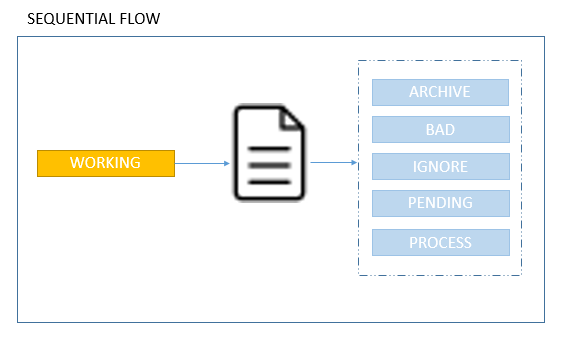
1. Start Loader for executing files.Process will decide whether to process files Parallel or Sequential, depending on **No. of files** to be processed and on the number of processors on the current machine.

**Max number of files processing parallel is 10.**

### ***Sequential / Serial Processing***

* Processing that occurs in sequence and not altogether.

**In below screen one file is processed at a time.**



### ***Parallel Processing***

We are dealing with 2 types of parallel processing.

* **Multiple Instances**: As XLC Loader is a multi-instance application; more than one instances can be executed simultaneously. Each instance of application is numbered uniquely and process a unique set of raw files.
* **Data parallelism** refers to scenarios in which the same operation is performed concurrently (that is, in parallel) on elements in a source collection or array. In data parallel operations, the source collection is partitioned so that multiple threads can operate on different segments concurrently or in simple words data processing wherein at least two or more chains of functions are carried out at the same time by independent processors.
* **MMSLOADER PROCESSES 10 FILES MAX IN PARALLEL**

**In below screen Max 10 files are processed concurrently at same time**

