

Uploading TAB Data to Remote Databases

Introduction

EasyLoader v 7.0 is a Windows-only utility available from MapInfo Corporation that allows you to upload MapInfo .tab files to a remote database, such as MapInfo Professional for SQL Server. The spatial information in the .tab files is maintained in the SQL Server database and is available for viewing and analyzing in MapInfo Professional. EasyLoader can upload a text object to SQL Server if text-supported SpatialWare (4.6 or later) is found on the server.

EasyLoader is installed into the Tools directory during the MapInfo Professional installation process.

The following databases are supported:

Oracle Spatial

Informix Universal Server with SpatialWare DataBlade

SQL Server

Microsoft Access

For spatial database support, the DBMS needs to have the ability to handle spatial geometry either by itself, like Oracle Spatial, or via additional SpatialWare DataBlade/ Extensions, like MapInfo SpatialWare for IUS, and SQL Server. To support the spatial database, the IUS driver must have a version of 2.8 or greater. EasyLoader does not support spatial object types via SpatialWare for Oracle, which is supported by a different utility provided by MapInfo Corporation. If an above DBMS does not have spatial object type support, the table can only be uploaded as XY data (XY or XY with MapInfo Key (MICode)). Only one server connection may be open at any one time.

Running EasyLoader

To upload MapInfo .tab files using EasyLoader:

1. Run the EasyLoader from the MapInfo Professional Tools menu. If necessary, run Tool Manager to load and/or autoloading EasyLoader onto the Tools menu. The main EasyLoader dialog displays.

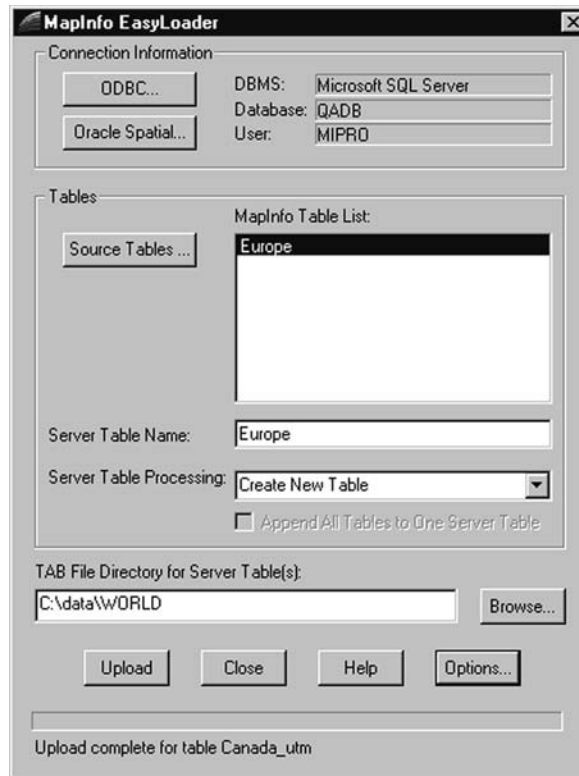
Uploading TAB Data to Remote Databases

2. Under Connection Information, click the appropriate button (ODBC or Oracle Spatial) to connect your database. Provide the necessary connection information (e.g., data source name or User ID, password and server name). Click OK to return to the EasyLoader dialog box.
3. Click the Source Tables button to display a list of MapInfo tables from a single directory. When tables are selected for uploading, the names will display in the MapInfo Tables list box.

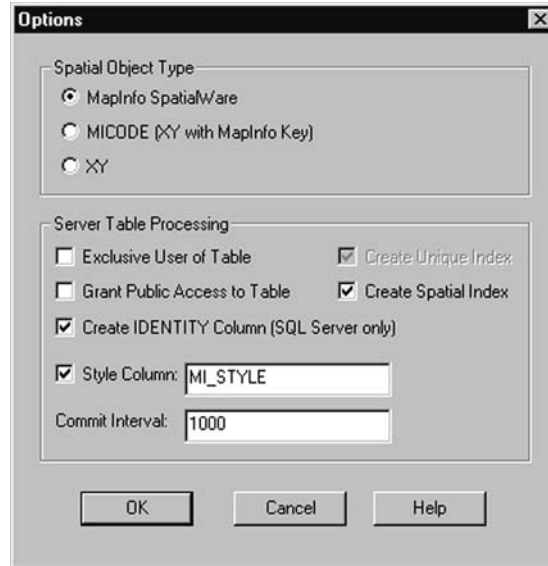
4. Choose the tables and select the appropriate Server Table processing task (Create new table, Append to existing table, Replace existing Table). These and additional options are explained in the Options section.

Note: The Upload button is not available until table(s) are chosen.

5. To create local TAB files, provide a directory or browse to its location. By default, EasyLoader will not generate these files. The file naming convention for these tables is *yourServerTableName_srv.tab*.



- To set options for the upload process, click the Options button. The Options dialog will display. See Options section for an explanation of available options.



- Click the Upload button to start the upload process. Close EasyLoader once the upload process is finished.

If you haven't already created the Spatial Index during the upload process, do so now by either executing a create index statement or re-uploading the table, making sure this time to create the Spatial Index and replace the table (see steps 1-3).

Loading Oracle Spatial Data

Loading for Oracle Locator

EasyLoader will load data for Oracle Locator. Loading data for Oracle Locator is the same as loading data for Oracle Spatial.

Loading Lat/Long Tables into Oracle 9i

When loading tables that use the Latitude/Longitude coordinate system(Geodetic Data) to Oracle 9i, it's important to verify that all geometry coordinates are between (-

Uploading TAB Data to Remote Databases

180,180) longitude and (-90, 90) latitude. Geodetic data coordinates beyond that range are not supported in Oracle Spatial and may cause problems. You may check your data using MapInfo Professional before loading, or by using the Oracle Spatial SDO_GEOM.VALIDATE_LAYER() function on the table after loading it to Oracle Spatial.

Note: EasyLoader assigns the default datum World Geodetic System 1984/WGS84 to the datumless Lat/Long coordinate system when loading Oracle 9i.

Validating Oracle Data

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

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

These functions may result in validation errors simply due to the tolerance level set by EasyLoader. These are errors such as; “ORA-13356 adjacent points in a geometry are redundant”, or “ORA-13022 polygon crosses itself”. The tolerance within the USER_SDO-GEOM_METADATA may be adjusted downwards (by a factor of 10) and the validation re-run. However, if you adjust the tolerance, you must recreate the spatial indexes because they use the tolerance when they are created.

EasyLoader Options

The options available in EasyLoader are described below.

Table Processing Options (Main Dialog)

Create New Table

A server table will be created with the name that you specify. If this option is chosen and a table with the same name already exists on the server, an error message will display, making you aware of this problem. You will need to use a different name or choose the option: Replace Existing Table, in order to upload the table.

Replace Existing Table

If a server table of the same name already exists, it will be dropped and a new table will be created to match the MapInfo table being uploaded when this option is selected.

Append to Existing Table

The MapInfo table will be appended to the server table if the server table exists and the structure of the two tables match. Otherwise, you will get an error and the table will not be uploaded. The tables **MUST** have the same table structure and be in the same projection for Oracle Spatial.

Append All to One Table

All MapInfo tables listed are uploaded to a single server table. The server table name is the one visible in the Server Table box. This feature is meant to be used to upload tables with the same structure and projection to one table. For example, instead of creating a new table for each street layer, check the Append All to One box, and only one table will be created. All of the tables will then be appended to this table. *Note:* It is possible that some tables will not be appended if their structure differs.

When this option is used with the Replace Table option, the server table will be dropped, a new table will be created, and all tables listed will be appended to that one. When used with the Create Table options, the server table will be created, and all tables listed will be appended to that one.

All tables **MUST** have the same table structure and be in the same projection for Oracle Spatial if this option is chosen.

Spatial Object Type

Choose from MapInfo SpatialWare or Oracle Spatial (depending on the type of connection), MICODE (XY with Key), and XY. The default for loading spatial data is MapInfo SpatialWare or Oracle Spatial.

MapInfo SpatialWare/Oracle Spatial

To select this option, the server must be Oracle Spatial or has MapInfo SpatialWare installed. Tables will be uploaded as spatial data.

XY and XY with MapInfo key (MICODE)

Uploading TAB Data to Remote Databases

Use this option if the server is not Oracle8i or does not have MapInfo SpatialWare installed. The data will be stored as xy coordinates on the server. Therefore, the server table will be created as a point table. If the MapInfo table to be uploaded is not a point table and this option is chosen, the centroid will be abstracted and stored on the server table if you instruct it to do so. The difference between XY and MICODE is that the MICODE will provide a MapInfo key as the spatial index, hence its performance is superior to XY.

Table Processing Options (Options Dialog)

Grant Public Access to Table

PUBLIC is granted all access to the server table.

Exclusive Use of Table

You can speed up load time on large tables significantly if you know that you will be the only one attempting to update/upload the table. By NOT checking this option, the loader will verify after each commit that no other updates are made to the table while it is being uploaded. Checking this option will prevent this check from occurring which can make a significant change to the run time for large tables.

Create Unique Index

A unique index is created on the column `sw_member` for SpatialWare, `mi_prinx` for Oracle, or `mi_sql_rec_num` for XY and MICODE. These columns are sequential numbers that are generated by the loader. These columns are always created, but do not have to be indexed.

Create Spatial Index

For SpatialWare tables the index is created on the geometry column called `hg<table_name>ind`. A spatial index is created and Update Statistics is executed after an r-tree index is created for SpatialWare. You may also build your own spatial index to suit your specific needs. If you choose to do this, clear this check box to save time in loading.

For Oracle Spatial tables the spatial index is created on the geometry column and is called `<table_name>_SX`. 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 8. After the index is built the ANALYZE table function is run on the index table. The spatial index will be R-Tree for Oracle8.17 or later.

Create IDENTITY Column (SQL Server only)

Check this box if you wish to create the key column (sw_member) with IDENTITY properties. When this feature is in use, the unique key column values will be generated automatically by SQL Server. You do not need to fill in the key manually when a new row is inserted.

In version 7.0, the key column(sw_member) is created with IDENTITY property by default, which is opposite from the previous version(6.8). To upload the table without IDENTITY select the Options button and uncheck Create IDENTITY Column

Style Column

This allows you to specify that per row symbology is to be loaded with the data. The symbology is loaded as a text string in the column specified. The name of the column to be used may be specified in the edit text box. The name is initialized to the default column name, which is MI_Style.

Note: To load per row symbology, the MapInfo Mapcatalog for the database must contain the following columns RENDITIONTYPE, RENDITIONCOLUMN, RENDITIONTABLE, and NUMBER_ROWS. See the section MapInfo MapCatalog for more information.

Command Line Flags

Easyloader supports the following command lines:

- **/A Append all tables to one** - This flag allows multiple tables to be uploaded to a single table as long as the table structures are the same.
SYNTAX: /A
- **/D Server CreateTable** - Generates TAB files to access remote DBMSs upon the your providing TAB file directory. By default, an empty directory, the loader will not generate TAB files. The newly generated TAB file is the Server Table Name plus _srv.tab

The directory must be a valid directory(empty is treated as valid) to upload a table.

Uploading TAB Data to Remote Databases

Command line option is /D PathName.

SYNTAX /D "C:\temp"

- **/E Exclusive Use of Table** - You can speed up load time on large tables significantly if you know that you will be the only one attempting to update the table. Note however, that specifying this flag does not guarantee that loader will obtain exclusive use, you must guarantee that to the loader. The loader checks on the current maximum value of the primary key column (prinx) after each commit to ensure that it detects any other entries that may have been made by other processes. This flag will prevent that check from occurring, which can make a significant change to the run time for large tables.

This flag may be placed within a shortcut, allowing the interactive use of the Loader interface for other functions.

SYNTAX: /E

- **/F Log File name** - The log file is always produced, this flag lets the user specify a log file name and location. EasyLoader.log is the default name and it will be created in the temp directory. If no path is provided, the log will be created in the same directory as the EasyLoader.EXE file.

SYNTAX: The first example shows specifying just the name of the log file, which will be written to the directory that EasyLoader.exe is in, the second specifies the full path for the log file.

/F myLogFile.txt

/Fc:\temp\myLogFile.txt

- **/G Grant all** - This flag will grant all rights to PUBLIC. This flag is turned OFF by default.

SYNTAX:/G

- **/I DO NOT Create a Spatial Index** - By default a spatial index will be created. This flag will cause a spatial index NOT to be created on the table. This flag is turned OFF by default, meaning a Spatial Index will be created. For IDS/UDO a Spatial Index will be created then the 'update statistics medium' statement will be issued. See /U which controls the unique index. For Oracle Spatial tables the spatial index is created on the geometry column and is called <table_name>_SX, for SpatialWare tables the index is created on the column geometry column and is called hg<table_name>ind.

SYNTAX: /I

- **/K Create Automated Key Column for SQL Server** - The key column (sw_member) can be created with IDENTITY property as an option. When this feature is in use, the unique key column values will be generated automatically by SQL Server and the user does not need to fill the key manually when a new row is inserted. In order for this feature to work, you need MapInfo Professional 6.5.0.3100 or later and EasyLoader 6.8 or later. In version 7.0, the key column(sw_member) is created with IDENTITY property by default, which is opposite from the previous version(6.8). Therefore, by not specifying the K option in the command line will have the same action as specifying it as /K, i.e. create the key column with IDENTITY property. If you want to turn off the property, you need to provide a keyword "NO_IDENTITY" following K

Example: /K NO_IDENTITY

See, Mixing Command Line Options with GUI.

SYNTAX: /K

- **/L List of MapInfo tables** - This flag allows the user to specify a text file that contains a list of tables to be uploaded. The format of each line is the same as the /T flag.

SYNTAX: /LListOTables.txt

- **/M MICCODE/XY** - this flag allows the user to specify the object type to be used if it is SpatialWare. If the /M flag is used, the user has to provide MICCODE (for XY with MapInfo key) or XY (for XY) after /M. Any words other than MICCODE or XY after /M are treated as an error. If the /M flag is not used, SpatialWare will be used as the default if the selected database has SpatialWare installed.

SYNTAX:/M micode

/M xy

- **/O Connection String** - This flag allows a connection string for Oracle Spatial to be passed to the program. See /S flag for ODBC connections.
SYNTAX: /O user_name/password@server_name
- **/P ACR** This flag specifies what to do with the table(s) being loaded to the server. A is append to an existing server table, C is create a new server table, and R is replace an existing table. You must specify only one of the three choices. The create table flag will fail if a table by the same name exists on the server.

Uploading TAB Data to Remote Databases

SYNTAX: /P A

- **/Q Quit**

This flag forces the loader to exit when done.

SYNTAX: /Q

- **/R Replace the server table**

This flag will cause the server table to be dropped. The new table will be created and uploaded. By default a will be created on the server table. (This is a change from 6.0)

SYNTAX: /R/

- **/U DO NOT Create a Unique Index**

A Unique index will be created by default. This flag will cause a unique index NOT to be created on the table. This flag is turned OFF by default, meaning a unique Index will be created by default. See /I which controls the spatial index. For Oracle Spatial tables the unique index is created on the column PRINX and is called <table_name>_IDX, for SpatialWare tables the index is created on the column SW_MEMBER and is called <table_name>_i

SYNTAX: /U

- **/V Oracle Version**

This flag allows you to load tables on an Oracle 8.1.6 server with the 8.1.5 format. This not generally recommended, but it is available if you have a special need to do this.

If you want to use this flag with the dialog, see the Windows Shortcut in the Running Command Line flag section.

SYNTAX: /V

- **/T MapInfoTable Name;ServerTable Name;Range**

This flag allows a single table name to be passed to the program. The separator between the MapInfo table name, the server name and the range is the semi-colon symbol. The range is in the format: starting number <COMMA> ending number. The server table name and the range are optional.

SYNTAX: /Tc:\data\states.tab;mystates;1,500

- **/X Commit interval**

This flag allows the user to specify a commit interval. Easyloader will commit the inserted records when the commit interval is reached. The default commit interval is 1000. This same interval applies to the creation of the spatial index for Oracle Spatial. If the commit interval is set to 0 (zero), the whole range of records will be inserted as a single transaction, before a commit is issued.

SYNTAX: /X 500

- **/Y Symbology Column Name**

This flag allows you to specify that per row symbology is to be loaded with the data. You can also specify the name of the column to be used, if none is provided the default column name will be used, which is MI_SYMBOLGY.

SYNTAX: /Y StyleColumnName

Mixing Command Line Flags with GUI

Command line flags may be mixed with the UI by using a windows shortcut. This makes it easier to set flags as your default while being able to override them from the UI. There are some flags only available from the command line.

- Create a shortcut to EasyLoader.
- Right click on the shortcut and choose Properties.
- Under the "Shortcut" tab, within the "Target" edit box, add on the command line flags at the end of the line.

When EasyLoader is run from that shortcut, the specified flags are in effect.

MAPINFO_MAPCATALOG

The MAPINFO_MAPCATALOG is a registry table for databases that stores metadata about geometry tables in the database. Using the tablename and ownername 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 Professional, that access map data from databases.

If a MAPCATALOG does not exist, it can be created during the upload process when running EasyLoader. When using ODBC, EasyLoader will not issue public grants, which must be done by other means. If you do not have adequate permissions then creation will not succeed and the table will not be uploaded.

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 IUS, SQL Server, or Oracle Spatial, and the symbol clause will have only the information for a polygon.

MAPINFO_MAPCATALOG Format

The MAPINFO_MAPCATALOG has the following table structure:

SPATIALTYPE	FLOAT
TABLERNAME	CHAR(32)
OWNERNAME	CHAR(32)
SPATIALCOLUMN	CHAR(32)
DB X LL	FLOAT
DB Y LL	FLOAT
DB X UR	FLOAT
DB 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

A script, which is shipped with EasyLoader, may be used to modify an existing MAPINFO.MAPINFO MAPCATALOG to add the rendition columns if they do not exist. This script is to be run as user MAPINFO.

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

Additional Resources

For more information on using EasyLoader, please view the online help provided with this utility, "EasyLoader.HLP".

Uploading TAB Data to Remote Databases
