**Technical Documentation**

**LoadData Program**

Table of Contents

[Overview 3](#_Toc508010496)

[Scalable Architecture for Data Loads 3](#_Toc508010497)

[C# .NET Program Usage – LoadData.cs 4](#_Toc508010498)

[Application Database Server Programs 5](#_Toc508010499)

[Application Database Server Objects – For Configuration, Logging and Monitoring Runs 5](#_Toc508010500)

[Metadata and Logging Tables 5](#_Toc508010501)

[Installation 7](#_Toc508010502)

## Overview

Most places have a need for various data loads (i.e. a SQL query that is ran against a source database table(s) that then loads a destination table). The approach usually is to create separate scripts, programs, SSIS packages, or lots of data flow tasks in a SSIS package to load the data. This approach is not scalable, and you basically have the same code (i.e. except the source system database connection string, query to run, destination table) duplicated for each individual request. So there is an explosion of scripts and/or SSIS packages that perform an individual data load.

Instead a metadata driven approach to ETL is needed. Where one has a single program that reads a table of ETL definitions. This table includes some of the following key items:

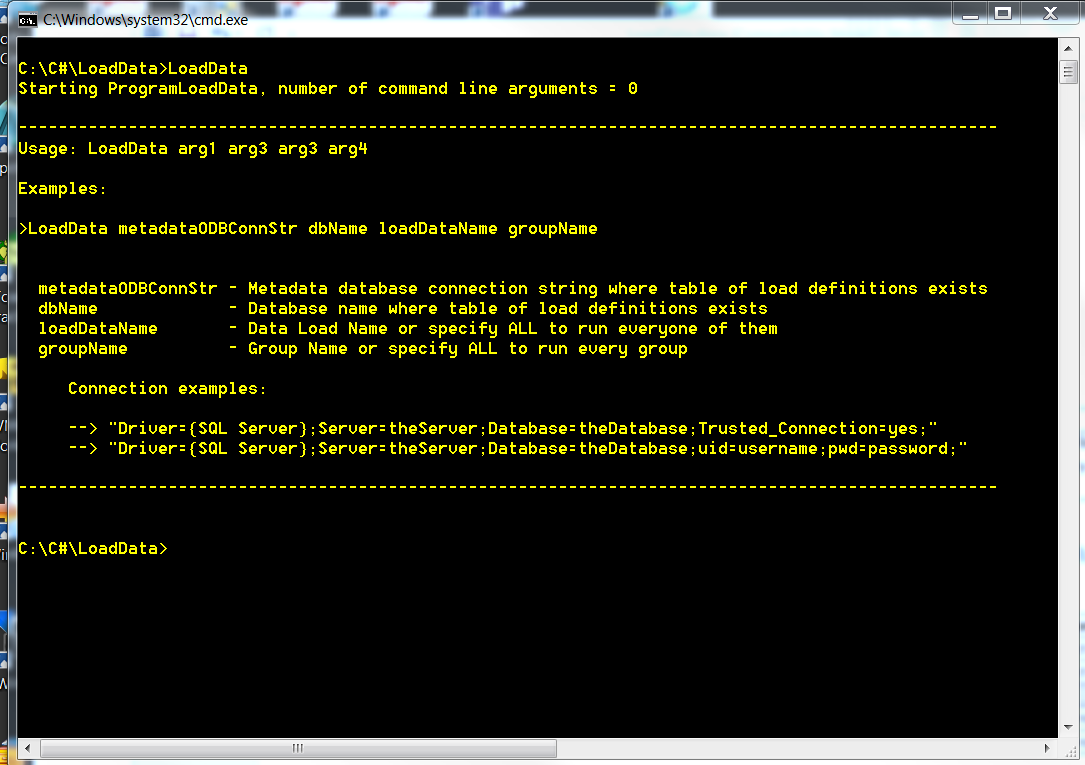
* Source system database to connect to
* Query to run against the source system
* Destination Table

## Scalable Architecture for Data Loads

This program is table driven, i.e. where the user defines the load definition in a table, and scalable because one does not have to create a separate SSIS package or have numerous data flow tasks in a single package. The major limitation of SSIS is that the data flow task needs to be bounded at design time (i.e. the query mapped to the destination table columns). Whereas a table-driven process is configured via a database table and then the load process dynamically connects to a source system database, extracts the data (performing transformations in the SQL statement or just a straight select) and loads it into a destination table (needs to be separately created).

In addition, any further transformations or incremental load merging can be done via a separate database stored procedure, after this program has loaded the data into a staging table. So this method works well in an ELT framework where you Extract, Load, and Transform (in the database).

## C# .NET Program Usage – LoadData.cs



## Application Database Server Programs

|  |  |
| --- | --- |
| **Program or Library** | **Purpose** |
| LoadData.cs | Generic data load program. Reads table and loads data based on input SQL query. |

## Application Database Server Objects – For Configuration, Logging and Monitoring Runs

|  |  |  |  |
| --- | --- | --- | --- |
| **Object Type** | **Schema** | **Object Name** | **Description** |
| Table | dbo | LoadDataDef | Data load definitions |
| Table | dbo | LoadDataOut | Data load run logs |
| Procedure | dbo | LogBegLoadData | Logging procedure |
| Procedure | dbo | LogEndLoadData | Logging procedure |

## Metadata and Logging Tables

**LoadDataDef** – Holds data load definitions

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Description** |
| LoadDataDefKey | NUMERIC | Identity – system generated sequence. |
| LoadDataName | VARCHAR | Data load name, must be unique. |
| SourceConnType | VARCHAR | Source system database type (i.e. MSQL, OLEDB, ODBC). |
| SourceConnStr | VARCHAR | Source system database connection string. |
| SourceQuery | VARCHAR | Query to run against source system database. |
| DestConnStr | VARCHAR | Destination database connection string. |
| DestDbName | VARCHAR | Destination database name. |
| DestSchemaName | VARCHAR | Destination schema name. |
| DestTableName | VARCHAR | Destination table name. |
| LoadType | VARCHAR | Table load type (i.e. FULL, APPEND). |
| BatchSize | BIGINT | Array batch size for rows committed. |
| Directory | VARCHAR | Directory name, i.e. if a file based load. |
| FilePattern | VARCHAR | File pattern, i.e. if a file based load. |
| Worksheet | VARCHAR | Excel worksheet name, i.e. for excel file load. |
| Header | VARCHAR | Header file (Y or N), i.e. for flat file based load. |
| Delimter | VARCHAR | Delimiter, i.e. if a file based load. |
| ActiveFlg | VARCHAR | Enables or disables (i.e. Y or N). |
| GroupName | VARCHAR | Extract Group Name, can be null. |

**LoadDataOut** – Holds logging of extract runs.

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Description** |
| LoadDataOutKey | NUMERIC | Identity – system generated sequence. |
| LoadDataName | VARCHAR | Data load name, must be unique. |
| ProgramName | VARCHAR | Program that logged. |
| SqlStatement | VARCHAR | SQL Statement executed against source system database. |
| BegDateTime | DATETIME | Beginning extract time. |
| EndDateTime | DATETIME | Ending extract time. |
| Status | VARCHAR | SUCCESS, FAILURE, or RUNNING. |
| NumRecs | NUMERIC | Number of records written to extract file (does not include header if present). |
| DestConnStr | VARCHAR | Destination database connection string. |
| DestDbName | VARCHAR | Destination database name. |
| DestSchemaName | VARCHAR | Destination schema name. |
| DestTableName | VARCHAR | Destination table name. |
| LoadType | VARCHAR | Table load type (i.e. FULL, APPEND). |
| BatchSize | BIGINT | Array batch size for rows committed. |
| Directory | VARCHAR | Directory name, i.e. if a file based load. |
| FilePattern | VARCHAR | File pattern, i.e. if a file based load. |
| Worksheet | VARCHAR | Excel worksheet name, i.e. for excel file load. |
| Header | VARCHAR | Header file (Y or N), i.e. for flat file based load. |
| Delimiter | VARCHAR | Data extract file field delimiter. |
| Filename | VARCHAR | Name of file that got loaded, i.e. if file or Excel based load. |
| LoadDataDefKey | NUMERIC | Optional implied foreign key to LoadDataDef table. |
| ErrorMsg | VARCHAR | Error message if applicable, null if no errors. |

## Installation

The following programs/scripts are provided:

|  |  |
| --- | --- |
| **Program/Script** | **Purpose** |
| InstallLoadDataObjects.bat | BAT file to install database objects, i.e. in SQL Server database for metadata and logging tables. |
|  |  |
| LoadData.cs | C# .NET program (compile in Visual Studio or with csc.exe at command line). |
|  |  |
| LoadDataDef.sql | Table of data load definitions. |
| LoadDataOut.sql | Logging table. |
| LogBegLoadData.sql | Stored procedure for logging. |
| LogEndLoadData.sql | Stored procedure for logging. |