MapInfo®

Where Opportunity Is Located™

MapInfo EasyLoader

v7.8

User Guide

MapInfo Corporate Headquarters:

Voice: (518) 285-6000

Fax: (518) 285-6060

Sales Info Hotline: (800) 327-8627

Government Sales Hotline: (800) 619-2333

Technical Support Hotline: (518) 285-7283

Technical Support Fax: (518) 285-6080

Contact information for North American offices is located at:**http://www.mapinfo.com/company/company_profile/index.cfm**

Contact information for European and Middle East offices is located at: **http://www.mapinfo.com/company/contact_europe.cfm**

Contact information for Asia Pacific offices is located at: **http://www.mapinfo.com/company/contact_asiapacific.cfm**

November 2004

Uploading Data with MapInfo EasyLoader

Welcome to the EasyLoader User Guide!

EasyLoader is a utility that facilitates the upload of MapInfo .tab data files to remote databases. EasyLoader can be used as a stand-alone tool or a MapBasic tool from the Tools menu within MapInfo Professional.

In this chapter:

What is the EasyLoader?

EasyLoader v 7.8 is a utility available from MapInfo Corporation that allows you to upload MapInfo .tab files to a remote database, such as MapInfo SpatialWare 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 also upload a text object to SQL Server if text-supported SpatialWare (4.6 or later) is on the server.

EasyLoader is installed into the Tools directory during the MapInfo Professional installation process. EasyLoader supports the following databases:

- Oracle
- IBM Informix
- SQL Server
- Microsoft Access

For spatial database support, the DBMS must be able to handle spatial geometry, either by itself (as in Oracle Spatial), or via extensions such as IBM Informix® Spatial DataBlade™, or MapInfo SpatialWare (for Informix or SQL Server). To support IBM Informix® Spatial DataBlade™, you must have Informix driver version 2.8 or higher. If one of the 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.

Uploading Data Using the EasyLoader

To upload MapInfo .tab files using EasyLoader:

1. In MapInfo Professional, run the EasyLoader. From the **TOOLS** menu, click EasyLoader and select **EASYLOADER** from the submenu to display the EasyLoader dialog box.

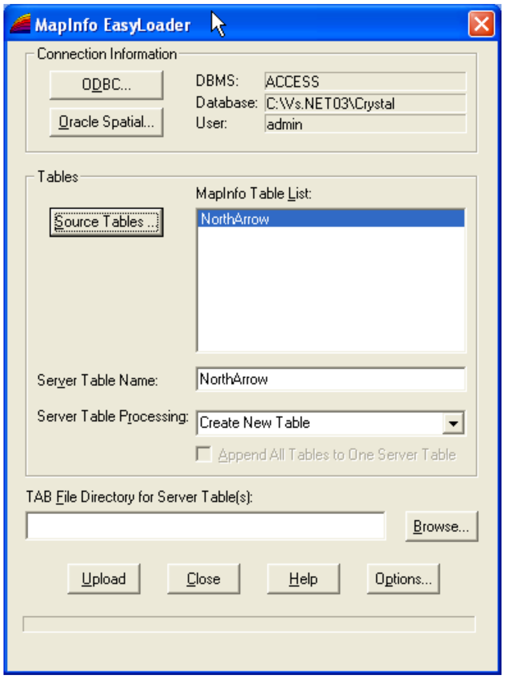
To load the EasyLoader for this session only:

a. From the **TOOLS** menu, click **TOOL MANAGER.**

b. Scroll to the **EASYLOADER** entry.

c. To load EasyLoader into the Tools menu for the current session of MapInfo Professional, check the **LOADED** checkbox. To automatically reload the EasyLoader for all subsequent sessions, check the **AUTOLOADED** checkbox as well.

Figure: MapInfo EasyLoader



DBMS: ACCESS
Database: C:\Vs.NET03\Crystal
User: admin

MapInfo Table List: NorthArrow

Server Table Name: NorthArrow

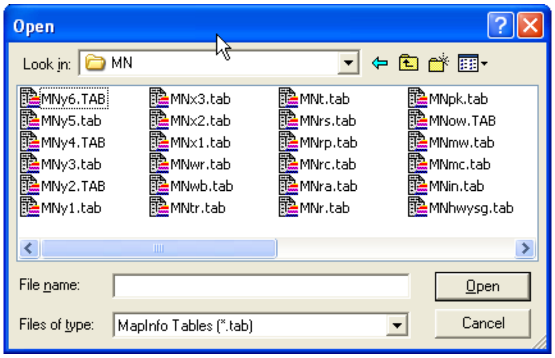
Server Table Processing: Create New Table

□ Append All Tables to One Server Table

TAB File Directory for Server Table(s):

2. Under **Connection Information**, click the appropriate button (**ODBC** or **Oracle Spatial**) to connect your database. Provide the necessary connection information (for example, 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 display in the MapInfo Tables list box.

Figure: Open



Look in: MN

MNy6.TAB
MNy5.tab
MNy4.TAB
MNy3.tab
MNy2.TAB
MNy1.tab
MNx3.tab
MNx2.tab
MNx1.tab
MNwr.tab
MNwb.tab
MNtr.tab
MNt.tab
MNrs.tab
MNrp.tab
MNrc.tab
MNra.tab
MNr.tab
MNpk.tab
MNow.TAB
MNmw.tab
MNmc.tab
MNin.tab
MNhwysg.tab

File name:

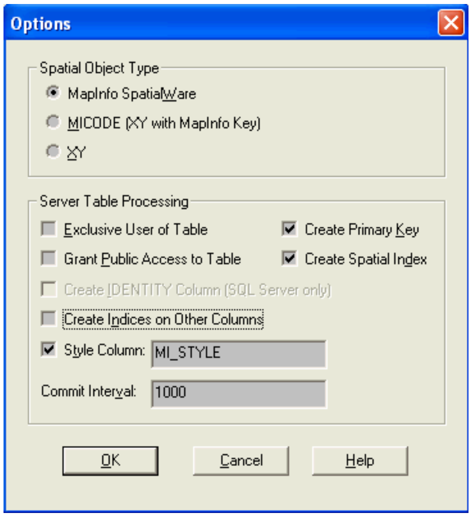
Files of type: MapInfo Tables (*.tab)

4. Choose the tables you want to upload 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 **Understanding the EasyLoader Options on page 7**.

   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 does not generate these files. The file naming convention for these tables is *yourServerTableName*_srv.tab.
6. To set options for the upload process, click the **Options** button. The Options dialog displays.

Figure: Options



Spatial Object Type

● MapInfo SpatialWare
○ MICODE (XY with MapInfo Key)
○ XY

Server Table Processing

□ Exclusive User of Table
☑ Create Primary Key
□ Grant Public Access to Table
☑ Create Spatial Index
□ Create IDENTITY Column (SQL Server only)
□ Create Indices on Other Columns
☑ Style Column: MI_STYLE
Commit Interval: 1000

See **Understanding the EasyLoader Options on page 7** for an explanation of available options. Select the appropriate options and click **OK** to confirm them.

7. Click the **UPLOAD** button to start the upload process.
8. **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 check the Create Spatial Index checkbox and replace the table (see steps 1-3).

Loading Oracle Spatial Data

You can upload Oracle Spatial data using the EasyLoader product.

Loading for Oracle Locator

The EasyLoader loads data for the Oracle Locator. Loading data for the Oracle Locator is the same as loading data for Oracle Spatial.

Loading Long/Lat Tables into Oracle 9i

When uploading tables that use the Longitude/Latitude coordinate system (Geodetic Data) to Oracle 9i, it is important to verify that all geometry coordinates are between (-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 any "datumless" Long/Lat 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 due to the tolerance level set by 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.

Understanding the EasyLoader Options

This section provides a comprehensive description of the options available in EasyLoader.

Table Processing Options (Main Dialog Box)

There are four table processing options associated with the main EasyLoader dialog box.

- **Create New Table**
  A server table is 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 need to 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.
- **Append to Existing Table**
  The MapInfo table is appended to the server table if the server table exists and the structure of the two tables match. Otherwise, you get an error and the table is not uploaded. The tables must have the same table structure and be in the same projection for Oracle Spatial.
- **Append All Tables to One Server 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 TABLES TO ONE SERVER TABLE** box, and only one table is created. All of the tables are then appended to this table.

  Note: It is possible that some tables will not be appended if their table structure differs.

  If the Replace Table option is chosen and this checkbox is checked, the server table is dropped, a new table is created, and all tables listed are appended to that one. If the Create Table option is chosen and this checkbox is checked, the server table is created, and all tables listed are appended to that one.

  If this option is chosen, all tables must have the same table structure and be in the same projection.

Spatial Object Type Options (Options Dialog Box)

Choose from **MAPINFO SPATIALWARE** or **ORACLE SPATIAL** (depending on the type of connection), **MICODE (XY WITH MAPINFO KEY)**, and **XY**. The default for loading spatial data is MapInfo SpatialWare or Oracle Spatial, if this option is available; otherwise the MICODE option is the default.

- **MapInfo SpatialWare/Oracle Spatial** – To select this option, the server must be **ORACLE SPATIAL** or have MapInfo SpatialWare installed. Tables are uploaded as spatial data. This option will be disabled (greyed out) if SpatialWare is not installed on the server, or if it is not available to the currently selected database.
- **MICODE (XY WITH MAPINFO KEY)** – Use these options if the server is not Oracle Spatial or does not have MapInfo SpatialWare installed. These options store 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 one of these options 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.

Server Table Processing Options (Options Dialog Box)

- **Exclusive User of Table**

  You can significantly speed up load time on large tables if you know that you are the only one attempting to update or upload the table. If you do not check this option, EasyLoader verifies after each commit that no other updates are made to the table while it is being uploaded. Checking this option prevents this test from occurring, which can improve runtime performance for large tables.
- **Create Primary Key**

  By selecting this checkbox, a primary key is created for the Create New Table and Replace Existing Table operations. This primary key is created in 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 EasyLoader. These columns are always created, but do not have to be a primary key.

  For the Append To Existing Table operation, the primary key is *not* created.
- **Grant Public Access to Table**

  The PUBLIC is granted all access to the server table.
- **Create Spatial Index**

  By selecting this check box, a spatial index (called <*table_name*>ind) is created for the tables on the geometry column. 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 loading time.

  By selecting this check box, a spatial index (called <*table_name*>_SX) is created for Oracle Spatial tables 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 8. After the index is built the ANALYZE table function is run on the index table. The spatial index will be R-Tree for Oracle 8.17 or later.

- **Create IDENTITY Column (SQL Server only)**

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

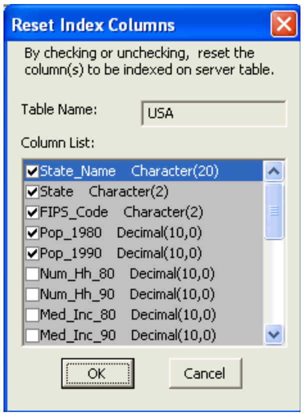
  In EasyLoader version 7.x, the key column (SW_MEMBER) is created with IDENTITY property by default, which is opposite from the previous version (6.8). To upload the table without IDENTITY, select the **Options** button and uncheck the **Create IDENTITY Column** checkbox.

- **Create Indices on Other Columns**

  Check this box if you want to select additional columns to index when uploading a new table or replacing an existing table. By default this box is not checked.

  If this box is checked, you will see the following dialog box when table upload begins after you click **OK** on the **Options** dialog box.

Figure: Reset Index Columns



Reset Index Columns

By checking or unchecking, reset the column(s) to be indexed on server table.

Table Name: USA

Column List:

☑ State_Name Character(20)
☑ State Character(2)
☑ FIPS_Code Character(2)
☑ Pop_1980 Decimal(10,0)
☑ Pop_1990 Decimal(10,0)
□ Num_Hh_80 Decimal(10,0)
□ Num_Hh_90 Decimal(10,0)
□ Med_Inc_80 Decimal(10,0)
□ Med_Inc_90 Decimal(10,0)

The Table Name textbox shows which server table is selected for indexing. The Column List shows each column followed by its data type. Check a box to select that column for indexing on the server table. Uncheck a box if you do not want to index that column.

Note: If the column name length is too long, the index creation will fail. The limit to the size of the index name will vary per database.

- **Style Column**

  This allows you to specify whether per-row styles are loaded with the data. You can also specify the name of the column in the text box next to the Style Column check box. The default column name is MI_STYLE.

  Note: To load per-row styles, the MapInfo Mapcatalog for the database must contain the following columns: RENDITIONTYPE, RENDITIONCOLUMN, RENDITIONTABLE, and NUMBER_ROWS. See **MAPINFO_MAPCATALOG Table Structure on page 14** for more information.

- Commit Interval

  Specify a Commit Interval for uploading. EasyLoader will commit 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 will be inserted as a single transaction, before a commit is issued.

Using the Command Line Flags

You can run the EasyLoader executable from the command line. For example:

```
easyloader.exe /T c:\data\states.tab;mystates /G /Y
```

Note: Do *not* enclose file names in quotation marks. Command line arguments will be interpreted correctly without quote marks, even if the filename includes spaces. If quotation marks are used, EasyLoader will not parse the filenames correctly and errors will occur.

EasyLoader supports the following flags to allow you to specify additional upload parameters from the command line:

- **/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`

- **/C Create Indices for All Locally-Indexed Columns** – When uploading a new table or replacing an existing table, this flag mandates that only the columns that were indexed on the local table will be automatically indexed on the server table.

  This command line option does *not* allow you to make or reset column selections. That additional capability is supported by the EasyLoader user interface (see **Create Indices on Other Columns on page 9**).

  Syntax: `/C`

- **/D .tab File Directory for Server Table(s)** – Generates .tab files to access remote DBMSs upon the your providing .tab file directory. The default is an empty directory, in which case EasyLoader will not generate .tab files. The newly generated .tab file is *yourServerTableName*_srv.tab. The directory must be a valid directory (empty is treated as valid) to upload a table.

  Command line option is /D *PathName*.

  Syntax: `/D C:\temp`

  Do not enclose file names in quotation marks.

- **/E Exclusive Use of Table** – You can improve 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 EasyLoader will obtain exclusive use; you must guarantee that to the loader. EasyLoader checks on the current maximum value of the primary key column (MI_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 significantly improve the upload time for large tables.

  This flag may be placed within a shortcut, allowing the interactive use of the EasyLoader 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. By default, a log file named EasyLoader.log will be created in the Windows TEMP directory. If a file name with no path is provided, the log will be created in the same directory as the EasyLoader.EXE file.

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

  ```
  /F myLogFile.txt
  ```

`/F c:\temp\myLogFile.txt`

Do not enclose file names in quotation marks.

- **/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 prevent a spatial index from being 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, and 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 EasyLoader version 7.x, the key column (SW_MEMBER) is created with IDENTITY property by default, which is opposite from the previous version (6.8). Therefore, omitting the K option in the command line will have the same action as specifying /K, (that is, it creates the key column with IDENTITY property). If you want to turn off the property, you must provide a keyword NO_IDENTITY following /K.

  See **Mixing Command Line Flags with the EasyLoader User Interface on page 13**.

  Syntax: `/K`

  Example: `/K NO_IDENTITY`

- **/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: `/L ListOfTables.txt`

  Do not enclose file names in quotation marks.

- **/M MICODE/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 must provide MICODE (for XY with MapInfo key) or XY (for XY only) after /M. Any words other than MICODE or XY after /M are treated as errors, and EasyLoader will not run (the main dialog will not appear). If the /M flag is not used, SpatialWare will be used as the default if the selected database has SpatialWare installed.

  Syntax: `/M MICODE`

  Example: `/M XY`

- **/O Connection String** – This flag allows a connection string for Oracle Spatial to be passed to the program. See the /S flag for ODBC connections.

  Syntax: `/O user_name/password@server_name`

- **/P A | C | R** – This flag specifies what to do with the table(s) being loaded to the server. The **A** keyword means append to an existing server table; **C** means create a new server table; **R** means replace an existing table. You must specify only one of the three choices. If the C option is specified, the upload operation will fail if a table by the same name exists on the server.

  Syntax: `/P A`

- **/Q Quit** – This flag forces EasyLoader 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. A table will be created on the server even if the table did not exist originally.

  Syntax: `/R`

- **/S Connection String** – This flag allows an ODBC connection string to be passed to the program. If enough information to connect is supplied, the ODBC connection dialog will not appear. See the /O flag for Oracle Spatial connections.

  The following examples illustrate the syntax of this flag. The first example uses a datasource, the second supplies the full connection string.

  ```
  /SDSN= MyDataSource
  /SUID=MyId;DATABASE=MyDB;HOST=MyServer;SERVER=MyServer_tli;
  SERVICE=sqlexec;PROTOCOL=onsoctcp
  ```

- **/T MapInfo Table Name;Server Table 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 semicolon symbol. The range is in the format *starting number* <COMMA> *ending number*. The server table name and the range are optional.

  Syntax: `/T c:\data\states.tab;mystates;1,500`

  Do not enclose file names in quotation marks.

- **/U Do Not Create a Primary Index** – A primary key constraint will be added by default. This flag will prevent a primary key from being created on the table. This flag is turned OFF by default, which means that a primary key will be created by default. See **/I** which controls the spatial index. For Oracle Spatial tables, the primary key is created on the column MI_PRINX and is called <*table_name*>_PK. For SpatialWare tables, the primary key is created on the column SW_MEMBER and is called <*table_name*>_PK.

  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 accomplish this using the graphical interface, see **Mixing Command Line Flags with the EasyLoader User Interface on page 13**.

  Syntax: `/V`

- **/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 Style Column Name** – This flag allows you to specify whether per-row styles will be loaded with the data. You can also specify the name of the column to be used. If no name is provided, the default MI_STYLE column name will be created. If the /Y flag is followed by the NO_STYLE keyword, EasyLoader will not create a style column on the server table.

  Syntax: `/Y [`*`StyleColumnName`*` | NO_STYLE]`

Mixing Command Line Flags with the EasyLoader User Interface

Command line flags may be mixed with the 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.

- Create a shortcut to EasyLoader. Right-click and choose **SEND TO** > **DESKTOP** to create the shortcut on the desktop, or right-click and drag to move a shortcut to a folder of your choice.
- Right-click on the shortcut and choose Properties.
- Under the “Shortcut” tab, within the “Target” edit box, add the appropriate command line flags to the end of the line, separated by spaces.

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

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

MAPINFO_MAPCATALOG Table Structure

The 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

COORDINATESYSTEM CHAR(254)

SYMBOL CHAR(254)

XCOLUMNNAME CHAR(32)

YCOLUMNNAME CHAR(32)

RENDITIONTYPE INTEGER

RENDITIONCOLUMN CHAR(32)

RENDITIONTABLE CHAR(32)

NUMBER_ROWS INTEGER

Note: The following 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. Run this script 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".