

MapInfo EasyLoader

Version 17.0

User Guide



Table of Contents

1 - Introduction

Database Support and Requirements	4
Getting Started	5

2 - Uploading Data

Understanding Your Data	9
Uploading Tables	12
About Upload Options	17
Using the Map Catalog	19

3 - Productivity Aids

Shortcuts by Keystroke	23
Using Command-Line Flags to Run MapInfo EasyLoader	24
Command-Line Flag Descriptions	26
Locating the MapInfo EasyLoader Log File	37
Locating the MapInfo EasyLoader Configuration File	37

4 - History of Changes

New in MapInfo EasyLoader 17.0	39
New in MapInfo EasyLoader 16.0	39
New in MapInfo EasyLoader 15.2	40
New in MapInfo EasyLoader 15.0	40
New in MapInfo EasyLoader 12.5.1	41
New in MapInfo EasyLoader 12.5	41
New in MapInfo EasyLoader 12.0	41

1 - Introduction

MapInfo EasyLoader is a utility to upload MapInfo® TAB files to a remote database. The spatial information in the TAB files is maintained in the remote database and is available for viewing and analyzing in MapInfo® Pro.

For a list of what is new in this release of MapInfo EasyLoader, see [History of Changes](#).

In this section

Database Support and Requirements	4
Getting Started	5

Database Support and Requirements

MapInfo EasyLoader uploads data to these databases:

- Oracle
- SQL Server
- Microsoft Access
- PostgreSQL / PostGIS

The database must be able to handle spatial geometry. If the database does not have spatial object type support, then MapInfo EasyLoader only uploads XY data for a table: XY or XY with a MapInfo Key (MICode). MapInfo EasyLoader opens only one server connection at a time.

MapInfo EasyLoader is available as a download on the web or it installs with MapInfo Pro into the \Tools directory.

Software Requirements

To work correctly, MapInfo EasyLoader requires the following:

- Microsoft .NET Framework 4.6.1 (KB3102433)
<https://www.microsoft.com/en-us/download/details.aspx?id=49981>
- Microsoft Visual C++ 2015 Redistributable (x64) 14.0.23918 (vc_redist.x64.exe)
<https://www.microsoft.com/en-us/download/details.aspx?id=51682>

MapInfo Pro installs these requirements. You do not need to install these when running MapInfo EasyLoader on the same machine as MapInfo Pro.

System Requirements

This product is tested on the following Microsoft Windows Desktop Operating Systems:

- Windows 10 64-bit
- Windows 8.1 64-bit
- Windows 7 Ultimate 64-bit SP1
- Windows 2012 Server R2 64-bit SP1
- Windows 2008 Server R2 64-bit SP1

Getting Started

You can either start MapInfo EasyLoader from within MapInfo Pro or separately if you downloaded your MapInfo EasyLoader from the web. To launch MapInfo EasyLoader from within MapInfo Pro, see [Starting MapInfo EasyLoader from within MapInfo Pro](#).

To start MapInfo EasyLoader:

Double-click the MapInfoEasyLoader.exe file in the EasyLoader directory.

When you download MapInfo EasyLoader from the web, you download a zip file. When you extract the files from this zip, it creates an EasyLoader folder with all of the files necessary to run MapInfo EasyLoader including MapInfoEasyLoader.exe.

MapInfo EasyLoader is a single window within which you can set everything necessary to upload data tables to a database.

Name	Icon	Description
Database Connection		Select the connection to the database you want to upload tables to. You can use existing database connections created from the Microsoft Start menu or from in MapInfo Pro.
Manage Connection List and Choose a Connection		Create a new connection by selecting New Connection from the Database Connection list and then clicking this button.
Output Directory		Select a path location to save live access files to. Unless you select not to create them, MapInfo EasyLoader saves .tab files to your machine that contain database access information. MapInfo Pro uses these files when accessing the tables that MapInfo EasyLoader uploads. If you select not to create these files, you can create them later from within MapInfo Pro. A live access file has the same name as the upload file with a _srv extension, such as YourServerTableName_srv.tab.
Browse for an Output Directory (Folder)		Select a different directory from what is in the Output Directory list by clicking this button.
Source Table		Lists the data tables to upload to the database. To add tables, click the Browse for one or more tables to upload  button and then select Add Tables or Add Tables from Folder.

Name	Icon	Description
Server Table		Shows the table name as it will appear on the database server. You can click on the name to change it.
Operation		<p>Select for each table what you would like to do in the database:</p> <p>Create a New Table creates a new table in the database using the source table name.</p> <p>Replace Existing Table replaces an existing table in the database that has the same name as the source table.</p> <p>Append to Existing Table appends the contents of the source table to an existing table in the database that has the same name as the source table.</p> <p>For more details, see Table Processing Options.</p>
Catalog		<p>By default, upload tables are added to the map catalog on the database unless you clear the check box in the Catalog column.</p> <p>MapInfo Pro uses the Map Catalog on the database for spatial indexing. Select this check box to add a new Map Catalog or clear it to unregister a table from the Map Catalog. For information about the Map Catalog and what options you can set, see Using the Map Catalog. If you do not add the upload table data to the Map Catalog now, then you can do so later in MapInfo Pro.</p>
Upload States		Gives information about the upload. If there was an error with the upload, then hovering the mouse pointer over the image in this column shows more information.
Map Catalog		<p>This command is available when you are connected to a database connection. Selecting this command displays the Map Catalog window where you can edit data on the database. For more information about the Map Catalog, see Using the Map Catalog.</p> <p>Owner Name - You can filter the list of tables by selecting the Owner Name.</p> <p>Unregister - Unregisters a table from the database.</p> <p>Copy - Copies the selected row as comma separated values (csv).</p> <p>Close - Closes the Map Catalog window.</p>
Options		<p>The drop-down list has following command options:</p> <p>Options opens the Options dialog box where you can access the Server Table Processing options. For details, see Server Table Processing Options.</p>

Name	Icon	Description
		View Log opens the EasyLoader.log file in a text editor. This file contains the database connection information currently in use to upload data tables with. For details, see Locating the MapInfo EasyLoader Log File . Close When Complete is off by default. When set, MapInfo EasyLoader closes after finishing the upload.
Help		Online Help opens the product help. You need access to the internet to view this. PDF Help opens the product help as a PDF file for printing and for viewing when you do not have access to the internet. About MapInfo EasyLoader opens a message dialog box displaying the version of MapInfo EasyLoader.
Start or Resume Uploading Tables		Starts the upload process or continues the process if you put it on pause.
Pause the Upload		Pauses the upload process, which you can resume later.
Cancel the Rest of the Upload		Stops and cancels the upload process.

Starting MapInfo EasyLoader from within MapInfo Pro

MapInfo EasyLoader is a tool that ships with MapInfo Pro and can be launched as a tool within MapInfo Pro.

To start MapInfo EasyLoader from within MapInfo Pro:

1. In MapInfo Pro, on the HOME tab, in the Tools group, click Tools to open the **Tools Manager** window.
2. On the Running tab, double click MapInfo EasyLoader to open the **MapInfo EasyLoader** dialog box. If you do not see it listed on the Running tab, then locate it on the Registered tab.

To load MapInfo EasyLoader into the current session of MapInfo Pro, click the Load Tool (Run) icon beside it.

To reload MapInfo EasyLoader for all subsequent sessions, select the AutoLoad check box.

To load MapInfo EasyLoader for the current session and all subsequent sessions, select both.

3. Continue with the steps under [Upload Tables](#).

2 - Uploading Data

In this section

Understanding Your Data	9
Uploading Tables	12
About Upload Options	17
Using the Map Catalog	19

Understanding Your Data

Before uploading MapInfo tables, you need to know how your database manages spatial data from MapInfo Pro. There are database-specific restrictions for some geometry types, text objects, and coordinate systems. There are also situations in which geometry is considered invalid by a database. Know these restrictions before uploading MapInfo Pro tables to your database.

Text objects are unsupported by MapInfo EasyLoader. For example, MapInfo EasyLoader discards text objects when uploading a TAB file to Oracle.

Unsupported Geometry

A MapInfo Pro map may contain geometries that are unsupported by the database that the results are stored in. MapInfo EasyLoader skips unsupported object type and inserts the attribute data. No message displays.

- Oracle does not support arcs, ellipses, rectangles, and rounded rectangles.
- PostGIS does not support arcs, ellipses, rectangles, and rounded rectangles
- SQL Server does not support arcs, and lines that do not have distinct points.

SQL Server uploads invalid geometry objects and converts them to valid using a SQL server command, but SQL Server does not upload invalid geography objects. Attempting to upload invalid geography objects causes MapInfo EasyLoader to fail. This is a limitation of SQL Server and not of MapInfo EasyLoader. Due to this SQL Server limitation, upload only as geometry.

Loading Oracle Spatial Data

There are things to be aware of when uploading data tables to an Oracle database, such as:

- You cannot replace version-enabled tables in Oracle. This would make child versions of these tables obsolete.
- Oracle has a limit on identifier names, which cannot be more than 30 bytes. For details, see http://docs.oracle.com/database/121/SQLRF/sql_elements008.htm#SQLRF51129. The total number of bytes for table names, columns names, primary key constraint names (table name + "_PK") and index names (column name + "_IX") must not exceed 30 bytes. The number of bytes that each character uses is dependent upon the character set that Oracle is using. If you see the message "Identifier is too long", then make server table names or column names shorter.

Validating Oracle Data

There are two functions that allow you to validate data in Oracle:

1. SDO_GEOM.VALIDATE_GEOMETRY()
2. SDO_GEOM.VALIDATE_LAYER()

These functions may result in validation errors due to the tolerance level set by MapInfo EasyLoader. You may get error messages such as:

"ORA-13356 adjacent points in a geometry are redundant", or
"ORA-13022 polygon crosses itself"

To resolve these errors, reset the tolerance within the USER_SDO-GEOM_METADATA by adjusting them downwards (by a factor of 10) and rerun the validation.

Note: If you adjust the tolerance, you must re-create the spatial indexes because they use the tolerance when they are created.

To view the unsupported geometries in Oracle, see [Unsupported Geometry](#).

Loading for Oracle Locator

MapInfo EasyLoader loads data for the Oracle Locator, which is the same as loading data for Oracle Spatial.

Loading PostGIS Spatial Data

MapInfo EasyLoader uploads MapInfo tables into PostGIS, which stores spatial data in a geometry column called sp_geometry. Unsupported data types are ignored resulting in an empty record in the sp_geometry geometry column. For a list of unsupported data types, see [Unsupported Geometry](#).

MapInfo EasyLoader uploads a table with a Serial datatype for the primary key column. This allows the database to automatically increment the key column.

To check if a geometry object is valid in PostGIS, use the function called isvalid. The following is an example SQL statement:

```
select isvalid(sp_geometry) from test
```

Loading Microsoft SQL Server Spatial Data

Microsoft SQL Server Spatial includes Geography and Geometry data types. Geography fields hold geometries specified in the Lat/Long coordinate system. Geometry fields hold geometries specified in all other coordinate systems. If you choose to upload to a Geometry field type, then MapInfo

EasyLoader runs the SQL Server Spatial MakeValid() function to fix any geometry that SQL Server Spatial deems invalid. This may shift point locations for certain geometries.

In the **Options** dialog box, MapInfo EasyLoader enables you to control how the geometry data in a native table is uploaded to SQL Server Spatial.

Keep the default selection of Always Geometry to have MapInfo EasyLoader upload geometry objects to Geometry fields regardless of the coordinate system they use.

Select Auto Select to have MapInfo EasyLoader automatically decide whether the geometry object is uploaded to a Geography field (in cases where the geometry object uses the Lat/Long coordinate system) or a Geometry field (in cases where the geometry object does not use the Lat/Long coordinate system).

Note: When the Auto Select option is selected, if for any reason a geometry object that uses the Lat/Long coordinate system cannot be uploaded to a Geography field, it will instead be uploaded to a Geometry field.

To view the Unsupported Geometries in SQL, see [Unsupported Geometry](#).

MapInfo EasyLoader supports uploading MapInfo native tables into SQL Server Spatial. SQL Server Spatial allows spatial data to be stored into two data types; Geometry and Geography. SQL Server's rules that define what a valid geometry or geography are different than MapInfo Pro's rules. The SQL Server Spatial's Geography data type does not support:

- polygons that contain self-intersecting boundaries
- geography instances that span more than one hemisphere
- lines/polygon boundaries that have two successive, identical nodes

The SQL Server Spatial's Geometry type does not support:

- polygons that contain self-intersecting boundaries
- lines/polygon boundaries that have two successive, identical nodes

Refer to SQL Server books online for complete information on the rules that define these two types at:

<http://www.microsoft.com/en-us/sqlserver/default.aspx>

This means that you may have geometry data that are valid in a MapInfo TAB format that cannot be loaded into SQL Server Spatial without correction/edits. MapInfo EasyLoader aborts the upload if it comes across the geometry that SQL Server rejects. It will display the primary key of the record that contains the rejected object. A "MapInfo Upload Utility Error" message displays.

You may want to try the following steps to make the geometry acceptable to SQL Server.

- Use MapInfo Pro's Clean operation on the Object menu to correct the geometry.
- If your upload was to Geography type and it fails, try loading it to Geometry, correct the instance using the MakeValid() method on the Geometry type, and uploading the instance again. You will need a working knowledge of SQL and access to a SQL Server Spatial client to use this approach.

Consider the following example. Let us suppose the Geometry update failed for the row (state = 'Florida').

- Selectively update the offending geometry, using the `MakeValid()` method.

```
update states_geom set geom = geom.MakeValid( ) where state = 'Florida'
```

- Force an operation that does not alter the geometry, but forces a topology construction. Here, we perform a union of the Geometry with its own start point.

```
update states_geom set geom = geom.STUnion(geom.STStartPoint()) where state = 'Florida'
```

- Now this query will attempt to create a Geography instance from the reformed Geometry.

```
select state_name, Geography::STGeomFromWKB(geom.STAsBinary(), 4269)
from states_geom, where state = 'Florida'
```

Uploading Tables

For instructions to start MapInfo EasyLoader, see [Getting Started with MapInfo EasyLoader](#). To start MapInfo EasyLoader from within MapInfo Pro, see [Starting MapInfo EasyLoader from within MapInfo Pro](#).

To upload MapInfo TAB files using MapInfo EasyLoader:

1. From the Database Connection list select your existing connection created previously in MapInfo EasyLoader.

To create a new connection from MapInfo EasyLoader, select New Connection and click the Manage Connection List and Choose Connection  button. Follow the instructions under [Creating an Oracle Connection](#) or [Creating an ODBC Connection](#).

2. From the Output Directory list select a directory to save live access files to or select Do Not Create .tab Files.

Unless you select not to create them, MapInfo EasyLoader saves .tab files to your machine that contain database access information. MapInfo Pro uses these files when accessing the tables that MapInfo EasyLoader uploads. If you select not to create these files, you can create them later from within MapInfo Pro. A live access file has the same name as the upload file with a `_srv` extension, such as `YourServerTableName_srv.tab`.

To select a different directory from what is in the list, click the Browse for an Output Directory (Folder)  button.

3. Click the Browse for one or more tables to upload  button located beside the Source Table column heading in MapInfo EasyLoader. Add individual tables for upload by selecting Add Tables or a folder of tables by selecting Add Tables from Folder.

In the dialog that opens, navigate to the table or folder and select it. The tables that you select display in MapInfo EasyLoader and are the tables that you will upload.

4. The Server Table column shows the table name as it will appear on the database server. Click on the name if you want to change it.
5. In the Operation column, select how to upload each table.
 - Create a New Table creates a new table in the database using the name in the Server Table column.
 - Replace Existing Table replaces an existing table in the database that has the same name as in the Server Table column.
 - Append to Existing Table appends the contents of the source table to an existing table in the database that has the same name as in the Server Table column.

For details, see [Table Processing Options](#). For instructions on how to append multiple tables to a table in the database, see [Appending Multiple Tables to an Existing Table on the Server](#).

6. By default, upload tables are added to the map catalog on the database unless you clear the check box in the Catalog column.

MapInfo Pro uses the Map Catalog on the database for spatial indexing. Select this check box to add a new Map Catalog or clear it to unregister a table from the Map Catalog. For information about the Map Catalog and what options you can set, see [Using the Map Catalog](#). If you do not add the upload table data to the Map Catalog now, then you can do so later in MapInfo Pro.

7. Click Start or Resume Uploading Tables  to start the upload process.

To pause the upload, click this  button again.

To cancel the upload, click Cancel the rest of the upload .

MapInfo EasyLoader provides status information for the upload and displays the message "Upload Completed" when finished.

For information about how to set the spatial object types and the server processing options for an upload, see [About Upload Options](#).

Note: On some operating systems, such as Windows 7, when you select Add Tables in MapInfo EasyLoader the resulting **Open** dialog has a drop-down list on the Open button. From this list, you can select Show previous versions. This is a Microsoft system feature that lets you recover deleted files, view or restore a file you saved over, and lets you compare versions of a file side by side.

Creating an Oracle Connection

You must have Oracle Client software installed, which does not come with MapInfo EasyLoader. Before you create a connection in MapInfo EasyLoader to an Oracle database server, you will need to know the server name, and the user ID and password to use.

To connect to an Oracle database in MapInfo EasyLoader:

1. From the Database Connection list, select New Connection and then click the Manage Connection List and Choose Connection  button.
2. In the **Database Connection** dialog, select the Oracle radial button.
3. Fill in the Server Name, User ID, and Password fields.
4. Click OK to create the connection and close the dialog.

Note: Click the Manage Connection List and Choose Connection  button for an existing connection to edit its connection information.

Creating an ODBC Connection

These instructions assume that you have Open Database Connectivity (ODBC) set up beforehand. You can create new ODBC connections by following these instructions if you know the database name to connect to and the user ID to access the database with.

To connect to a database using ODBC:

1. From the Database Connection list, select New Connection and then click the Manage Connection List and Choose Connection  button.
2. In the **Database Connection** dialog, select the ODBC radial button and then click the button beside it.
3. In the **Select Data Source** dialog, select an existing ODBC connection by doing one of the following:
 - On the File Data Source tab, use the Look in drop-down list or the Up button to locate the database you want to upload to and select it.
 - On the Machine Data Source tab, double-click the data source driver you want to connect to and navigate to the database you want to upload to and select it.

If you do not have an existing ODBC connection, click New to open the **Create New Data Source** wizard and create one. Select the connection after creating it.

4. Click OK to use your selection.

The **Database Connection** dialog populates with the ODBC connection information. Click OK to save your selection.

Table Processing Options

This section describes the table options available in MapInfo EasyLoader from the Operation list. All operations use the name that you specify in the Server Table column. By default the name in this column is the same as the source table.

Create New Table A server table is created with the name that you specify in the Server Table column. If this option is chosen and a table with the same name already exists on the server, an error message displays. Use a different name or choose the option Replace Existing Table to upload the table.

Replace Existing Table When this option is selected, if a server table of the same name already exists, it is dropped and a new table is created to match the MapInfo table being uploaded. This option uses the table name from the Server Table column. You can change this to the name of an existing table on the server.

Append to Existing Table The MapInfo table is appended to the server table named in the Server Table column when the server table exists and the structure of the two tables match. Otherwise, you get an error and the table is not uploaded. For Oracle Spatial, the tables must have the same table structure and be in the same projection. When there are multiple tables selected, the **Table Append** dialog box opens. For instructions, see [Appending Multiple Tables to an Existing Table on the Server](#).

When there are multiple tables selected in MapInfo EasyLoader, then selecting or clearing the Catalog check box applies to all of the tables in the selection.

Appending Multiple Tables to an Existing Table on the Server

The **Table Append** dialog box opens when you select more than one table in MapInfo EasyLoader and then either enter a new name in the Server Table column for one of those tables or select Append from the list in the Operation column.

In the **Table Append** dialog box, select an operation from the Operation for Collected Table list for uploading the tables:

- Create New generates a new server table to append all selected tables to.
- Replace drops the existing server table and then creates a new table to append all of the selected tables to.
- Append appends all of the selected tables to an existing server table.

If the dialog includes a Server Table Name field, then enter the name of the server table that you want to create, replace, or append to. If it does not include this field, then MapInfo EasyLoader uses the name that you entered in the Server Table column.

All tables must have the same table structure and be in the same projection. Tables will not append if their table structure differs.

Uploading Tables with Time and DateTime Columns

Time and DateTime data types sometimes require conversion because of variations from server to server, and between the server and MapInfo data types. The following table shows how the data types are converted from MapInfo Pro to each server.

From MapInfo Pro	To Oracle	To MS Access	To MS SQL Server
DATE	DATE	DATETIME*	DATETIME*
TIME	TIMESTAMP(3)*	DATETIME*	DATETIME*
DATETIME	TIMESTAMP(3)	DATETIME	DATETIME

* The MapInfo data type will be extended on the servers. When the same data comes back to MapInfo Pro, the data type will be as is indicated on the server. This conversion is consistent with MapInfo Pro behavior when a MapInfo table is saved to a server using MapInfo Pro.

On servers that do not support DATE or TIME data types, the data is converted to a DATETIME type. In this conversion, part of the data will be missing because the MapInfo types contain either the date or the time, but not both. The server default values for the date or the time are used to fill in the missing data. Conversions to DATETIME are made for SQL Server versions earlier than version 10 (prior to SQL Server Spatial).

For example, if the server does not support the MapInfo DATE type, the upload process converts the DATE type to a DATETIME type. The date value comes from the MapInfo table, but the time value is filled in with the server default value for time. The following table shows how the MapInfo data types are converted when they are not supported on the server and what default value are used to fill in the missing data:

MapInfo Data Type	To Server Data Type	Server Default Values	Databases
MapInfo DATE type	DATETIME/TIMESTAMP type	midnight: 12:00.00.000 AM	All databases
MapInfo TIME type	DATETIME/STAMP type	current date	MS Access, MS SQL Server*
MapInfo TIME type	DATETIME/STAMP type	first day of current month and year	Oracle

* SQL Server versions earlier than version 10 (prior to SQL Server Spatial).

About Upload Options

This section explains the settings in the **Options** dialog box, which describe how to process tables. To open this dialog in MapInfo EasyLoader, click Options and then select Options from the list.

Server Table Processing Options

Exclusive User	You can speed up load time on large tables when you are the only one attempting to update or upload the table. If you do not select this option, MapInfo EasyLoader verifies after each commit that no other updates are made to the table while it is being uploaded. Selecting this option prevents this test from occurring, which can improve runtime performance for large tables.
Create Primary Key	This creates a primary key for the Create New Table and Replace Existing Table operations. This primary key is created in the MI_PRINX column for Oracle Spatial, SQL Server Spatial, and PostGIS, or MI_SQL_REC_NUM for XY and MICODE. These columns contain sequential numbers generated by MapInfo EasyLoader. These columns are always created, but do not have to be a primary key. The Append to Existing Table operation does not create a primary key.
Grant Public Full Access	This grants PUBLIC read and write access to the server table.
Create Spatial Index	<p>This creates a spatial index called <tablename>ind on the geometry column of tables. When building your own spatial index, for your specific requirements, clear this check box to save loading time.</p> <p>For Oracle Spatial tables, this creates a spatial index called <tablename>_SX on the geometry column. The index tiling level is based on the SDO_TUNE.ESTIMATE_TILING_LEVEL function. For tables with fewer than 7500 rows, the tiling level is restricted to eight (8). After the index is built, the ANALYZE table function is run on the index table. The spatial index is R-Tree for Oracle 8.17 or later.</p>
Create IDENTITY Column (SQL Server only)	Creates the primary key column with IDENTITY properties. When selected, SQL Server automatically generates the primary key column values for new rows, so that you do not need to add them manually.
Create Indexes	<p>When uploading a new table or replacing an existing table, this indexes all of the columns that are indexed in the original table. By default, this option is not selected.</p> <p>The table upload begins when you select this check box and click OK.</p>

Note: If the column name length is too long, the index creation fails. The name length limit varies per database.

Style Column	Specifies whether per-row styles are loaded with the data. You can set the name of the column in the text box beside the Style Column check box. The default column name is MI_STYLE. Note: To load per-row styles, the MapInfo Map catalog for the database must contain the following columns: RENDITIONTYPE, RENDITIONCOLUMN, RENDITIONTABLE, and NUMBER_ROWS. For more information, see MAPINFO_MAPCATALOG Table Structure on page 20.
Commit Interval	Specifies a commit interval for uploading. MapInfo EasyLoader commits the inserted records when the commit interval is reached. The default commit interval is 1000. If the commit interval is set to 0 (zero), the whole range of records is inserted as a single transaction, before a commit is issued.
Table Errors Threshold %	A value of zero (0) to 100 that is the percentage of table upload failures allowed before the upload stops. A value of 100 allows the upload to complete no matter the number of failures.
Insert Errors Threshold %	A value of zero (0) to 100 that is the percentage of insert failures allowed on a table before the table upload stops. The percentage is the number of rows that fail based on the number of rows in the table. A value of 100 allows the process to complete no matter the number of failures. The Error Threshold percentages work together. An insert failure for a row in a table counts towards the Table Errors Threshold.
Schema Name	Specifies a schema name when uploading tables to SQL Server, or PostgreSQL and/or PostGIS.
Spatial Object Type	Specifies an object type, which depends on the server connection in use. For details, see Spatial Object Type Options on page 18.

Spatial Object Type Options

Choose from Oracle Spatial (depending on the type of connection), MICODE (XY with MapInfo Key), XY, PostGIS Spatial, or SQL Server Spatial. The default for loading spatial data is SQL Server Spatial (Auto Select) if this option is available. Otherwise, MapInfo EasyLoader defaults to the MICODE option.

Oracle Spatial	Use this option if the server is Oracle Spatial.
MICODE (XY with MapInfo Key)	This option stores the data as XY coordinates on the server and creates the server table as a point table. If the MapInfo table to be uploaded is not a point table and this option is chosen, the centroid is abstracted and stored on the server table, if you instruct it to do so. The difference between XY and

MICODE is that the MICODE provides a MapInfo key as the spatial index, making its performance superior to XY.

XY This option is the same as the MICODE option, except that this option does not create a spatial index.

PostGIS Spatial Use this option with a spatialized PostgreSQL database.

SQL Server Spatial Use these options if the server is Microsoft SQL Server Spatial:

Auto Select – MapInfo EasyLoader automatically decides whether to upload the geometry object to a Geography field (if the geometry object is using a Lat/Long coordinate system) or to a Geometry field (if the geometry object is using another coordinate system).

Always Geometry – MapInfo EasyLoader always uploads the geometry object to a Geometry field, regardless of the coordinate system specified for the geometry object.

Note: When the Auto Select option is selected, if for any reason a geometry object that uses the Lat/Long coordinate system cannot be uploaded to a Geography field, it will instead be uploaded to a Geometry field.

Using the Map Catalog

The MAPINFO_MAPCATALOG is a registry table for databases that stores metadata about geometry tables in the database. Using both the table name and owner name as the key, the MAPINFO_MAPCATALOG identifies the geometry column, geometry type, projection, projection bounds, and table and feature level rendition information. The MAPINFO_MAPCATALOG is used by a number of MapInfo products, including MapInfo Pro, that access map data from databases.

If a Map Catalog does not exist, it can be created during the upload process by selecting the check box in the Catalog column for the table being uploaded. The user account for the server connection must have permission to the MapInfo schema on the server to create the map catalog. If it does not have adequate permissions, then creation will not succeed and the table will not upload. If this happens, ask the database administrator to grant you permission to the MapInfo schema.

After the table is uploaded, an entry is made in the MAPINFO.MAPINFO_MAPCATALOG to represent that table. A separate entry is made for every table you upload.

If the table is made up of a single type of object, then the server object type is restricted to that type, otherwise the type is ALL. Also, the symbol clause generated is based on the server type. For example: After uploading the table states.tab the server type will be x.2 (polygons), where x is a number that represents either IDS, SQL Server, or Oracle Spatial, and the symbol clause will have only the information for a polygon.

Note: See also, [Loading Microsoft SQL Server Spatial Data](#).

Unregister Tables from the Map Catalog

The Map Catalog stores the metadata used by MapInfo Pro and other MapInfo products to open tables with geometry data. If your database has a Map Catalog, you can delete entries in the catalog that are no longer valid.

Note: If the Map Catalog is present and there are no entries in it, the Catalog check box beside the table name in MapInfo EasyLoader does not enable.

To unregister entries from an existing Map Catalog:

1. Select the data owner from the Owner Name drop-down list. The Tables Registered in Map Catalog list populates with columns of information for each registered table.
2. Select the table or tables you want to unregister from the Map Catalog and then click the Unregister button.
3. Click Close when you are done.

The Copy button copies the selected rows in a comma separated format to the clipboard, which you can then paste into other applications.

MAPINFO_MAPCATALOG Table Structure

MAPINFO_MAPCATALOG has the following table structure:

```

SPATIALTYPE FLOAT
TABLENAME CHAR(32)
OWNERNAME CHAR(32)
SPATIALCOLUMN CHAR(32)
DB_X_LL FLOAT
DB_Y_LL FLOAT
DB_X_UR FLOAT
DB_Y_UR FLOAT
VIEW_X_LL FLOAT
VIEW_Y_LL FLOAT
VIEW_X_UR FLOAT
VIEW_Y_UR FLOAT
COORDINATESYSTEM CHAR(254)
SYMBOL CHAR(254)
XCOLUMNNAME CHAR(32)
YCOLUMNNAME CHAR(32)
RENDITIONTYPE INTEGER
RENDITIONCOLUMN CHAR(32)
RENDITIONTABLE CHAR(32)
NUMBER_ROWS INTEGER

```

The following script may be used to modify an existing MAPINFO.MAPINFO_MAPCATALOG to add the rendition columns if they do not exist. Run this script as user MAPINFO.

```
ALTER TABLE MAPINFO.MAPINFO_MAPCATALOG ADD RENDITIONTYPE INTEGER;  
ALTER TABLE MAPINFO.MAPINFO_MAPCATALOG ADD RENDITIONCOLUMN CHAR(32);  
ALTER TABLE MAPINFO.MAPINFO_MAPCATALOG ADD RENDITIONTABLE CHAR(32);
```

3 - Productivity Aids

In this section

Shortcuts by Keystroke	23
Using Command-Line Flags to Run MapInfo EasyLoader	24
Command-Line Flag Descriptions	26
Locating the MapInfo EasyLoader Log File	37
Locating the MapInfo EasyLoader Configuration File	37

Shortcuts by Keystroke

This section lists the short-cuts available in MapInfo EasyLoader.

Command	Icon	Shortcut
Manage Connection List and Choose a Connection		Ctrl+D
Browse for an Output Directory (Folder)		Ctrl+N
Source Table		Ctrl+T
Add Tables from Folder		Ctrl+F
Map Catalog		Ctrl+M
Options		Ctrl+O
View Log		Ctrl+L
Online Help		F1
PDF Help		Alt+F1
About MapInfo EasyLoader		Shift+F1
Start or Resume Uploading Tables		Ctrl+U
Pause the Upload		Ctrl+P
Cancel the Rest of the Upload		Ctrl+S

The following are shortcuts for selecting and deleting tables in MapInfo EasyLoader's table list.

Command	Shortcut
Scroll the list of tables	Page up, Page down
Top of list	Home
Bottom of list	Bottom
Move between columns	Right Arrow, Left Arrow
Select a single table	Click
Select tables in the list	Ctrl+Click
Select a range of tables	Shift+Click
Select all tables	Ctrl+A
Unselect all tables	Ctrl+Shift+A
Delete selected tables	Delete

When there are multiple tables selected in MapInfo EasyLoader, then selecting or clearing the Catalog check box and choosing from the Operation list applies to all of the tables in the selection.

Using Command-Line Flags to Run MapInfo EasyLoader

You can run MapInfo EasyLoader from the Windows command-line to apply specific upload parameters or invoke MapInfo EasyLoader with upload parameters from a custom script or application.

To open MapInfo EasyLoader from the **Command-Line** window, on the Windows Start menu type cmd. Change directories (cd) into the folder that has the MapInfoEasyLoader.exe file in it, and then type MapInfoEasyLoader.exe. MapInfo EasyLoader opens on your screen and is ready to use.

To upload a table when you open MapInfo EasyLoader from the **Command-Line** window, modify the following example. It uploads a table (/T) called states.tab and gives it the server table name mystates, grants all rights (/G) to make mystates public, and specifies that per-row styles are being loaded with the data (/Y).

```
MapInfoEasyLoader.exe /T C:\data\states usa.tab;mystates /G /Y
```

If a path contains spaces, use quotation marks around it, for example:

```
MapInfoEasyLoader.exe /T "C:\data\states usa.tab;mystates" /G /Y
```

```
MapInfoEasyLoader.exe /g /P R /s "DRIVER={PostGreSQL  
UNICODE (x64) };DATABASE=QADB;Server=MiProPostGIS;UID=pro;PWD=pro;Port=5432"
```

For a list of available command-line flags, see [Command-Line Flags Descriptions](#).

For instructions on how to locate the MapInfo EasyLoader state file, which stores command-line flag settings for the session, see [Locating the MapInfo EasyLoader Command-Line Flags RSP File](#).

Mixing Command-Line Flags and MapInfo EasyLoader Window Options

Command-line flags may be mixed with the MapInfo EasyLoader user interface by using a Windows shortcut. This makes it easier to set flags as your default while being able to override them from the interface. These flags are only available from the command-line.

To create a shortcut for MapInfo EasyLoader that includes preset flags:

1. Right-click on the MapInfoEasyLoader.exe file and then select Send to and Desktop (create shortcut). The resulting file saves to your desktop.

To place in a different location, right-click and select Create shortcut, and then move the resulting shortcut file.

2. Right-click the shortcut file and select Properties to open the **Shortcut Properties** dialog.
3. On the Shortcut tab, within the Target edit box, add command-line flags to the end of the path. Separate the flags with spaces.

To have the flags take effect, launch MapInfo EasyLoader using this shortcut. You can override the settings by changing them in the MapInfo EasyLoader window.

Locating the MapInfo EasyLoader Command-Line Flags RSP File

Closing MapInfo EasyLoader saves a state file with the command-line flags used on startup and set during the session. The file is called MapInfoEasyLoader.Settings.rsp and is saved under the AppData folder (for example, C:\Users\\AppData\Roaming\MapInfoEasyLoader.Settings.rsp).

The AppData folder is in a path like C:\Users\\AppData. If you cannot see the AppData folder, it is because it is hidden. To locate it, on the Microsoft Start menu in the Search programs and files field type %AppData%.

This file contains all of the flags used to start MapInfo EasyLoader regardless of whether you used the command-line. It includes the default flags and their settings, such as /CommitInterval,

`/TableUploadErrorThreshold`, `/InsertRowErrorThreshold`, and `/StyleColumnName`. This flag also records the settings made during the session that affect startup flags. You can use the information in this file to restart EasyLoader via the command-line using the same settings as the previous session.

To customize or automate your MapInfo EasyLoader experience, you can create your own state file by starting MapInfo EasyLoader from the command-line with the `/SaveCommandArgs` flag. You can then use this state file to restart EasyLoader via the command-line with the `/CommandFile` flag to apply the same settings as the previous session. MapInfo EasyLoader does not create the `MapInfoEasyLoader.Settings.rsp` file when you use the `/SaveCommandArgs` flag.

Command-Line Flag Descriptions

The following flags specify upload parameters when launching MapInfo EasyLoader from a command-line. Many of the arguments have a long form and a legacy short form, such as the command to set a schema name when you upload a table: `/SchemaName` and `/B`.

Use a space or an equal sign (=) to separate a command argument and its value. This example sets a target schema name, `/SchemaName=schema_name` or `/B schema_name`.

Do Not Treat Empty (Zero-Length) Strings as Null: /NoTreatEmptyStringAsNull

This flag is used to override the default upload behavior so that empty strings are not treated as NULL values, but are maintained as zero-length string values. This flag is added for backward compatibility with 32-bit EasyLoader behavior. The default behavior is to treat empty strings as NULL values.

Note: Since Oracle treats zero-length strings as indistinguishable from NULL, this flag is not applicable for Oracle.

Syntax: `/NoTreatEmptyStringAsNull`

Help: /?, /Help

This flag shows the list of available command-line options.

Syntax: `/?`

Syntax: `/Help`

Append All Tables to One: /A, /AppendAll

This flag uploads multiple tables to a single table, when the table structures are the same.

Syntax: `/A`

Syntax: `/AppendAll [=server_table_name]`

server_table_name is optional for specifying a server table to append to.

Set the Application Locale: /AppLocale

This flag sets the locale at runtime. It requires an ISO language code in the four character form as outlined by Microsoft's Language Culture Names list

[https://msdn.microsoft.com/en-us/library/ee825488\(v=cs.20\).aspx](https://msdn.microsoft.com/en-us/library/ee825488(v=cs.20).aspx).

Supported languages are:

Cs-CZ	Czech – Czech Republic
-------	------------------------

da-DA	Danish – Denmark
-------	------------------

de-DE	German – Germany
-------	------------------

en-US	English – United States
-------	-------------------------

es-ES	Spanish – Spain
-------	-----------------

fi-FI	Finnish – Finland
-------	-------------------

fr-FR	French – France
-------	-----------------

he-IL	Hebrew – Israel
-------	-----------------

it-IT	Italian – Italy
-------	-----------------

ja-JP	Japanese – Japan
-------	------------------

nl-NL	Dutch – Netherlands
-------	---------------------

pl-PL	Polish – Poland
-------	-----------------

pt-PT	Portuguese – Portugal
-------	-----------------------

ru-RU	Russian – Russian Federation
-------	------------------------------

sv-SE	Swedish – Sweden
-------	------------------

tr-TR	Turkish – Turkey
-------	------------------

zh-CN

Chinese (Simplified) – China

Syntax: /AppLocale=languagecode

```
/AppLocale=fr-FR
```

Schema Name: /B, /SchemaName

This flag specifies a schema name when uploading tables to SQL Server. If you do not supply a schema name, then this flag uses the user's default schema on the server. On SQL Server, the default is usually set to dbo.

Syntax: /B string_value

Syntax: /SchemaName=string_value

```
/SchemaName=MIPROUSER
```

Create Indices for all Locally-Indexed Columns: /C, /CreateIndexes

When uploading a new table or replacing an existing table, this flag tells MapInfo EasyLoader to automatically index columns on the server table when they are indexed on the local table.

This flag does not make or reset column selections. Instead, use the MapInfo EasyLoader user interface. For details, see [Create Indexes](#) under **Server Table Processing Options** and **Mixing Command-Line flags and MapInfo EasyLoader Window Options**.

Syntax: /C

Syntax: /CreateIndexes

Input Command File: /CF, /CommandFile

This flag loads command-line arguments from an input file. Each line in the file must contain only one command. MapInfo EasyLoader skips blank lines and comments that begin with a hash tag (#). You must specify the path to the input file with this flag.

You can use the state file that MapInfo EasyLoader automatically creates under the AppData folder with the name MapInfoEasyLoader.Settings.rsp as input. For details, see [Locating the MapInfo EasyLoader Command-Line Flags RSP File](#). You can also generate a state file using the [/SaveCommandArgs](#) flag.

Syntax: /CF CommandFilePath

Syntax: /CommandFile=CommandFilePath

```
/CommandFile=C:\MyFiles\EasyLoader\Commands\myCommandFile.txt
```

Convert WKB on Oracle: /ConvertWKBOnOracle

This flag passes geometries as WKB (Well-Known Binaries) to an Oracle server for conversion to the SDO_GEOMETRY type. Only use this flag with OCI connections. By default, MapInfo EasyLoader does this conversion before passing geometries to the Oracle server to improve performance. You would only use this flag to work around WKB to SDO_GEOMETRY conversion problems.

Syntax: /ConvertWKBOnOracle

File Directory for Server Table(s): /D, /OutputFolder

This flag generates a TAB file in a specific directory location to access a remote database with. The new .tab file name appends _srv.tab to your server table name, yourServerTableName_srv.tab. The directory must be valid to upload a table to it. An empty directory is considered valid.

Syntax: /D string_value

Syntax: /OutputFolder=string_value

If the path contains spaces, use quotation marks around it.

```
/OutputFolder="C:\my work folder\Tables\My Upload Tables"
```

Exclusive Use of Table: /E, /ExclusiveUse

This flag improves load time on large tables when you are the only person updating the table. This flag does not guarantee that MapInfo EasyLoader can obtain exclusive use of a table. After each commit, MapInfo EasyLoader checks the maximum value of the primary key column (called MI_PRINX) to detect if entries were been made by other processes. This flag prevents this check, which improve the upload time for large tables.

Syntax: /E

Syntax: /ExclusiveUse

You can use this flag in a MapInfo EasyLoader shortcut, see [Mixing Command-Line flags and MapInfo EasyLoader Window Options](#).

Log File name: /F, /LogFileName

This flag specifies the name and location of the log file created when MapInfo EasyLoader uploads a table. By default, a log file named EasyLoader.log is created in the TEMP directory. If you specify a file name but do not provide a path, then MapInfo EasyLoader saves the log file to the same directory as the EasyLoader.exe file.

Syntax: /F string_value

Syntax: /LogFileName=string_value

This example shows only the name of the log file, which saves to the same directory as MapInfoEasyLoader.exe.

```
/F myLogFile.txt
```

```
/LogFileName=myLogFile.txt
```

This example specifies the full path for the log file. If the path contains spaces, use quotation marks around it.

```
/F "C:\my work folder\myLogFile.txt"
```

```
/LogFileName="C:\my work folder\myLogFile.txt"
```

Table Folder: /Folder

This flag loads all tables from a folder. Use a semicolon (;) to separate each element.

Syntax: /Folder=FolderName;ServerOperation;AddToMapCatalog;IncludeSubFolders

Where:

- FolderName is the name of the root folder.
- ServerOperation is the default server operation for the folder, use Append, CreateNew, or Replace. The default is Append.
- AddToMapCatalog creates an entry for the folder tables in the Map Catalog. This is set to true by default, so set to false to not add an entry in the Map Catalog.
- IncludeSubFolders includes subfolders when querying for the folder's tables. This is set to true by default, so set to false to not include subfolders in the query.

The following example does not include the optional AddToMapCatalog and IncludeSubFolders elements, both of which are set to true by default.

```
/Folder=C:/MyData;CreateNew
```

The following example does not include tables in subfolders under MyData. The third AddToMapCatalog element is blank, because it is set to true by default.

```
/Folder=C:/MyData;CreateNew;;false
```

Grant All: /G, /GrantAll

This flag grants all rights as Public. By default, this is not set and tables are private.

Syntax: /G

Syntax: /GrantAll

Do Not Create a Spatial Index: /I, /NoCreateSpatialIndex

This flag prevents MapInfo EasyLoader from creating a spatial index on the uploaded table. By default, this flag is turned off and MapInfo EasyLoader creates a spatial index.

For Informix Dynamic Server (IDS) and User Defined Object (UDO) tables, MapInfo EasyLoader creates a spatial index and then issues the "update statistics medium" statement.

Syntax: /I

Syntax: /NoCreateSpatialIndex

For more information, see [/NoCreatePrimaryKey](#), which controls the unique index.

Create Automated Key Column for SQL Server: /K, /AutomatedKey

This flag automatically generates unique key column values in SQL Server, so that you do not need to fill the key manually when you insert a new row. The key column is called MI_PRINX or MI_SQL_REC_NUM depending on the spatial object type that is selected: MI_PRINX is for spatial types and MI_SQL_REC_NUM is for XY types.

Syntax: /K [IDENTITY | NO_IDENTITY]

Syntax: /AutomatedKey [IDENTITY | NO_IDENTITY]

The key column is created with the IDENTITY property by default, so you do not need to include the IDENTITY property. If you want to turn this off, use the NO_IDENTITY property.

```
/AutomatedKey
```

```
/AutomatedKey=NO_IDENTITY
```

You can use this flag in an MapInfo EasyLoader shortcut, see [Mixing Command-Line flags and MapInfo EasyLoader Window Options](#).

List of MapInfo Tables: /L, /TableList

This flag specifies a text file that lists the tables to upload. The format of each line in the text file is the same as for [Table Specification: /T\[table\]](#).

Syntax: /L [TextFile | Path]

Syntax: /TableList=[TextFile | Path]

Where:

- TextFile is the name of the text file that lists the tables to upload.
- Path is the full path location to the text file.

This example shows only the name of the file, which is located the same directory as MapInfoEasyLoader.exe.

```
/L ListOfTables.txt
```

This example shows the path to the file. If the path contains spaces, use quotation marks around it.

```
/TableList="C:\my work folder\Tables\ListOfTables.txt"
```

Spatial Object Type: /M, /SpatialObjectType

This flag specifies the object type to use: either MICODE for XY with MapInfo key or XY for XY only. MapInfo EasyLoader does not display for any value other than MICODE and XY.

Syntax: /M [XY | MICODE]

Syntax: /SpatialObjectType=[XY | MICODE]

Examples:

```
/M XY
```

```
/SpatialObjectType=MICODE
```

Start Application Minimized: /Min, /Minimized

This flag starts MapInfo Easyloader in a minimized state. This does not alter the behavior of the **/NoStart** flag. By default, this is set to false.

Syntax: /Min

Syntax: /Minimized

Do Not Auto Start the Table Load: /NoStart

This flag disables the auto start behavior, so that MapInfo EasyLoader does not automatically connect and begin to upload tables, even if the connection and tables are specified. By default, this is set to false.

Syntax: /NoStart

Run Without User Interface: /NoUi

This flag runs MapInfo EasyLoader without a user interface. This is useful when running MapInfo EasyLoader from a batch file for auto loading tables.

This flag requires a connection and a list of tables to upload. At minimum, you would include a server connection string flag **/ConnectionString** or the Oracle connection string flag **/OracleConnectionString**, and a list of MapInfo tables **/TableList** or a folder of tables **/Folder**. You do not need to include the **/QuitWhenDone** flag with **/NoUi**.

Syntax: /NoUi connection tables

Where:

- connection is either **/ConnectionString** or **/OracleConnectionString**.

- tables is either [/TableList](#) or [/Folder](#).

Connection String: /O, /OracleConnectionString

This flag sets a connection string for Oracle Spatial.

Syntax: /O [friendly_name]user_name/password@server_name

Where:

- friendly_name is optional for setting a specific name for the connection identifier.
- user_name is the Oracle user name to connect with.
- password is the password for the Oracle user name.
- server_name is the name of the server you are connecting to.

This example sets a connection string without defining a name for the connection identifier.

```
/OracleConnectionString=myUid/myPwd@oraserver11gr2
```

This example sets the same connection and names the connection Oracle1. If the connection name has spaces, use quotation marks around it.

```
/OracleConnectionString=[Oracle1]myUid/myPwd@oraserver11gr2
```

For ODBC connections, see [Server Connection String: /S, /ConnectionString](#).

Default Server Operation: /P, /ServerOperation

Use this flag to specify what to do with the upload tables in the server.

- A appends to an existing server table.
- C creates a new server table. If the table you are uploading has the same name as a table on the server, the upload fails.
- R replaces an existing server table.

Syntax: /P [A | C | R]

Syntax: /ServerOperation=[A | C | R]

```
/P A
```

```
/SpatialObjectType=C
```

Exit after Upload: /Q

This flag closes MapInfo EasyLoader after completing the upload.

Syntax: /Q

Syntax: /QuitWhenDone

Server Connection String: /S, /ConnectionString

This flag passes an ODBC connection string to the program. The **ODBC Connection** dialog box does not display when you supply enough information to connect.

Syntax: /ConnectionString=[(friendly_name)connection_string | user_name/connection_string]

Where:

- friendly_name is optional for setting a specific name for the connection identifier.
- user_name is the Oracle user name to connect with.
- connection_string is the ODBC connection string.

This example uses a data source (DSN).

```
/ConnectionString=DSN=MyDataSource
```

This example sets a connection name to MyDBConnection. It then uses the full connection string to the database, where items in all capitals are connection string parameters.

```
/ConnectionString=[MyDBConnection]UID=MyId;DATABASE=MyDB;  
HOST=MyServer;SERVER=MyServer_tli;SERVICE=sqlexec;PROTOCOL=onsoctcp
```

This example sets a connection name to My Data Source. It uses a data source (FileDSN=C:\Tenop\MyDataSource.dsn) with a user ID (UID=MyUserID) and password (PWD=MyPassword). If the connection name has spaces, use quotation marks around the entire connection string.

```
/ConnectionString="[My Data  
Source]FILEDSN=C:\Tenop\MyDataSource.dsn;UID=MyUserID;PWD=MyPassword"
```

For ODBC connections, see [Connection String: /O, /OracleConnectionString](#).

Save Command Response File: /Save, /SaveCommandArgs

This flag creates a state file when MapInfo EasyLoader shuts down that contains the command-line arguments used to start MapInfo EasyLoader and that were set during the session. You can use this state file to restart EasyLoader on the command-line with the [/CommandFile](#) flag to apply the same settings as the previous session.

MapInfo EasyLoader automatically creates a state file that you can also use with the [/CommandFile](#) flag. It is located under the AppData folder with the name MapInfoEasyLoader.Settings.rsp (for example, C:\Users\<UserID>\AppData\Roaming\MapInfoEasyLoader.Settings.rsp). MapInfo EasyLoader does not create the MapInfoEasyLoader.Settings.rsp file when you use the [/SaveCommandArgs](#) flag.

Note: The AppData folder is in a path like C:\Users\<UserID>\AppData. If you cannot see the AppData folder, it is because it is hidden. To locate it, on the Microsoft Start menu in the Search programs and files field type %AppData%.

Syntax: /SaveCommandArgs=StateFilePath

Where, StateFilePath is a TXT file and any valid path location.

Example:

```
/SaveCommandArgs=C:\MyFiles\EasyLoader\Commands\myCommandFile.txt
```

Table Specification: /T /Table

This flag passes a single table name to MapInfo EasyLoader.

Syntax: /Table=TableName(;ServerTableName;UploadRange;ServerOperation;AddToMapCatalog)

Where:

- TableName is the .tab file name to upload.
- ServerTableName (optional) is the server table name. If left empty, MapInfo EasyLoader uses TableName.
- UploadRange (optional) is the range of table rows to upload. The range is a starting number and ending number with a comma (,) separator. A range of three to six would be 3,6.
- ServerOperation (optional) is the server operation for the table. Use Append, CreateNew, or Replace.
- AddToMapCatalog creates an entry for the table in the Map Catalog when set to true. By default, this is set to true, so you do not need to specify it in the command-line.

This example includes all parameters.

```
/T C:\data\states.tab;mystates;1,500;CreateNew>true
```

This example leaves the upload range value empty, so that it uploads all of the rows in the table. It excludes the Map Catalog entry, so an entry is not created for this table in the Map Catalog.

```
/Table=C:\data\states.tab;MyOtherStates;;Replace
```

This example includes only the path to the table to upload.

```
/Table=C:\data\atable.tab
```

If the path of the upload file contains spaces, use quotation marks around it.

```
/Table="C:\My Documents\tables\atable.tab";MyTable;;Replace>true
```

Do Not Create a Primary Index: /U, /NoCreatePrimaryKey

This flag prevents MapInfo EasyLoader from creating a primary key when uploading a table. It is turned off by default. When uploading to Oracle Spatial, MapInfo EasyLoader creates a primary key on the MI_PRINX column and names the key table_name_PK.

Syntax: /U

Syntax: /NoCreatePrimaryKey

Also see [Do Not Create a Spatial Index: /I, /NoCreateSpatialIndex](#), which controls the spatial index.

Commit Interval: /X, /CommitInterval

This flag sets a commit interval. MapInfo EasyLoader inserts records to the database and then commits them when it reaches the commit interval. The same interval value applies when creating the spatial index for Oracle Spatial. If the commit interval is set to zero (0), MapInfo EasyLoader inserts the entire range of records as a single transaction before issuing a commit.

Syntax: /X integer_value

Syntax: /CommitInterval=integer_value

Where, integer_value is an integer value. The default value is 1000.

This example sets the commit interval to 500, so that MapInfo EasyLoader commits all of the records to the database every 500th record.

```
/CommitInterval=500
```

Style Column Name: /Y, /StyleColumnName

This flag loads per-row styles with the upload data. If you do not provide a name for the column to use, MapInfo EasyLoader uses the default MI_STYLE column name. If you do not want to create a style column on the server table, include the NO_STYLE parameter.

Syntax: /Y (string_value) (NO_STYLE)

Syntax: /keyword=(string_value) (NO_STYLE)

This example uses the column called TableStyleCol to load per-row styles.

```
/Y TableStyleCol
```

```
/StyleColumnName=TableStyleCol
```

This example uses the default column called MI_STYLE to load per-row styles.

```
/Y
```

```
/StyleColumnName
```

This example does not create a style column on the server table.

```
/Y NO_STYLE
```

```
/StyleColumnName=NO_STYLE
```

SQL Server Auto Select Geometry: /Z, /AutoSelectSqlServerSpatial

Use this flag to upload the table as a Geometry data type in Microsoft SQL Server Spatial regardless of the coordinate system in the native table.

Syntax: /Z

Syntax: /AutoSelectSqlServerSpatial

For more information, see [Loading Microsoft SQL Server Spatial Data](#).

Locating the MapInfo EasyLoader Log File

MapInfo EasyLoader saves a log file called EasyLoader.log under the AppData folder (for example, C:\Users\\AppData\Local\EasyLoader.log).

The log file contains the database connection information currently in use to upload data tables with. You can see the contents of this file from within MapInfo EasyLoader by clicking Options and View Log.

If you experience difficulty uploading your data and you wish to review the upload settings that MapInfo EasyLoader last used, see [Locating the MapInfo EasyLoader Configuration File](#).

Locating the MapInfo EasyLoader Configuration File

MapInfo EasyLoader saves a configuration file called MapInfoEasyLoader.Recent.xml to your system under the %AppData% folder (for example, C:\Users\\AppData\Roaming\MapInfoEasyLoader.Recent.xml).

This file lists the MapInfo EasyLoader settings for the last upload, such as output folder location, connection settings, and data tables uploaded. Look at this file when you experience difficulty uploading your data and you wish to review the upload settings that MapInfo EasyLoader last used.

Do not edit this file. Deleting the configuration file returns MapInfo EasyLoader to its default settings and MapInfo EasyLoader replaces the configuration file when you next start it.

The AppData folder is in a path like C:\Users\\AppData. If you cannot see the AppData folder, it is because it is hidden. To locate it, on the Microsoft Start menu in the Search programs and files field type %AppData%.

4 - History of Changes

This section provides a history of features and enhancements that have been added in each version of MapInfo EasyLoader.

In this section

New in MapInfo EasyLoader 17.0	39
New in MapInfo EasyLoader 16.0	39
New in MapInfo EasyLoader 15.2	40
New in MapInfo EasyLoader 15.0	40
New in MapInfo EasyLoader 12.5.1	41
New in MapInfo EasyLoader 12.5	41
New in MapInfo EasyLoader 12.0	41

New in MapInfo EasyLoader 17.0

There are new command-line flags available:

- **Do Not Treat Empty (Zero-Length) Strings as Null: /NoTreatEmptyStringAsNull**

New in MapInfo EasyLoader 16.0

MapInfo EasyLoader is now a 64-bit release that works with MapInfo Pro 64-bit releases 15.2 and 16.0. As a result, it uses 64 bit Oracle and ODBC drivers. Oracle Text is not supported.

MapInfo EasyLoader has a new look and feel that makes it easier to use and gives you new features. With this release you can:

- Pause and cancel your uploads. For an overview of the new interface, see [Getting Started](#).
- Work with both native MapInfo TAB files and NativeX (extended) TAB files.
- Work with files in different character encodings, because MapInfo EasyLoader now supports Unicode.
- Apply error threshold options for better error handling. For details, see the descriptions for Table Errors Threshold % and Insert Errors Threshold % settings under [Server Table Processing Options](#). These settings are made in the **Options** dialog.
- Use keyboard shortcuts as a faster way to access commands. For details, see [Shortcuts by Keystroke](#).

Several files are available for tracking and logging MapInfo EasyLoader settings:

- EasyLoader.log contains the database connection information currently in use to upload data tables with. For details, see [Locating the MapInfo EasyLoader Log File](#).
- MapInfoEasyLoader.Recent.xml lists the MapInfo EasyLoader settings for the last upload, such as output folder location, connection settings, and data tables uploaded. For details, see [Locating the MapInfo EasyLoader Configuration File](#).
- MapInfoEasyLoader.Settings.rsp contains all of the flags used to start MapInfo EasyLoader. For details, see [Locating the MapInfo EasyLoader Command-Line Flags RSP File](#).

The command-line flags that you use when launching MapInfo EasyLoader from a command-line have been updated. They now have longer names that are more user friendly. Instead of using /O to set the oracle connection string, you can use [/OracleConnectionString](#). Shorter flags, like /O, still work, but using the longer flags is more intuitive. Also, there are new command-line flags available:

- **Help: /?, /Help**

- **Set the Application Locale : /AppLocale**
- **Input Command File: /CF, /CommandFile**
- **Convert WKB on Oracle: /ConvertWKBOnOracle**
- **Table Folder: /Folder**
- **Start Application Minimized: /Min**
- **Do Not Auto Start the Table Load: /NoStart**
- **Run Without User Interface: /NoUi**
- **Save Command Response File: /SaveCommandArgs**

The following command-line flags have been retired and are no longer available.

- /R Replace the server table. You can choose to replace the server table when using the new **/Folder** flag, which uploads all tables from a folder.
- /V Oracle version. MapInfo EasyLoader gets the Oracle version from the Oracle connection string.

New in MapInfo EasyLoader 15.2

MapInfo EasyLoader 15.2 is a 32-bit release that works with the MapInfo Pro 15.2 64-bit release. MapInfo Pro 15.2 supports Unicode and requires the PostgreSQL Unicode ODBC drive. MapInfo EasyLoader does not yet support Unicode and requires the PostgreSQL ANSI ODBC drive. You must have both drivers installed when both products are on the same machine.

This release fixes the following issue:

Issue Number	Description and Resolution
MIPRO-49213	<p>Oracle: Problems with MapInfo EasyLoader and EPSG-Code 25832.</p> <p>When MapInfo EasyLoader uploads a table, it does not put a SRID entry in the Oracle meta data table (SDO_GEOM_METADATA) and the resulting geometry column has NULL in the SDO:SRID column. As a result, the spatial index is not generated and MapInfo Pro cannot generate geographic queries.</p>

New in MapInfo EasyLoader 15.0

MapInfo EasyLoader 15.0 is a 32-bit release that works with the MapInfo Pro 15.0 32-bit release. There are no new features or fixes.

New in MapInfo EasyLoader 12.5.1

MapInfo EasyLoader 12.5.1 is a 32-bit release that works with the MapInfo Pro 12.5.1 64-bit release. There are no new features or fixes.

New in MapInfo EasyLoader 12.5

This release fixes the following issue:

Issue Number	Description and Resolution
MIPRO-40742	Non-English versions of MapInfo EasyLoader do not launch.

New in MapInfo EasyLoader 12.0

MapInfo EasyLoader 12.0 supports the following database versions:

- PostGIS 2.0 – Note that when uploading files to a PostGIS 2.0 database, spatial objects upload as a geometry spatial data type.
- SQL Server 2012 – MapInfo EasyLoader processes any invalid geography using the SQL Server 2012 MakeValid() function, so that the geography is valid.

The server table processing option Grant Public Access to Table has been changed to Grant Public Full Access to Table.

This release fixes the following issues:

Issue Number	Description and Resolution
MIPRO-12309	A message displays when appending a native table with a MI_PRINX column in an Oracle Spatial database with an OCI connection, "Table cannot be Appended—It has a different table structure."

Issue Number	Description and Resolution
MIPRO-12320	When uploading a table with no geometry into MapInfo EasyLoader 11.5, an error message appears: "Table cannot be Appended—It has a different table structure."
MIPRO-32642	MapInfo EasyLoader is unable to import to "geography" data type in SQL Server 2008 SP2. You will need to upgrade to SQL Server 2012. SQL Server 2012 uses the function STIsValid() to check if a geography data type is well formed and recognized as a valid object based on its OGC type. The MakeValid() function is also used that converts an invalid geography data type into a valid one. In MapInfo Pro 12.0, you can use the extended spatial methods in SQL Server 2012 to manage valid geography data types. Refer to the SQL Server 2012 MSDN library for more information.

Notices

© August 2018 Pitney Bowes Software Inc.

Information in this document is subject to change without notice and does not represent a commitment on the part of the vendor or its representatives. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, without the written permission of Pitney Bowes Software Inc., One Global View, Troy, New York 12180-8399.

© 2018 Pitney Bowes Software Inc. All rights reserved. Pitney Bowes Software Inc. is a wholly owned subsidiary of Pitney Bowes Inc. Pitney Bowes, the corporate logo, MapInfo, Group 1 Software, and MapInfo EasyLoader are trademarks of Pitney Bowes Software Inc. All other marks and trademarks are property of their respective holders.

Contact information for all Pitney Bowes Software Inc. offices is located at:

<http://www.pitneybowes.com/us/contact-us.html>.

libtiff © 1988-1997 Sam Leffler, © 2018 Silicon Graphics International, formerly Silicon Graphics Inc. All Rights Reserved.

libgeotiff © 2018 Niles D. Ritter.

Portions thereof LEAD Technologies, Inc. © 1991-2018. All Rights Reserved.

Portions © 1993-2018 Ken Martin, Will Schroeder, Bill Lorensen. All Rights Reserved.

ECW by ERDAS © 1993-2018 Intergraph Corporation, part of Hexagon Geospatial AB and/or its suppliers. All rights reserved.

Portions © 2018 Intergraph Corporation, part of Hexagon Geospatial AB. All Rights Reserved.

MrSID, MrSID Decompressor and the MrSID logo are trademarks of LizardTech, a Celartem company, used under license. Portions of this computer program are copyright © 1995-1998 LizardTech, A Celartem Company, and/or the University of California or are protected by US patent no. 5,710,835 and are used under license. All rights reserved. MrSID is protected under US and international patent and copyright treaties and foreign patent applications are pending. Unauthorized use or duplication prohibited.

Contains FME® Objects © 2005-2018 Safe Software Inc. All Rights Reserved.

© 2006-2018 TomTom International BV. All Rights Reserved. This material is proprietary and the subject of copyright protection and other intellectual property rights owned or licensed to TomTom. The use of this material is subject to the terms of a license agreement. You will be held liable for any unauthorized copying or disclosure of this material.

This product contains 7-Zip, which is licensed under GNU Lesser General Public License, Version 3, 29 June 2007 with the unRAR restriction. The license can be downloaded from

<http://www.7-zip.org/license.txt>. The GNU License may be downloaded from <http://www.gnu.org/licenses/lgpl.html>. The source code is available from <http://www.7-zip.org>.

Products named herein may be trademarks of their respective manufacturers and are hereby recognized. Trademarked names are used editorially, to the benefit of the trademark owner, with no intent to infringe on the trademark.



3001 Summer Street
Stamford CT 06926-0700
USA

www.pitneybowes.com