

MapMarker 3.x Product Guide

Abridged Version for MapMarker 4.0 Beta Release



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 MapInfo Corporation, One Global View, Troy, New York 12180-8399.

©1992-1998 MapInfo Corporation. All rights reserved.

MapInfo Help ©1992-1998 MapInfo Corporation. All rights reserved.

MapInfo, MapInfo Professional, MapBasic, MapMarker and the MapInfo Logo are registered trademarks of MapInfo Corporation.

Contact MapInfo Corporation on the Internet at: <http://www.mapinfo.com>

MapInfo Corporate Headquarters:

Voice: (518) 285-6000

Fax: (518) 285-6060

Sales Info Hotline: (800) 327-8627

Federal Sales: (800) 619-2333

Technical Support Hotline: (518) 285-7283

Technical Support Fax: (518) 285-6080

MapInfo Europe Headquarters:

England

Voice: +44 (0)1753 848 200

Fax: +44 (0)1753 621 140

Germany

Voice: +49 6196 6700 0

Fax: +49 6196 6700 11

Toll-free telephone support is available in the U.S. and Canada. Contact your MapInfo sales representative for details. For international customers, please use the Technical Support Fax number.

Portions copyright (c) 1992-1995 Matchware Technologies, Inc., All rights reserved.

ZIP+4® is a registered trademark of the U.S. Postal Service.

Portions of the data are under license to MapInfo by Geographic Data Technologies, Inc., 11 Lafayette St., Lebanon, NH 03766 (c) 1990-1995.

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.

MapInfo welcomes your comments and suggestions.

**MapMarker 3.x Abridged
February 1998**

Table of Contents

Introduction	5
Chapter 1: What's New for MapMarker 3.x	7
Geocode Data Where It Lives: Direct Access via ODBC	8
U.S. Postal Service CASS Certification	8
Create Custom Address Dictionaries	8
Batch Geocoding	8
Chapter 2: About MapMarker	9
What Is MapMarker?	10
MapMarker's Geocoding Model	13
Comparing MapMarker with MapInfo's Geocoding Feature	14
Chapter 3: Using MapMarker	15
Before You Geocode	16
Running MapMarker	18
Identifying Address Columns	19
Geocoding Your Table Automatically	22
Geocoding Your Table Interactively	23
Displaying Geocoded Records in MapInfo	26
Geocoding to ZIP Code Centroids	27
Geocoding for CASS Certification	27
Geocoding Remote Tables	29
Accessing a Remote Table for Geocoding	34
Additional Considerations for Remote Table Geocoding	37
Batch Geocoding	38
Geocoding dBASE Files	41
Saving Settings to a Table	41
Setting Geocoding Preferences	42
Setting System Preferences	46
Chapter 4: Making the Most of MapMarker	49
Maximizing Match Rate and Accuracy	50
Choosing Match Strategies	52
Impact of Relaxing Match Conditions	55
Re-Geocoding a Table	56
Making the Most of Interactive Geocoding	56

Table of Contents

Understanding Result Codes	59
S3 and Z3 Result Codes: What's the Difference?.....	64
Census Block Codes	65
Performance Tuning	65
Creating a Customized User Dictionary	66
Understanding Datums	69
Chapter 5: MapMarker Client/Server Toolkit	72
Client/Server Geocoding	73
MapMarker Server	73
Running MapMarker Server	74
Geocoding Request Timed Out	76
MapMarker Client OCX	76
Adding MapMarker OCX to Your Client Application.....	77
Setting MapMarker OCX Properties	80
Geocoding via the Client OCX	81
Falling Back to ZIP Code Centroids	81
MapMarker OCX Events and Methods	82
The Toolkit and the Internet	84
Appendix A: Geocoding Terminology	85
Appendix B: Append Utility	89

Introduction

This document is an abridged version of the MapMarker 3 Product Guide for use as a reference during the beta testing period for MapMarker 4.0. It includes all the pertinent discussions from the Product Guide to help you learn MapMarker and run it successfully. Additionally, all supplemental documentation for MapMarker releases 3.1 through 3.4 have been incorporated into this document.

For information on the MapMarker 4.0 features and enhancements, see the supplemental document “Beta Release Notes” available from the MapMarker Beta web download area or on the Beta CD-ROM.

This document is organized as follows:

Chapter 1: What’s New for MapMarker 3.x

- Geocode Data Where It Lives: Direct Access via ODBC
- U.S. Postal Service CASS Certification
- Create Custom Address Dictionaries
- Batch Geocoding

Chapter 2: About MapMarker

- What Is MapMarker?
- MapMarker’s Geocoding Model
- Comparing MapMarker with MapInfo’s Geocoding Feature

Chapter 3: Using MapMarker

- Before You Geocode
- Running MapMarker
- Identifying Address Columns
- Geocoding Your Table Automatically
- Geocoding Your Table Interactively
- Displaying Geocoded Records in MapInfo
- Geocoding to ZIP Code Centroids
- Geocoding for CASS Certification
- Geocoding Remote Tables
- Accessing a Remote Table for Geocoding

- Additional Considerations for Remote Table Geocoding
- Batch Geocoding
- Geocoding dBASE Files
- Saving Settings to a Table
- Setting Geocoding Preferences
- Setting System Preferences

Chapter 4: Making the Most of MapMarker

- Maximizing Match Rate and Accuracy
- Choosing Match Strategies
- Impact of Relaxing Match Conditions
- Re-Geocoding a Table
- Making the Most of Interactive Geocoding
- Understanding Result Codes
- S3 and Z3 Result Codes: What's the Difference?
- Census Block Codes
- Performance Tuning
- Creating a Customized User Dictionary
- Understanding Datums

Chapter 5: MapMarker Client/Server Toolkit

- Client/Server Geocoding
- MapMarker Server
- Running MapMarker Server
- Geocoding Request Timed Out
- MapMarker Client OCX
- Adding MapMarker OCX to Your Client Application
- Setting MapMarker OCX Properties
- Geocoding via the Client OCX
- Falling Back to ZIP Code Centroids
- MapMarker OCX Events and Methods
- The Toolkit and the Internet

Chapter 1: What's New for MapMarker 3.x

Overview

MapMarker 3.0 has been enhanced with a number of new features and updates to existing features that help you to get the most out of geocoding your records. This chapter gives a brief summary of what's new. For details on a feature, see the appropriate chapter in this product guide.

In This Chapter . . .

Geocode Data Where It Lives: Direct Access via ODBC	8
U.S. Postal Service CASS Certification	8
Create Custom Address Dictionaries	8
Batch Geocoding	8

Geocode Data Where It Lives: Direct Access via ODBC

MapMarker includes expanded support for geocoding remote databases via ODBC. In this release MapMarker supports remote mappable tables using the XY spatial index on relational database management systems (RDBMS). The spatial object for a geocoded record can now be stored directly in the remote table.

Effective with version 3.4, MapMarker supports SpatialWare 2.2 on Oracle and the SW_GEOMETRY spatial data type so point objects are stored directly in the spatial database. For more on geocoding remote tables, see Chapter 3.

U.S. Postal Service CASS Certification

MapMarker version 3.x meets the USPS CASS requirements for address standardization including the ability to append ZIP+ 4 information to your data. This results in delivering even higher hit rates with improved accuracy. Moreover, the MapMarker Address Dictionary is updated every two months in compliance with CASS requirements. For more on geocoding for CASS certification, see Chapter 3, Using MapMarker.

Create Custom Address Dictionaries

It's easier than ever to create and use customized address dictionaries. This functionality uses the same indexing methods as the MapMarker Address Dictionary so you can create large address dictionaries from MapInfo tables and geocode to them without significant performance degradation. For instructions on creating a user dictionary, see Chapter 4, Making the Most of MapMarker.

Note: This functionality replaces the sample utility program Makeaddr.mbx from earlier MapMarker versions.

Batch Geocoding

Effective with version 3.3, MapMarker supports batch geocoding for processing one or more tables without constant user interaction. This feature is useful for overnight processing of large databases or when a table is updated regularly via a scheduling program.

MapMarker reads a text file that you create containing the path and file names of the tables to geocode and the geocoding preferences you want applied to the session. You can create the text file through MapMarker interface or generate one by hand.

Chapter 2: About MapMarker

Overview

MapMarker contains a wealth of features that allows you to geocode your database the way you want. Choose from automatic batch processing, or selecting a match one record at a time. Control where the record geocodes to: street level or ZIP Code centroid. Use the MapMarker Dictionary or create one of your own. All this and more can be found in MapMarker 3.x.

In This Chapter . . .

What Is MapMarker?	10
MapMarker's Geocoding Model	13
Comparing MapMarker with MapInfo's Geocoding Feature.	14

What Is MapMarker?

Street-level Matching

MapMarker is a street-level geocoding tool that allows you to turn non-geographic data records into geographic ones so that they can be displayed on a computerized map. MapMarker does this by matching your address records containing street-style addresses against a search table that contains address information and geographic (longitude and latitude) coordinates. It then assigns coordinates to the record and optionally creates a point.

ZIP Code Centroid Matching

As an alternative to street-level geocoding, you can tell MapMarker to geocode your records to the appropriate ZIP Code centroid for the record. This is the fastest method of geocoding because it eliminates the need to match based on street address. This method matches the ZIP Code in your table with the ZIP Code in the Address Dictionary. Use ZIP Code Centroid matching when you have a large database that you want to geocode quickly and you do not need to spot points at street level.

MapMarker can geocode to street level and ZIP Code centroid during the same pass through your table. It always uses the highest accuracy possible when matching to ZIP Codes.

Address Dictionary

MapMarker's search table, called the Address Dictionary, contains street addresses and ZIP+4 centroids for the entire US. With CD-ROM unlocking, you can purchase the right amount of matching data for your needs such as the entire U.S. or individual states. You can match your database of local, regional or nationwide addresses in a single geocoding pass. You do not even have to open, display, or specify the name of the search table. MapMarker handles it automatically. You only need to open your data table, tell MapMarker the columns that contain the address information and run MapMarker. MapMarker automatically matches your data records against the appropriate data and assigns the geographic coordinates.

The Address Dictionary is based on three sources of address and geographic information:

- Addresses and ZIP+4 data from the U.S. Postal Service ZIP+4 database
- Street geometry and related information from the U.S. Census Bureau's TIGER 95 files
- ZIP+4 centroids from GDT, Inc.

The Address Dictionary is updated bimonthly in keeping with USPS CASS requirements.

In addition to the MapMarker Address Dictionary, you can create your own customized address dictionary to contain localized information and new addresses. Both dictionaries can be used at the same time to geocode your table. The information in your custom dictionary takes precedence over the MapMarker Dictionary.

Automatic and Interactive Geocoding

MapMarker runs in both automatic and interactive mode. Use automatic on your first pass to geocode most of your table. At the end of the pass, MapMarker displays a summary of matched and unmatched records. The matched records now have coordinates assigned to them and are ready to display on a map as point objects, differentiated according to matching precision.

With the unmatched records, you can run MapMarker interactively to individually choose the best match from a list of suggestions. You can also expand the search for a match, change geocoding settings, go backwards in the table and edit the record directly in the MapMarker dialog.

Accepts MapInfo TAB Format, dBASE Files or ODBC Tables

MapMarker will read any MapInfo table (.tab format) or any dBASE (.dbf) format table.

In addition, MapMarker, via Open Database Connectivity (ODBC) can geocode data stored on remote databases, including MS Access, Informix, SQL Server, Sybase, Oracle and SpatialWare 2.2 on Oracle.

Adds Columns to Your Table

MapMarker will automatically add columns to your MapInfo or dbf table to store longitude, latitude and result codes. For ODBC tables, be sure to add these columns before geocoding with MapMarker.

Creates Points Automatically

MapMarker creates point objects for geocoded records stored in .tab, dBASE or remote tables automatically. You are then ready to display them using MapInfo. As an option, MapMarker can write the longitude and latitude coordinates directly to your table.

Keep in mind that points created with MapMarker will overwrite any points previously created by geocoding with MapInfo.

Returns Result Codes

MapMarker returns a result code for each record it attempts to match so you can see whether a match was made and how precisely each address component matched. The codes are stored in your table. Double-click on the code in the Browser window to display a full explanation of the matching result.

MapMarker creates points that display with color-coded symbols in a MapInfo Map window. Each color represents a different result code category. This allows you to instantly see the distribution of street-level matches versus matches made to ZIP Code centroids, etc.

Geocodes to Place Names

In addition to geocoding records with street style addressing, MapMarker will automatically geocode records containing place names. MapMarker's Address Dictionary contains place names for a wide variety of large single mail drop locations, significant buildings and organizations such as Sears Tower and Candlestick Park. Because MapMarker is a CASS-certified geocoder, MapMarker matches on firm name.

Note: place name geocoding is not available when geocoding to ZIP+4 centroids.

Customizable with Application Program Interface

MapMarker comes complete with an Application Program Interface (API) that allows you to add MapMarker to your custom Windows or UNIX applications. For a full discussion, see Part II of this product guide (not included in this abridged version).

MapMarker's Geocoding Model

MapMarker is based on a model of relative matching that is governed by a set of weights that score each portion of the address against candidate records (possible matches) in the Address Dictionary. The resulting scores are summed and the candidate's total score is used to determine the best match or matches. An exact match is made when there is a candidate that scores well above other candidates. If there is no clear best match, then the record is not geocoded. You can, however, attempt to match those records later during interactive geocoding where you can pick the best match from a list of candidates.

In addition, the matching routine uses the geocoding preferences that are set for the table to determine whether certain matching conditions are required or relaxed. For example, MapMarker's default preferences include relaxing a match on street name and ZIP Code, but requiring a match on the house number. This gives the best return of hit rate with the fewest erroneous matches (false positives) and the best performance.

Geocoding Trade-offs

With a relative matching system such as MapMarker, there are trade-offs that must be considered. It is important when using MapMarker that you consider these trade-offs in light of how you will use the geocoded data.

For example, if you have a very large database that covers the entire U.S. and you plan to analyze that data on a national level, then you will likely want as many records to geocode as possible. You are willing to accept some erroneous matches (or geocode those to a less precise point) to ensure that as many records as possible geocode.

If, on the other hand, you will be doing close-in analysis where the location of every record is important, you will want to geocode each record to a street address and minimize the number of erroneous matches (false positives).

Matching preferences and the impact they have on your geocoding session are further discussed in Chapter 4, Making the Most of MapMarker.

Comparing MapMarker with MapInfo's Geocoding Feature

Both MapMarker and MapInfo Professional's geocoding feature assign geographic coordinates and create points for your data so that you can display them in a Map window. Both geocode to street level accuracy, spotting your records to the exact side of the street if possible.

The difference between the two geocoding methods is that MapMarker can do this using a single nationwide matching table (Address Dictionary) that allows you to geocode records anywhere in the U.S. in a single pass. You do not even need to display the Address Dictionary (in fact, you can't).

MapInfo's geocoder uses county StreetInfo files as its search table. Each county level file must be opened and searched individually, so geocoding nationwide in one pass is not possible.

MapMarker is able to differentiate among streets with the same name, thus eliminating the need to use a refining boundary, as necessary when using MapInfo's geocoder.

MapMarker gives you a lot of control with the accuracy with which it makes a match. You can set or relax matching conditions to increase the likelihood of a match. If MapMarker cannot match on street level, it can fall back to ZIP Code centroid in the same pass. When using MapInfo's geocoder, geocoding to street level and ZIP Code centroids must be carried out in separate passes.

Chapter 3: Using MapMarker

Overview

Now that you have been introduced to MapMarker, it's time to get to the business of geocoding your table. This chapter shows you how to run MapMarker. It begins with a discussion on preparing to geocode that will help you get the most out of running MapMarker right from the start.

In This Chapter . . .

Before You Geocode	16
Running MapMarker	18
Identifying Address Columns	19
Geocoding Your Table Automatically	22
Geocoding Your Table Interactively	23
Displaying Geocoded Records in MapInfo.	26
Geocoding to ZIP Code Centroids	27
Geocoding for CASS Certification	27
Geocoding Remote Tables	29
Accessing a Remote Table for Geocoding	34
Additional Considerations for Remote Table Geocoding.	37
Batch Geocoding	38
Geocoding dBASE Files	41
Saving Settings to a Table	41
Setting Geocoding Preferences	42
Setting System Preferences	46

Before You Geocode

MapMarker is simple to use since all you have to do is open your table, tell MapMarker which columns contain address information and choose Table > Geocode. MapMarker automatically matches your records to those in its Address Dictionary, determines if there is a match and assigns the geographic coordinates and optionally creates a point.

Sort Before Geocoding

For fastest operation, we recommend that you sort your table by ZIP Code before running MapMarker. You do not need to scrub your data first since MapMarker's sophisticated matching technology can read and successfully geocode a wide variety of addresses. MapMarker includes an interactive geocoding mode that lets you not only correct your record to make a match, but have the edits permanently written to your base table.

Input Address Columns

MapMarker uses the address information stored in your table and matches it against address records in the Address Dictionary. MapMarker must know what columns contain your addresses. The Select Input Columns dialog is the first step to geocoding where you tell MapMarker that information. For street-level geocoding, you should have at least one street address column plus one column each for city and state or ZIP Code. For ZIP Code Centroid matching, you do not need to specify a street address or city and state.

Output Address Columns

MapMarker can store longitude/latitude coordinates and geocoding result codes in your table. For MapInfo or dBASE tables you can either prepare the table ahead of time by adding columns to store this information, or tell MapMarker to create the columns for you automatically. For ODBC tables, you or your database administrator must add the columns before you run MapMarker.

Note: The longitude/latitude columns are required if you do not want MapMarker to create points automatically. A result code column is also optional, but it is recommended.

If you want MapMarker to return the matching address for you to compare with your input address, you must provide columns for your table before starting MapMarker. While you can have an input address stored in two fields, MapMarker will only return the matching address to a single field. Additionally, in the interactive dialog, you can only edit the address in the first input field.

For geocoding to CASS standards, you are required to provide additional output fields.

Output Column Type/Width Requirements

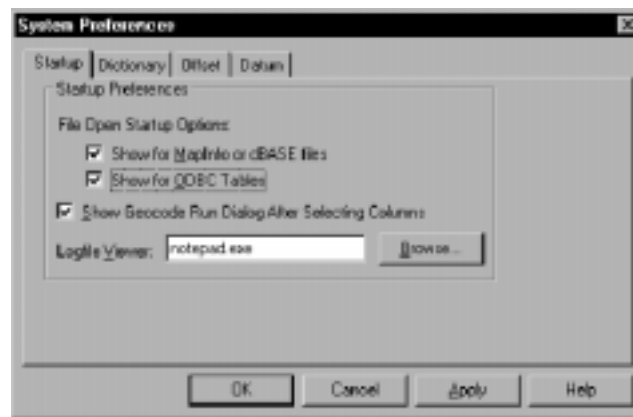
The chart below lists the type and width requirements for output columns.

Output Column	Type(Width)
Longitude	Decimal(11,6) or Float
Latitude	Decimal(11,6) or Float
Result Code	Character(10)
Firm ¹	Character(30)
Address ¹	Character(64)
City ¹	Character(30)
State ¹	Character(2)
ZIP Code ¹	Character(9)
ZIP+4 ¹	Character(4)
Urbanization ¹	Character(30)
Census Block ID	Character(15)
Delivery Point ¹	Character(2)
Check Digit ¹	Character(1)

¹ CASS required field

Setting Startup Preferences

MapMarker provides startup options to get you through the setup phase of geocoding with ease. In the Options > System Preferences dialog, the Startup tab displays as shown below.



You can check any and all items in this dialog. If you check either Show MapInfo/ dBASE files or Show for ODBC tables, the Open dialog displays automatically when you start

MapMarker. If you have both options checked, MapMarker will prompt you with another dialog to choose which type of table you want to open.

Show Geocode Run Dialog After Selecting Columns displays the Geocode dialog from where you begin geocoding once you have selected your columns.

The default settings for the Startup Preferences dialog is to Show File Open for MapInfo or dBASE tables and to Show Geocode Run dialog after selecting columns. Any changes you make here will be saved to the table as metadata and used as new defaults when you open the table again.

Other system preferences are discussed later in this chapter.

Running MapMarker


MapMarker geocodes your records to street level or ZIP Code centroid either automatically (making a match without your involvement) or interactively (allowing you to decide the match from a list of suggestions). In brief, the geocoding process follows these steps:

1. Open the table to be geocoded and identify the columns that contain addresses.
2. Geocode your records automatically to match as many records as possible.
3. Geocode remaining unmatched records interactively by choosing to accept or ignore proposed matches.
4. Display your newly geocoded records as point objects in MapInfo.

Identifying Address Columns

Summary: Open your table and tell MapMarker which columns contain the address information MapMarker will use to attempt a match.

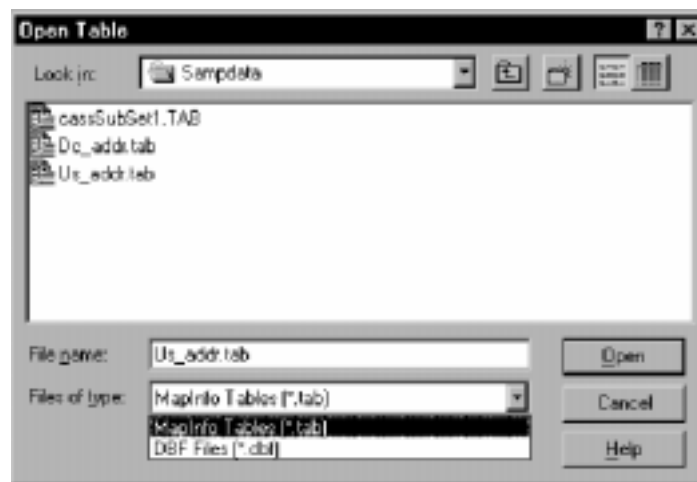
Note: This procedure applies to opening MapInfo and .dbf files. For instructions on opening remote tables, see Geocoding Remote Tables, later in this chapter.

1. Start MapMarker from the Windows 95 Start menu or Shortcut menu. For Windows NT users, double-click on the icon from Program manager 

If MapMarker is already running, choose File > Open Table, click




or click the secondary mouse button to display a pop-up menu. The Open dialog displays.



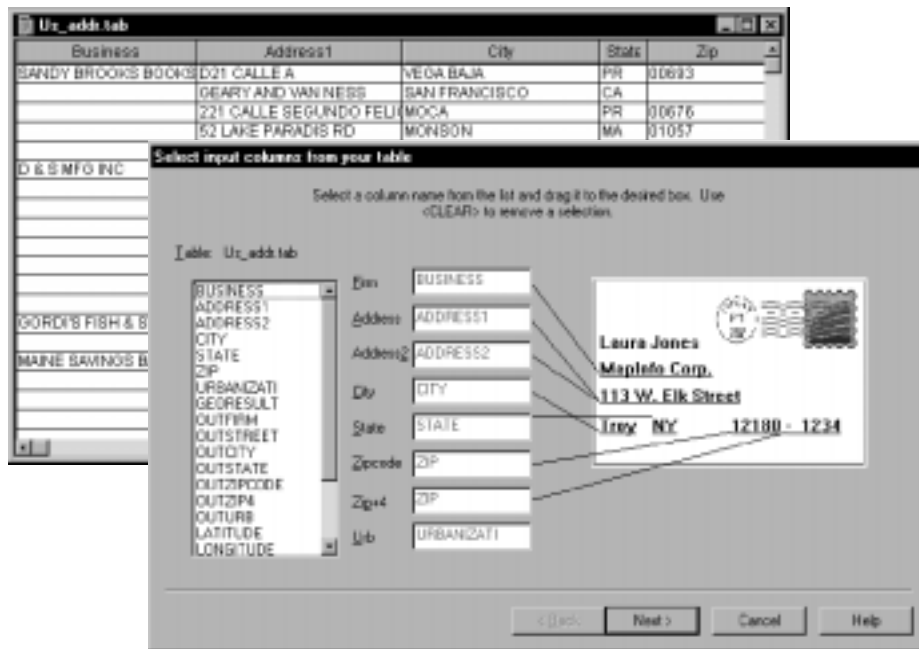
2. Choose the .tab or .dbf table you wish to geocode. Click Open.

If your table does not contain a result column, the Add Columns dialog displays telling you it will be created for you. If you wish to store longitude and latitude coordinates in your table and you do not have those columns, MapMarker will automatically add them as well.

 **Note** The result column is recommended. Coordinate columns are required only when you do not want points created automatically.



3. Click OK. A progress bar indicates your table is being updated. Following that, a Browser window of your table and the Select Input Columns dialog displays.



4. Choose the columns that contain the appropriate address components.

If your table has separate columns for ZIP Code and ZIP+4 information, identify each column in the ZIP/ZIP+4 and Plus4 drop-down lists. If your table has ZIP Code and ZIP+4 extensions in the same column, specify the column in both boxes.

You will notice that MapMarker attempts to choose the address columns for you. It can make some determinations for fields, but if this is the first time this table has been opened in MapMarker, be sure to check these columns. Once you have geocoded the

table, the column settings will be saved to the table as metadata and used for future geocoding sessions.

- Click the Next button. The Select Output Columns dialog displays.

- Identify columns in your table where MapMarker will store the result codes, coordinate information and output addresses from the Address Dictionary.

If you do not want MapMarker to create points for your table, clear the checkbox marked Create points in MapInfo table. You must choose longitude and latitude columns in order to proceed with geocoding.

Specify the columns to contain output information from the Address Dictionary for each matched record. If you are geocoding for CASS certification, all fields in this group with the exception of Census are required.

Warning Use care when choosing the same columns for output and input addresses. MapMarker will overwrite your data with the output address.

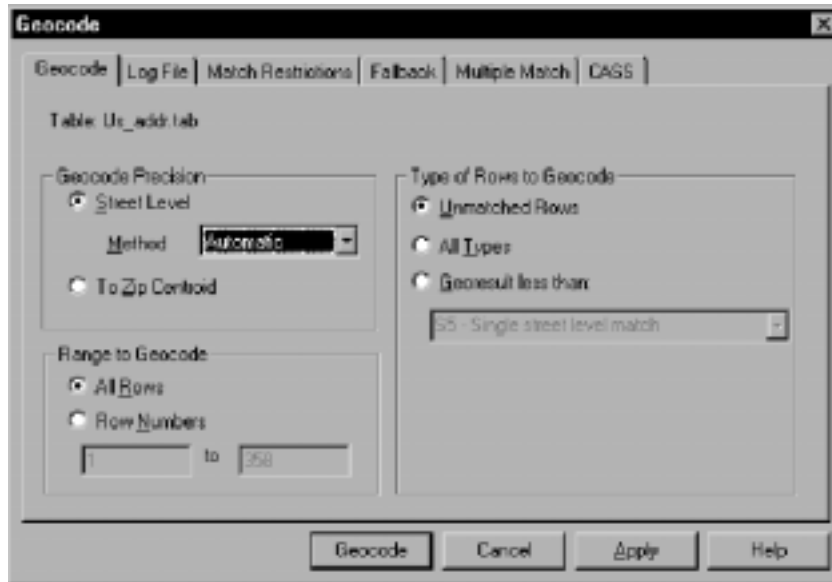
MapMarker only returns the parsed street name to a single field.


Identify a census column if you want MapMarker to return the Census Block code for the matched address. MapMarker returns the code (up to 15 characters) that represents the state and county FIPS code, census tract and tabulation block.

- Once you have identified all your columns, click Finish to leave this dialog. You are returned to the MapMarker menu or to the Geocode dialog automatically.

Geocoding Your Table Automatically

Summary: MapMarker runs automatically to match as many records as possible.




1. If the Geocode dialog does not automatically display on leaving the Select Output dialog, choose Table > Geocode from the MapMarker menu, click  or click the secondary mouse button to display a pop-up menu from which to choose Geocode.
2. Set up the geocoding parameters here. Choose the appropriate geocoding precision: street level or ZIP Code centroid. Tell MapMarker how much of the table to geocode. By default, MapMarker automatically geocodes the unmatched rows. For previously geocoded tables, you can also try to improve a match for a record by geocoding rows that have result codes below the desired level.
3. Set additional match settings by checking the appropriate boxes in the other Geocode dialogs. For an explanation of these settings, see Setting Geocoding Preferences later in this chapter.
4. Click Geocode to begin geocoding. The Automatic Geocode dialog displays. A progress bar updates the match count every 25 records.
5. At completion of the geocoding run, the Cancel button changes to Done. Click Done to leave the summary dialog. MapMarker prompts you to view the Log file if you checked Update Log file in the Log File dialog.

The Log file contains the results of the geocoding session, including the number of matched, unmatched and previously matched records, and the number of matched records by result code.

Information from subsequent passes are appended to the top of the Log file. The Log file size limit is 1 MB.


- To view the Summary dialog again, choose View > Summary. Note: only the summary from the most recent geocoding pass for the active table is available from this menu.

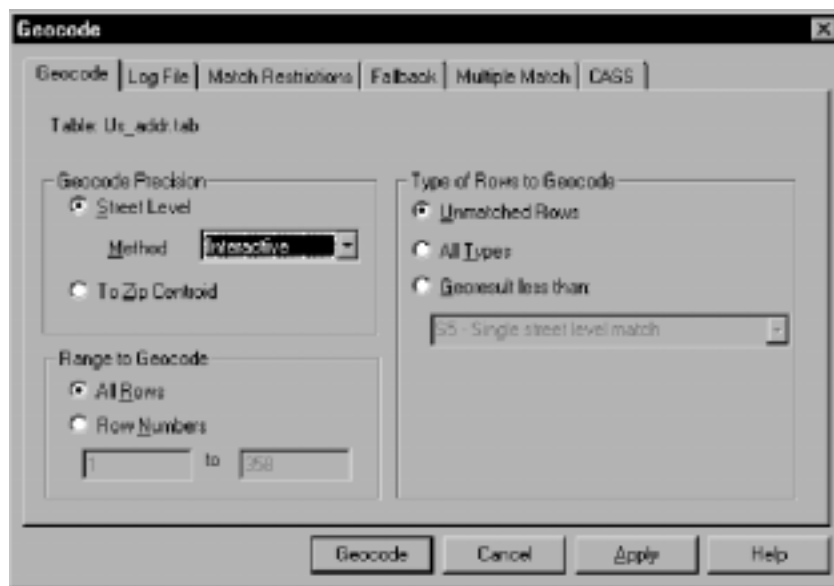
 **Note** the summary is for the current table (the frontmost browser). If you run multiple tables in a single session, to get their summary information, make the table active by clicking on the Browser title bar and choose View > Summary.

To view information about previous geocoding passes, view the Log file by choosing View > Log file.

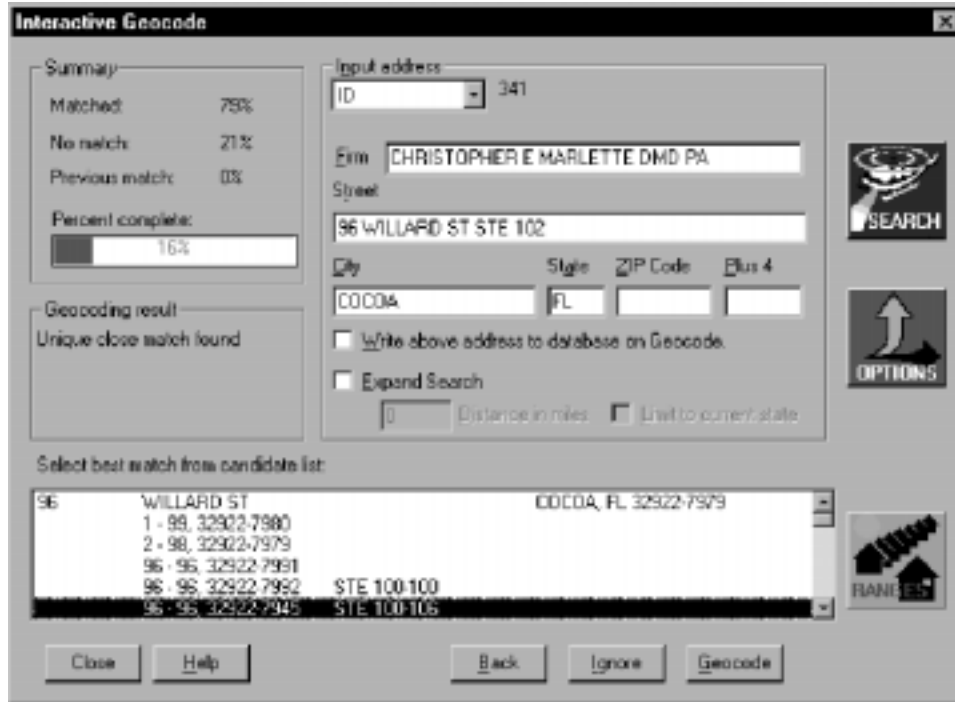
Geocoding Your Table Interactively

Summary: Geocode records interactively by choosing to accept or ignore proposed matches. Interactive geocoding also allows you to expand the search for a match, move back up the table to an earlier record, and correct your records directly from the MapMarker dialog.

- Choose Table > Geocode, click , or click the secondary mouse button to display a pop-up menu from which to choose Geocode. The Geocode dialog displays.



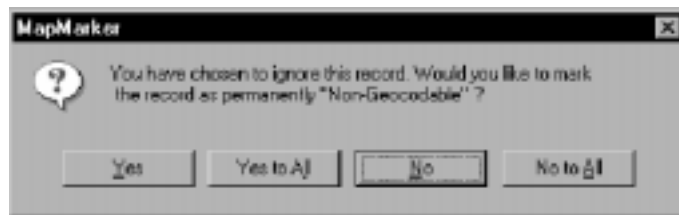
- In the Geocode Precision group, choose street level and the interactive method.
- Choose the range and type of rows to geocode.
- Click Geocode. The Interactive Geocode dialog displays showing the first unmatched record and a list of suggested matches.



The input address shows the row number for the record by default. If you wish to display another field from the input table choose it from the drop-down list.

5. Choose the Geocode button if you agree that the highlighted suggested match is the correct match. MapMarker assigns coordinates from that row and moves on to display the next unmatched record. To choose a different candidate in the list, move the highlight with your cursor to your choice and choose Geocode.
6. Choose Ignore if MapMarker offers no match suggestions or if you do not agree with the choices in the match list.

When you click Ignore, MapMarker displays a dialog asking if you want to mark the record as “Non-geocodable,” meaning that you do not want MapMarker to attempt to match this record during future geocoding passes. This is useful when you know the record will not geocode and you do not want it to display again in the interactive dialog. You have several choices in this dialog:



Choose **Yes** to have MapMarker mark the record with the result code NG (non-geocodable). The dialog will display each time you click Ignore.

Choose **Yes to All** if you want to mark all future “Ignore” records in this geocoding session as NG. The dialog will no longer display.

Choose **No** if you do not wish to mark the record as NG. The dialog will display each time you click Ignore on a record.

Choose **No to All** if you do not want to mark any “Ignore” records as NG. The dialog will not display again during the session.

7. To expand the search area, check Expand Search and specify a radius up to 99 miles. Check the box Limit to current state if desired. This feature is helpful to find a match when the input address contains limited or inaccurate city or ZIP Code information.
8. To change the matching conditions for a record, click the Options button. The tab dialogs for Match Restrictions, Fallback and Multiple Match display. Make the necessary changes to increase the potential for a match. Click Search to search the Address Dictionary using the new matching conditions.
9. Edit the record and choose Search to bring up a new list of suggestions that may better match the edited record. Edit the fields by highlighting the text in the input address and typing new text to correct the address.

Additionally, incorporate the edits into your table by checking the **Write above address to database on Geocode** box. Choose the match from the list and click on Geocode. MapMarker geocodes the record and updates your table with your edits.

10. To see an expanded list of ranges, including apartment numbers, office suites, etc., choose Ranges. A detailed list of address ranges for the highlighted street displays. Click on the Ranges button again to close the list of ranges.
11. Choose the Back button to move backward in the table to redisplay a record.
12. Click Close to leave the Interactive dialog. A Geocoding Summary displays the results of the session.

	Rows	Percent
Matched	135	82.8%
No Match	28	17.2%
Previous Match	0	0.0%
<hr/>		
Total	163	29.6%

Done

13. Click Done. To view the summary again, choose View > Summary.

Note Interactive geocoding is not completely interactive. MapMarker does attempt to geocode the row first. Only when it fails to find a single close match is the record displayed in the Interactive dialog.

Displaying Geocoded Records in MapInfo

Summary: After MapMarker geocodes your records, display them in MapInfo Professional to analyze them geographically.

1. Close your table in MapMarker and start MapInfo Professional. The Quick Start dialog displays.
2. Choose Open Table and choose the table you just geocoded. Click OK. Your table displays its records as point objects in a Map window

The points display with color-coded symbols that represent the level of geocoding precision for the record, as described below.



- **Green stars:** all records geocoded to a street address or intersection (S5, M5, SX and MX result codes)
 - **Blue stars:** all records geocoded to a shape path centroid (S4, M4)
 - **Yellow stars:** all records geocoded to a ZIP+4 centroid (S3, M3, Z3)
 - **Cyan stars:** all records geocoded to a ZIP+2 centroid (S2, M2, Z2)
 - **Red Stars:** all records geocoded to a ZIP Code centroid (S1, M1, Z1)
3. Add other map layers such as StreetWorks or ZIP Code boundaries to give your points a geographic reference.

You are now ready to analyze your data geographically using the power of MapInfo.

- Note** The above steps assume that the point objects were created automatically during geocoding. If MapMarker does not create points for your table, you must create them separately in MapInfo, using Table > Create Points, before you can display your records as point objects in a Map window. For more on creating points in MapInfo, see the MapInfo *Reference*.

Geocoding to ZIP Code Centroids

As an alternative to street-level geocoding, you can tell MapMarker to geocode your records to the appropriate ZIP Code centroid for the record. This is a fast method of geocoding because it eliminates the need to match based on street address. This method matches the ZIP Code in your table with the ZIP Code in the Address Dictionary. MapMarker geocodes to the most precise ZIP Code level contained in your table.

Use ZIP Code Centroid matching when you have a large database that you want to geocode quickly and you are less concerned with spotting the points to street level.

To geocode to ZIP Code centroids:

1. At the Geocode dialog, choose To ZIP Centroid under Geocode Precision.
2. Choose the range and type of rows you want to geocode.
3. Click Geocode. MapMarker updates the Automatic Geocode dialog every 25 records.
4. When through, click Done to view the Log file or return to the MapMarker menu.

Effective with version 3.4, you may set the option to fallback to ZIP Code Centroids if MapMarker does not find a match.

Geocoding for CASS Certification

CASS certification is a process by which a table of mailing addresses is standardized to meet U.S. Postal Service requirements for bulk mailing discounts. MapMarker performs this address standardization while it geocodes your records for displaying on a map.

Note: you cannot geocode your records to a user dictionary during CASS geocoding.

To geocode your data for CASS certification:

1. Add the following output columns to your source table. You can name the columns to any name you desire. Make sure the type and length are as follows:
 - URB - Character(30) - This field is will store urbanization codes for Puerto Rican addresses. You must have this field even if you do not have Puerto Rican addresses.
 - Firm - Character(30)
 - Zip - Character(5)
 - Zip+4 - Character(4)
 - Address - Character(64)
 - City - Character(30)

State - Character(2)

Latitude - Float OR Decimal(11,6)

Longitude - Float OR Decimal(11,6)

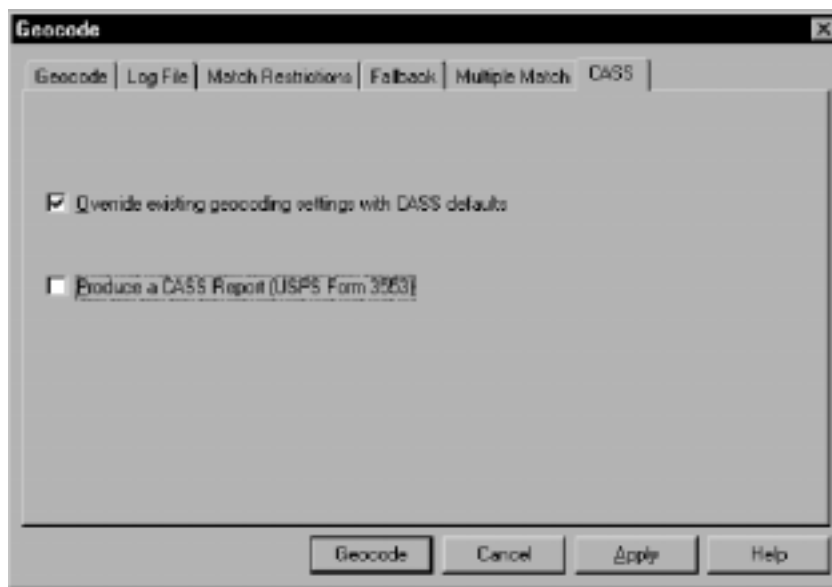
Result - Character(10)

Delivery Pt - Character(2)

Check Dg - Character(1)

Census - Character(15) (this field is not required for CASS geocoding)

2. Start MapMarker.
3. Open the table you want to geocode.
4. Choose Table > Geocode and click the CASS tab. Check Override existing geocoding settings with CASS defaults. MapMarker will use these defaults to geocode your data to CASS specifications.



5. To produce US Postal Service Form 3553 CASS Report check the box Produce a CASS Report. You will be prompted for a name and location for the report at the Save CASS Report As dialog.
6. When you are returned to the CASS tab dialog, click Geocode to begin the operation.

MapMarker geocodes your data to CASS standards and provides a printed report of the address standardization operation (optional).

Geocoding Remote Tables

Remote Databases and Spatial Objects

Effective with versions 3.3 and 3.4, MapMarker includes expanded support for geocoding remote databases via ODBC. MapMarker now supports two types of remote mappable tables: those on relational database management systems (RDBMS) that use the XY spatial index, and SpatialWare tables that use the SW_GEOMETRY. Both types of tables can store the geocoded spatial object directly.

Remote table geocoding was introduced in MapMarker 3.0. At that time, only the geographic coordinates of a successfully geocoded record were stored in the table. Now MapMarker stores the spatial object for that record as well. As a significant difference, this allows you to update and map the data points while maintaining its integrity on your secure server.

The following section introduces you to mappable tables and the MapInfo Map Catalog, features that are an integral part of remote geocoding with MapMarker.

Mappable Tables and the MapInfo Map Catalog

Remote mappable tables and the MapInfo Map Catalog were first introduced in MapInfo Professional as a way to communicate spatial information about remote data to the MapInfo product line. A mappable table is a table of data that has its spatial information recorded in the MapInfo Map Catalog. In MapInfo mapping software, a mappable table displays its spatial data in a map to allow better visualization and analysis of that information.

Mappable tables are managed by the MapInfo Map Catalog, a special table that contains information from the mappable table about how to spatial data is stored. MapMarker reads the Map Catalog each time you geocode a remote mappable table in order to know where to put the coordinate values and how to handle the spatial object. One Map Catalog is required for each remote database you want to access in MapInfo or MapMarker. In every Map Catalog an entry is required for each table in the database that you wish to geocode.

We recommend that you create your Map Catalog and make a table mappable within MapInfo Professional. Specific tools and commands are provided there for this purpose.

If, however, you do not have access to MapInfo Professional, this chapter contains instructions for carrying out these two procedures by hand.

Supported Spatial Index Types

MapMarker currently supports two spatial types: SW_GEOMETRY (for SpatialWare on Oracle) and X,Y spatial index (for Oracle, Informix, Sybase, SQL Server and MS Access). They are represented in the Map Catalog under the column SPATIALTYPE with the value 5.0 for SpatialWare and 4.0 for X,Y. Additional spatial type support is under development.

If MapMarker encounters a mappable table with a spatial type it does not support, it will display an error message and not geocode the table.

Preparing to Geocode Remote Databases

Before you can geocode your remote table with MapMarker, you and/or the database administrator must set up MapMarker and your database to handle the geocoding operation.

In general, the process follows these steps:

1. Install MapMarker ODBC support.
2. Prepare the remote table with the appropriate columns and spatial handling requirements.
3. Create a Map Catalog for each remote database.
4. Make the remote table mappable.

Each step is explained below:

Step 1: Installing MapMarker ODBC support

On the MapMarker side, at installation, choose the Remote Data Access (ODBC) component and follow the installation process. At the MapInfo ODBC Driver Installer dialog, choose the ODBC drivers to install. Click Next. The installation proceeds to completion.

After the drivers are installed, create a data source for your database. Use the ODBC Administrator tool in Control Panel for this purpose.

Step 2: Preparing the Remote Table

You or your database administrator must add columns to the table that will store the results of the geocoding operation. For the X,Y spatial index method, these columns include one for longitude and one for latitude values. Each should be of decimal (11,6 or larger) or double. Refer to the section Output Column Type/Width Requirements in the MapMarker Product Guide earlier in this chapter.

In addition, you may wish to add a result column to your table for storing the result code that MapMarker generates to describe the type of geocoding match that was attempted.

For SpatialWare tables on Oracle, the spatial information is stored in a column called SW_GEOMETRY instead of separate longitude/latitude columns. Refer to your database and SpatialWare documentation for more information (Note: SpatialWare is sold separately from MapMarker).

Step 3: Creating a Map Catalog

Create a Map Catalog for each remote database you want to access in MapMarker. To do this, in MapInfo Professional, run the utility MIODBCAT.MBX from the Run MapBasic Program command. For complete details see the MapInfo Professional *User's Guide*, Chapter 19: Accessing Remote Database Data.

Manually Creating a MapInfo Map Catalog

If you are not a MapInfo Professional user, you will need to create the Map Catalog (or have your database administrator create it) manually, as described below. This process only needs to be performed once for each database.

1. If the RDBMS requires owners and users, create the user MAPINFO with the password MAPINFO in the specific database where the mappable tables are located.
2. Create the table MAPINFO_MAPCATALOG in the database. The Create Table statement needs to be equivalent to the following SQL Create Table statement:

```

Create Table MAPINFO_MAPCATALOG (
    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)
)

```

Note: It is important that the structure of the table is exactly like this statement. The only substitution that can be made is for databases that support varchar or text data types; these data types can be substituted for the Char data type.

3. Create a unique index on the TABLENAME and OWNERNAME. For RDBMS's that do not support owners, a unique table name is required. For other database systems, the Map Catalog requires a unique owner.tablename.

- Grant Select privileges on the MAPINFO_MAPCATALOG. This allows users to make tables mappable. The Update, Insert and Delete privileges should be granted at the discretion of the database administrator.

SPATIALTYPE	TABLENAME	OWNERNAME	SPATIALCOLUMN	DB_X_LL	DB_Y_LL	DB_Z_LL	DB_M_LL	COORDINATESYSTEM
4	ACCESS0u	Admin	NO_COLUMN	-360	-90	360	90	EARTH PROJECTION 1
4	ACCESS0u27	OWNER	NO_COLUMN	-360	-90	360	90	Earth Projection 1,0
4	Customer	David	NO_COLUMN	-360	-90	360	90	Earth Projection 1,0
4	Orders	Fran	NO_COLUMN	-360	-90	360	90	Earth Projection 1,62

Step 4: Making a Remote Table Mappable

For each spatial table in the remote database that you want to access in MapMarker, you must add a row to the MAPINFO_MAPCATALOG table. This is carried out in MapInfo Professional when you choose Table > Maintenance > Make ODBC Table Mappable. For details, see the section Making an ODBC Table Mappable in the MapInfo Professional *User's Guide*, Chapter 19.

If you do not use MapInfo Professional to manage the Map Catalog, you must manually add rows to the MAPINFO_MAPCATALOG table for each spatial table in the database that you want to geocode. Each entry must contain the following information about the table.


Column Name	Values to Assign	Example
SPATIALTYPE	5.0 for SpatialWare tables 4.0 for X,Y spatial index tables (Support for additional spatial servers is under development)	5.0
TABLENAME	Name of the table.	Drainage
OWNERNAME	Owner name.	Georgetown
SPATIALCOLUMN	Name of the column, if any containing spatial features. Will be either: SW_GEOMETRY (for mappable SpatialWare tables) or NO_COLUMN (for mappable tables using X,Y)	SW_GEOMETRY

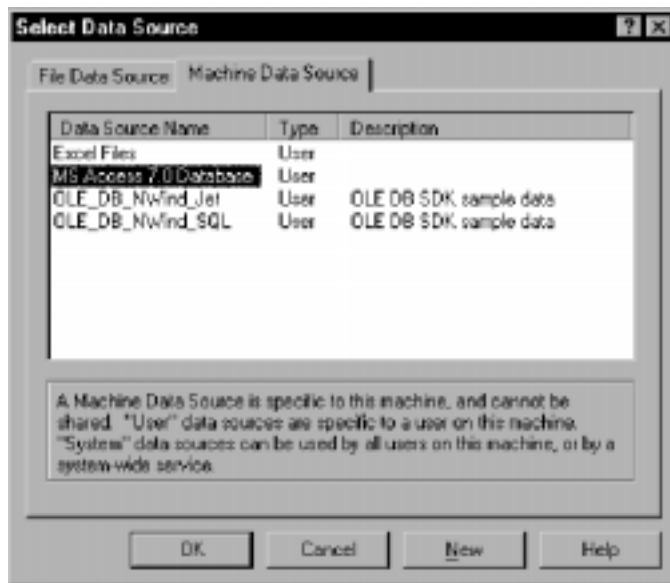
Column Name	Values to Assign	Example
DB_X_LL	X coordinate of the lower left corner of the layer's bounding rectangle, in units indicated by the COORDINATESYSTEM as defined by MapInfo Professional (see below).	-360
DB_Y_LL	Lower left bounding Y value.	-90
DB_X_UR	Upper right bounding X value.	360
DB_Y_UR	Upper right bounding Y value.	90
COORDINATESYSTEM	A string representing a MapInfo-supported coordinate system which specifies a map projection, coordinate units, etc. Will be one of: Earth Projection 1,0 (for NAD27) Earth Projection 1,62 (for NAD27) Earth Projection 1,33 (for NAD 83) or Earth Projection 1,74 (for NAD 83)	Earth Projection 1,0
SYMBOL	A MapInfo Symbol clause (for a layer containing points)	Symbol1 (35,0,12)
XCOLUMNNAME	Specify the name of the column containing X coordinates. For SpatialWare tables that use a single column, specify NO_COLUMN.	NO_COLUMN
YCOLUMNNAME	Specify the name of the column containing Y coordinates. For SpatialWare tables that use a single column, specify NO_COLUMN.	NO_COLUMN

Accessing a Remote Table for Geocoding

Now that your remote database is prepared for geocoding with MapMarker, you are ready to open it in MapMarker and begin geocoding. Follow the steps below.

1. If you have specified to show File Open dialog at Startup for ODBC tables (by setting the System Preferences), when you start MapMarker, the Select Data Source dialog displays automatically.

Otherwise, choose File > Open ODBC Table or click  to display the Select Data Source dialog.



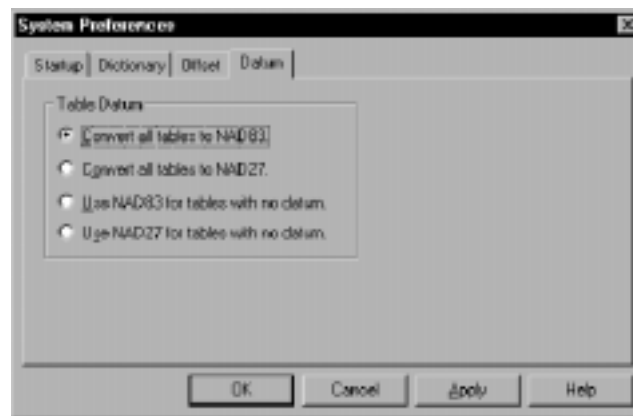
2. From the Select Data Source dialog, click the Machine Data Source tab.
3. Choose your data source. Your data source is the location of the specified database. For example, CUSTOMER could be the name of the data source that provides access to one or more ORACLE tables located in a specified directory. Once connected to the CUSTOMER data source, you could access information from any Oracle table.
4. Sign on to your data source in the usual way. This procedure differs depending on what type of ODBC table you are using.



5. Choose the table you want to access from the Select Table dialog. Each of the tables owned by you or accessible to you will be displayed. The table displays. Keep in mind you cannot open any other tables while you have an ODBC table open.
6. Proceed with geocoding as you would with any other MapInfo table. The procedure is the same as geocoding local tables, with two exceptions, as explained below.

Setting the Datum for Remote Tables

For remote table geocoding, MapMarker only geocodes to a NAD 27 or NAD 83 datum. If your remote table uses another datum, MapMarker will prompt you to choose NAD 27 or NAD 83 before proceeding.



In the Datum tab of the System Preferences dialog, the options “Use NAD83 for tables with no datum” and “Use NAD 27 for tables with no datum” act the same as if you chose one of the first two options, “Convert all tables to NAD 83” or “Convert all tables to NAD 27.”

Be sure that you use the same datum for the entire table to ensure consistency among geocoded records.

For more on datums, see the section Understanding Datums in Chapter 4.

Identifying Output Columns for Remote Tables

The Coordinate Column group in the Select Output Columns dialog behaves differently for mappable remote tables compared to non-mappable remote tables or local tables.



MapMarker reads the Map Catalog to determine if an entry for the remote table exists, and therefore, is mappable. If so, the checkbox labeled Create points in MapInfo Table will be checked and grayed out, as illustrated above. You cannot turn off the create points feature for a mappable table.

MapMarker leaves the Long(X) and Lat (Y) output columns boxes empty because it already knows from the Map Catalog where to output the coordinate values. Unlike the rest of the output columns for the table, you cannot drag and drop a column name into the Long and Lat boxes. In fact, MapMarker does not display your coordinate output column names in the list for remote mappable tables.

For remote mappable tables, MapMarker will update the appropriate coordinate columns and create points for the table without interaction from you. Once you set your other output columns, you are ready to geocode.

For non-mappable remote tables, you must either accept the column choices MapMarker displays (based on intelligent guesses of typical coordinate column names) or drag and drop the coordinate column names from the list to the Long (X) and Lat (Y) boxes. The Create points in MapInfo table checkbox will be cleared and grayed out. MapMarker will only update the coordinate columns, not store the spatial object in the table. For that the table must be mappable.

Additional Considerations for Remote Table Geocoding

Transaction Control with ODBC Tables

MapMarker uses as non-obtrusive locking mechanism as possible when geocoding your ODBC table. It attempts to use the lowest level of locking available (e.g., row, page or, at worst case, table locking). Locks are not placed until the row is to be geocoded. MapMarker issues a commit after each update so that each lock is held for an extremely short time (the time it takes to lock a row, update it and commit it).

If MapMarker attempts to lock a row that is already locked, MapMarker will follow the default behavior of the database. Usually this means waiting until that row is released. MapMarker does not use any deadlock detection beyond that provided by the database.

IMPORTANT: You are responsible for maintaining the relationship of the street address and the latitude/longitude for that address. For example, whenever you change any street address information, you should regeocode that record. For Informix databases, MapInfo provides a MapInfo Geocoding Datablade based on MapMarker technology that will automatically geocode rows as they are inserted or updated. An equivalent Oracle Data Cartridge is also planned.

Finally, if rows are deleted from your table after MapMarker opens the table, MapMarker will report an error when it tries to geocode that deleted row.

Geocoding Sybase Databases

In the Performance tab of the ODBC Sybase Driver Setup dialog, be sure to set the Prepare Method to 2 - Full and the Select Method to 1 - Direct. The default settings of None and Cursor may cause locking if there concurrent users accessing the database.

Geocoding Preferences

You cannot geocode a subset of a remote database unless you make that subset a new table. At the Geocode dialog, in the Range to Geocode group, the Row Numbers option button is grayed out.

View Tables

MapMarker is not able to geocode view tables.

Rollback Segment Limit

Be sure that you set your database rollback segment or temp space to be sufficiently large enough to accommodate the size of your database.

Supported Unique Index and Primary Key Data Types by Database

For the latest information on supported data types, see the document, "MapMarker Supported Data Types" on the MapMarker home page (<http://testdrive.mapinfo.com> and go to the MapMarker area).

MapMarker 3.x supports ODBC tables in MS Access7, Informix7, Oracle7, SQL Server6 and Sybase 10.02 and 11.02 formats and SpatialWare 2.2 on Oracle. If you will be geocoding your ODBC table, you or your database administrator must first add the fields for storing result codes and latitude and longitude coordinates. A Primary Key must also be defined for your table.

Batch Geocoding

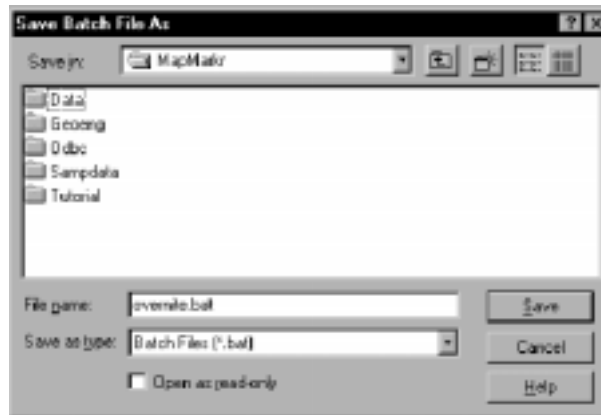
Effective with version 3.3, MapMarker supports batch geocoding for processing one or more tables without constant user interaction. You can now set up a text file containing the path and file names of the tables to geocode and specify the geocoding preferences you want applied to the session.

Batch geocoding is useful for overnight processing of large databases or when a table is updated regularly via a scheduling program.

You may create the batch file from the MapMarker interface by opening a table and selecting the geocoding preferences.

To create a batch file:

1. Open a table in MapMarker and proceed through the select column dialogs to confirm the input and output columns.
2. At the Geocode dialog, choose the settings you want for the geocoding pass. Choose from the Geocode, Log File, Fallback and Multiple Match tabs. You cannot choose CASS settings for batch geocoding. Click the dialog Help buttons to learn more about each geocoding option.
3. When through with the settings, choose Table > Create Batch File. The Save Batch File As dialog displays.



4. Specify the path and filename for the batch file. By default the batch file will use the root file name of the table with the extension .bat. Choose Save.
5. Repeat steps 1-4 for each additional table you want to include in the batch file. When saving it to the batch file, specify the same name you created in step 4. To create a new batch file, use a new filename.
6. To run the batch file, double-click on the filename in Explorer. MapMarker will start running and open each table listed in the batch file in turn and geocode it to the specified settings. If MapMarker encounters an error, it will stop.

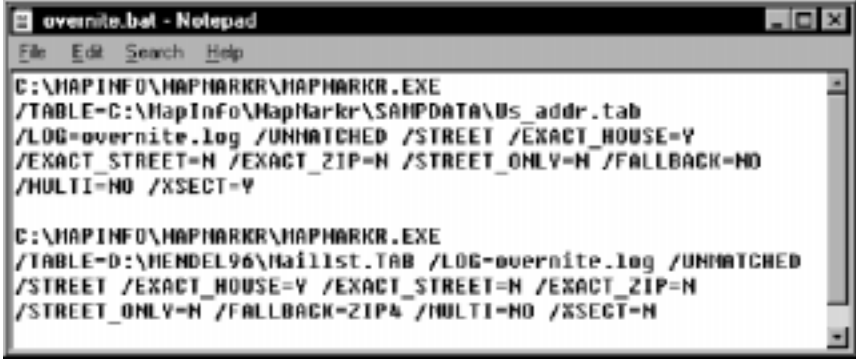
You may also create the batch file by hand, if you prefer. In a text editor, list the path and filename for the table and any geocoding parameters you want MapMarker to follow. The list of parameters is shown below. In order to activate the parameters, you should have opened the table in MapMarker at least once and set the preferences. Otherwise MapMarker will geocode to the default settings for that table as found in the metadata.

The command line parameters are as follows:

Parameter	Description
/TABLE	Path and filename of table to geocode.
/LOG=<logfile name>	Name of log file. MapMarker will write the geocoding results to this file. If a log file is not specified, MapMarker will create a default Mapmarkr.log file.

<code>/STREET</code> or <code>/ZIP</code>	Specify whether MapMarker geocodes to streets or to ZIP Code centroids.
<code>/START=<number></code>	Specify the starting row number to geocode. If omitted, MapMarker will begin with row 1.
<code>/END=<number></code>	Specify the ending row number to geocode. If omitted, MapMarker will end at the end of the table.
<code>/ALL</code> or <code>/UNMATCHED</code>	Specify whether to geocode all records or only unmatched records.
<code>/EXACT_HOUSE=Y,N</code>	Specify yes or no to geocode to the exact house number.
<code>/EXACT_STREET=Y,N</code>	Specify yes or no to geocode to the exact street name.
<code>/EXACT_ZIP=Y,N</code>	Specify yes or no to geocode to the exact ZIP Code.
<code>/XSECT=Y,N</code>	Specify yes or no to geocode to street intersections.
<code>/STREET_ONLY=Y,N</code>	Specify yes or no to geocode only to street addresses, not to ZIP Codes.
<code>/FALLBACK=NO,ZIP4,ZIP2,ZIP</code>	Specify whether MapMarker will fallback to a ZIP Code, and if so, to which precision: ZIP+4, ZIP+2 or ZIP Code.
<code>/MULTI=NO,FIRST,STREET,ZIP</code>	Specify how MapMarker will handle multiple matches: No = do not geocode First = geocode to the first multiple match Street = choose a street match over a ZIP Code match ZIP = choose a ZIP Code match over a street match.

The illustration below shows two tables and the geocoding instructions for MapMarker.



```
overnite.bat - Notepad
File Edit Search Help
C:\MAPINFO\MAPHARKR\MAPHARKR.EXE
/TABLE=C:\MapInfo\MapMarker\SAMPDATA\Us_addr.tab
/LOG=overnite.log /UNMATCHED /STREET /EXACT_HOUSE=Y
/EXACT_STREET=N /EXACT_ZIP=N /STREET_ONLY=N /FALLBACK=NO
/MULTI=NO /XSECT=Y

C:\MAPINFO\MAPHARKR\MAPHARKR.EXE
/TABLE=D:\MENDEL96\mailist.TAB /LOG=overnite.log /UNMATCHED
/STREET /EXACT_HOUSE=Y /EXACT_STREET=N /EXACT_ZIP=N
/STREET_ONLY=N /FALLBACK=ZIP4 /MULTI=NO /XSECT=N
```

Batch Geocoding Considerations

If you list a table in the batch file without parameters, MapMarker will geocode to the default settings for the table (whatever the settings were the last time the table was opened in MapMarker).

MapMarker's batch geocoder uses the system preferences that are set at the time of geocoding. To ensure that a particular system setting will be used, such as the location of the Address Dictionary, in the System Preferences dialog choose the setting, apply it, make the batch file and run the batch file during the same MapMarker session.

MapMarker uses the input and output columns it finds in the table's metadata. Columns cannot be changed for batch geocoding.

CASS geocoding is not supported in MapMarker's 3.x batch geocoder.

For a hand-generated batch file, you may use uppercase, lowercase or mixed case text. The parameters can be listed in any order.

Geocoding dBASE Files

MapMarker will read dBASE format (.dbf) files as well as MapInfo native format (.tab), thus enabling you to turn any table of records into geographic objects. This section explains some things to keep in mind when geocoding .dbf files.

MapMarker creates a .tab file automatically when you open a .dbf file. If you open the .dbf a second time, MapMarker tells you that a .tab file already exists. Choose Yes to replace it.

Since MapMarker automatically creates a .tab file for any .dbf file you open, all of the options available to .tab files apply to .dbf files. Among them, MapMarker adds coordinate and result columns and creates points for your table automatically.

MapMarker does not display memo fields included with .dbf files. The fields are not changed in any way by MapMarker.

MapMarker does not maintain indexes for .dbf files. If you write to an indexed field during geocoding, the table must be re-indexed outside of MapMarker if you wish to maintain the index.

Saving Settings to a Table

MapMarker saves the settings you choose for geocoding and stores them as metadata in your table. These settings include input and output columns, geocoding preferences such as match restrictions, fallback settings and CASS options, and system preferences for startup, datum

and address dictionaries. The next time you open the table in MapMarker, you can proceed directly to geocoding. To view the settings, open the .TAB file in a text editor and look under the section called “Metadata.”

Setting Geocoding Preferences

This section discusses geocoding matching options that are available from the tabs in the Geocode dialog. These preferences affect the conditions under which MapMarker attempts to match a record.

Changing match settings can affect the hit rate of successfully geocoded records. It can also affect the time in which MapMarker takes to geocode a table, as well as the precision with which the geocoded record will display on a map. It is important to keep in mind your intended use of geocoded data when setting these conditions. (The impact of changing geocoding conditions is explored further in the next chapter.)

Choose to accept or change any of the matching conditions to fit your needs. By relaxing some parameters, MapMarker is, in essence, expanding the search area or criteria used to match the record. Each condition is explained below.

Geocode Preferences

Geocoding preferences include all the settings that affect how MapMarker makes a match. This includes automatic or interactive geocoding, exact or relaxed matching settings, fallback options, how to handle multiple matches, Log file setup and CASS certification settings. MapMarker allows you to control every aspect of the geocoding operation.

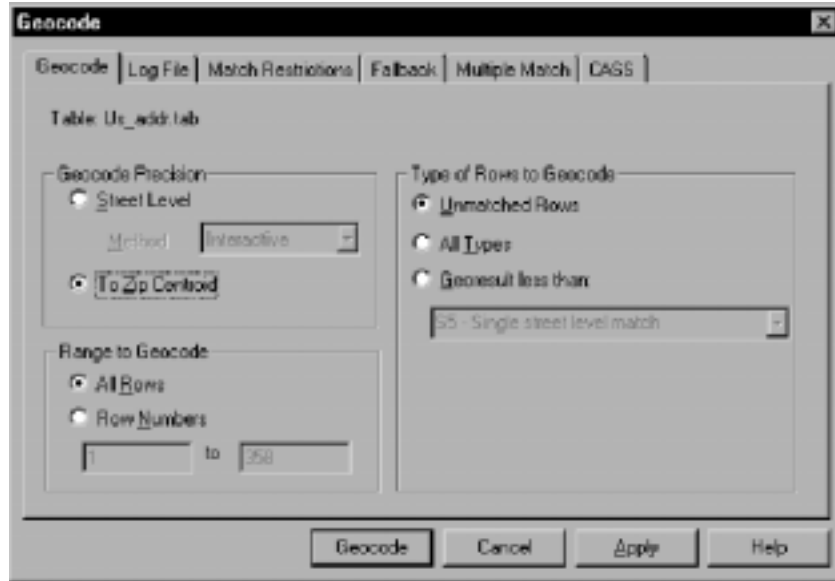
To set geocoding preferences:

1. Choose the settings you want from any of the six geocoding preferences dialogs: Geocode, Log File, Match Restrictions, Fallback, Multiple Match, or CASS.
2. Click Apply to set the changes or Geocode to proceed directly to processing.

Each of the preferences is explained below:

The geocoding precision is how close a match is to the true location of the record you want to geocode. MapMarker provides for two types of precision: **Street Level** and **ZIP Centroid**.

Street Level precision means that MapMarker will attempt to geocode all records to street address, but in all likelihood, some matches may end up at a less precise location such as a ZIP centroid (ZIP+4, ZIP+2, or ZIP Code) or shape path.



ZIP Centroid precision means that MapMarker will attempt to match a record to the most precise ZIP Code it finds. Keep in mind that a match to a ZIP+4 centroid is nearly as precise as a match to a street address. The added bonus of ZIP Centroid matching is the speed of the operation; MapMarker processes records quicker if it only has to match on the ZIP Code.

The disadvantage of ZIP Code matching is that MapMarker only examines the ZIP Code field. If your records only contain 5-digit ZIP Codes, MapMarker can do no better than match to 5-digit ZIP Code centroids. On the other hand, if you use street-level precision, MapMarker looks at both the street name and ZIP Code fields and attempts to return street-level coordinates and optionally fall back to ZIP Code coordinates, in many cases to ZIP+4 centroid.

The geocoding method defines the run type for geocoding operation. MapMarker will process records **automatically** or **interactively** (where you must choose the match from a list of candidates).

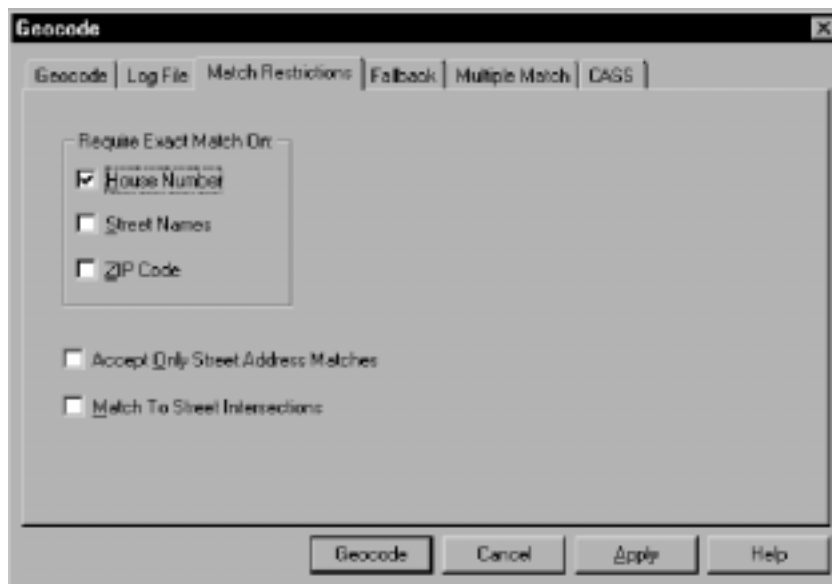
For automatic geocoding, the operation continues until all records are processed or you cancel the operation. For interactive geocoding, the Interactive dialog displays when MapMarker cannot determine the best match for the record based on your geocoding criteria.

Match Restrictions

The conditions in this dialog tell MapMarker how precisely you want to match your record. By default MapMarker requires an exact match on House Number, but not for Street Name and ZIP Code—conditions that strike the best balance among hit rate, performance, and minimizing erroneous matches.

When street name is unchecked, MapMarker will consider similar street names. When the ZIP Code is relaxed (unchecked), it will search surrounding ZIP Codes for a match. When ZIP Code is checked, MapMarker will only geocode records that have a ZIP Code.

Other conditions of matching that you can set here are accepting only street level matches and geocoding to street intersections. By accepting only street level matches, MapMarker will not place your records at a ZIP Code centroid.



For street intersections MapMarker can read input street addresses that contain an “&,” “&&,” or “and” between Street1 and Street2 in the same field. Note that this type of matching will affect MapMarker’s geocoding speed since it requires additional processing.

If you wish to change the matching conditions, clear or check the appropriate box in the group. For instance, clear the House Number box and check ZIP Code if you will allow a close house number as a viable match and want MapMarker to search for matches in the ZIP Code contained in your address.

Note: If you choose to relax the match on Street Name, MapMarker's performance will be affected because it has additional street match candidates to consider.

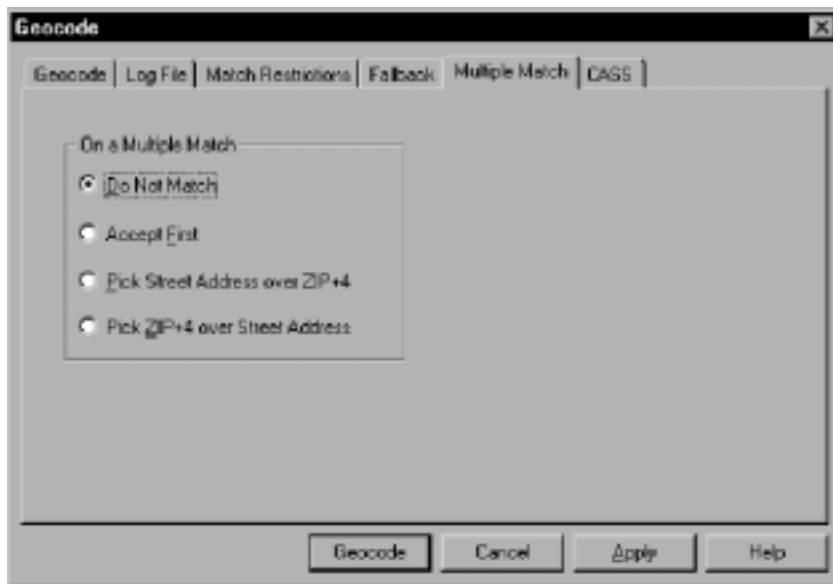
Fallback Settings

If MapMarker cannot match a record to street level, it will match to the ZIP Code centroid when you check the box Fallback to ZIP Code Centroid in the Fallback dialog. You can further define the fallback to a particular level of ZIP Code as well. For example, if you will only accept a ZIP+4 centroid as a fallback match, choose the last item (ZIP+4 Centroid only).

Effective with MapMarker 3.4, you may set the option to fallback to a particular ZIP Code type when you are geocoding to ZIP Code centroids.

Multiple Matches

If MapMarker determines that more than one match candidate exists for a record, it will follow the instructions you set in the Multiple Match dialog. If you choose Accept First, it will use the first candidate with the highest score as the match. Multiple match candidates are potentially strong matches, but none of them stands out enough from the others to be considered the definitive match.



CASS Certification Preferences

MapMarker can standardize your mailing addresses and print a report describing the success of the operation if you plan to qualify your table for bulk mailing discounts from the U.S.

Postal Service. This process is known as Coding Accuracy Support System (CASS) certification.

Because the address standardization requirements for CASS certification are so rigid, you have little control over much of the geocoding settings you normally have when geocoding with MapMarker. For example, you cannot geocode to a user dictionary when you are CASS geocoding. The options will be grayed out when CASS geocoding is in effect.


The default setting is to geocode **without** using CASS settings.

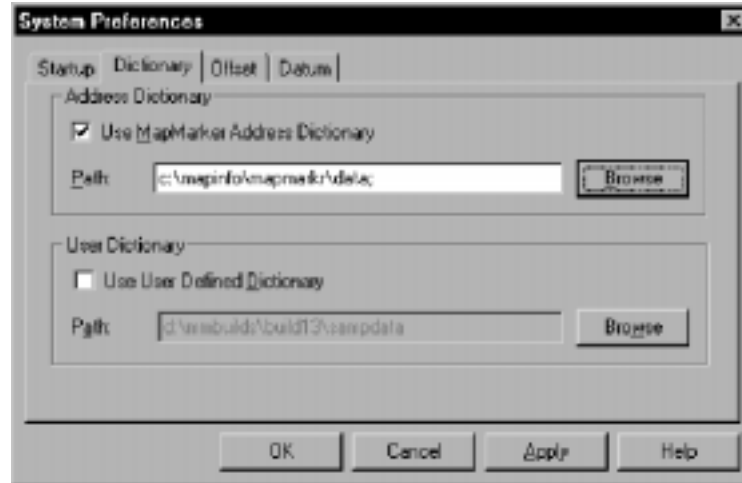
Log File Preferences

The Log file is a text file that MapMarker generates during geocoding to contain setup and processing information about the geocoding session. It contains the results of the session, including number of matches, non-matches, previously matched records, and the number of matched records by result code. MapMarker appends new information to the top of the Log file for each geocoding pass. The Log file is named from the root name of the table you are geocoding, for example, customer.log. The Log file is written to the MapMarker directory. The maximum file size is 1 MB.

Setting System Preferences

To control the behavior of MapMarker, you can set preferences that apply to the entire system.

The following dialog displays when you choose Options > System Preferences, click , or click the secondary mouse button display a pop-up menu from which to choose System Preferences. The System Preferences dialog contains four tabs: Startup, Dictionary, Offset, and Datum.



Any changes you make to these dialogs will be written to the Windows Registry at HKEY_CURRENT_USER\SOFTWARE\MAPINFO\MAPMARKR and saved for future geocoding sessions. MapMarker will also look in HKEY_LOCAL_MACHINE if a particular setting cannot be found in CURRENT_USER.

Startup

Refer to the section Setting Startup preferences on page 17.

Dictionary

The Address Dictionary group identifies the path of all installed address dictionaries. These can include those you purchased from MapInfo that cover the entire U.S., one or more states, or the ZIP+4 Centroid dictionary. The path(s) reflects all address dictionaries you identified during installation, whether the data was copied to a fixed drive or left on CD-ROM. When adding additional paths to the list, separate each by a semicolon. You do not need to specify the file name.

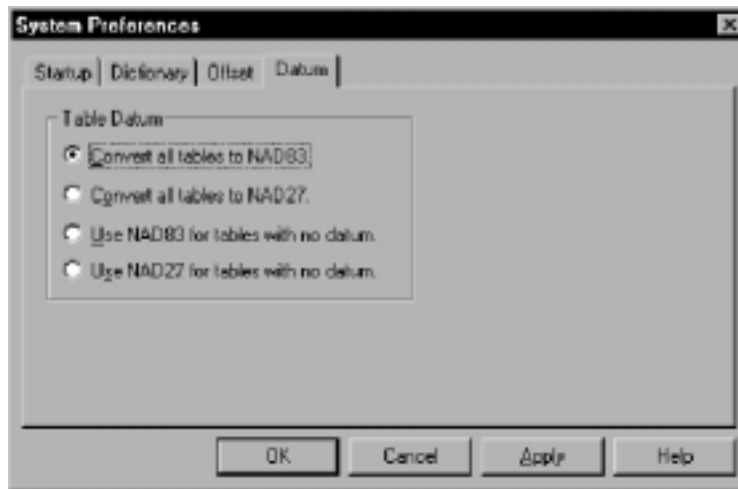
In addition, you can identify a customized dictionary that contains street addresses that may not be included in the purchased Address Dictionary. To use a customized dictionary, check the box marked Use User Dictionary and type the path or browse to its location. You can geocode using both the MapMarker Address Dictionary and a customized dictionary. The customized dictionary takes precedence when both dictionaries encounter a match. Instructions for creating a user dictionary are provided in Chapter 4.

Offset

These values reflect the distance the geocoded point will spot from the corner and road. When changing the units of the offset distance, be sure to also type the appropriate offset distance values, as changing the units **does not** automatically recalculate the offset values.

Datum

You can choose whether MapMarker returns matching coordinates with the datum in NAD83 or NAD27. The difference between them is in how a record matched to the same address in MapMarker can spot at one location as NAD27 coordinates and at a slightly different location as NAD83 coordinates. Consider how you plan to display your geocoded points. If your street data is in NAD27, geocode your table using the NAD27 setting. Keep in mind that the new standard for TIGER 95 data is NAD83.



To choose the datum settings, select the item in the Datum dialog that best meets your needs. To keep your coordinates meaningful, geocode the entire table using the same datum setting. On previously geocoded tables, be sure to know the datum for those geocoded records.

For more on datums, see Understanding Datums in Chapter 4.

Chapter 4: Making the Most of MapMarker

Overview

MapMarker is a powerful geocoding tool that lets you control much of the operation for your particular needs. Understanding its full potential will help you get the most from MapMarker, and as a result, the most from your data. Now that you are familiar with running MapMarker, this chapter presents in-depth discussions on the finer points of geocoding successfully.

In This Chapter . . .

Maximizing Match Rate and Accuracy	50
Choosing Match Strategies	52
Impact of Relaxing Match Conditions	55
Re-Geocoding a Table	56
Making the Most of Interactive Geocoding	56
Understanding Result Codes	59
S3 and Z3 Result Codes: What's the Difference?	64
Census Block Codes	65
Performance Tuning	65
Creating a Customized User Dictionary	66
Understanding Datums	69

Maximizing Match Rate and Accuracy

MapMarker contains a powerful matching system that allows you to control the conditions under which it attempts to match your records. With this power comes trade-offs among the number of close matches, geographically accurate matches, and false matches. The conditions you choose depend on your intended use of the geocoded data.

It is important that you consider questions like the following when setting up a geocoding session:

- What level of matching accuracy are you looking for (exact address match, close match)?
- What level of geographic accuracy do you need for your geocoded points (street level, ZIP Code centroid)?
- Is your goal to geocode as many records as possible?

The answers to these questions are driven by how you intend to use the geocoded records. For example, perhaps you are determining the location of a new retail location and need to know the distribution of current and potential customers. In this case, you want to geocode as many of these customers as possible and do not need an exact street address match for each one. Geocoding to ZIP+4 centroid or ZIP Code centroid is fine for your analysis.

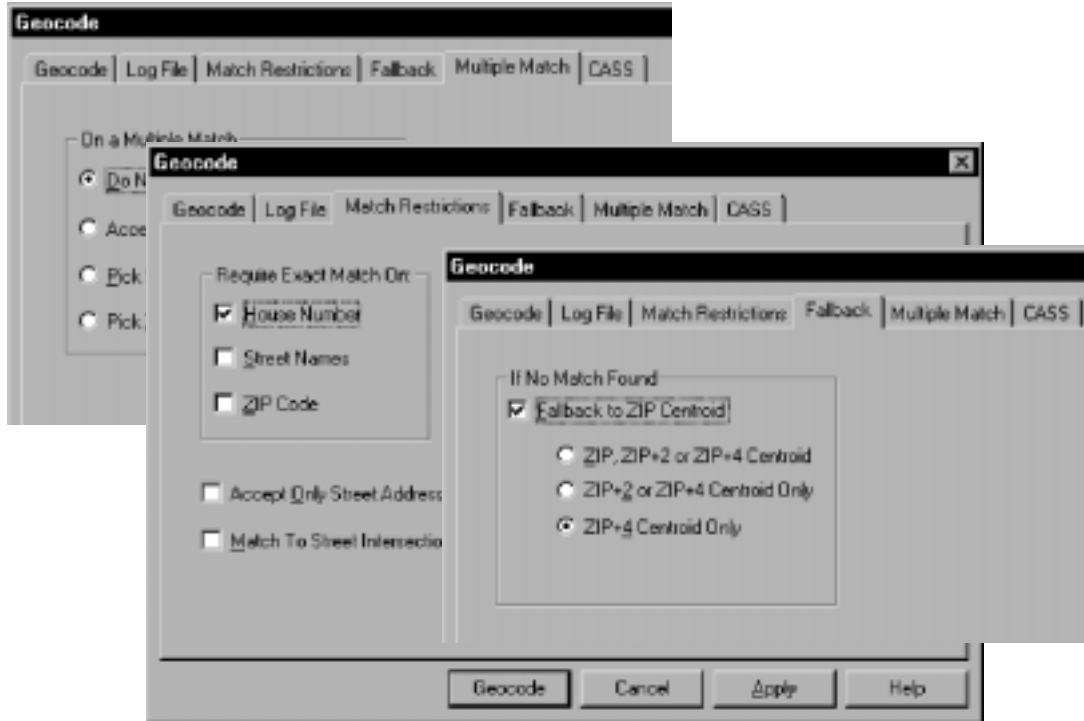
On the other hand, if you, as a utility service coordinator, need to know where your customers are in relation to neighborhood gas lines, the positional accuracy of each customer is of critical importance to you.

This section explains the impact of the matching conditions and presents some strategies that will help you set MapMarker to meet your needs.

Matching Conditions

Under the Geocode dialog are tab dialogs that allow you to set the match restrictions, fallback and multiple match settings for a geocoding pass. Here you set MapMarker to be as strict or relaxed as you need by checking or clearing the appropriate box. The default settings, which give you the best compromise among match rate, accuracy and performance, are:

- require an exact match for house number only (set in the Match Restrictions dialog)
- do not fall back to ZIP Code centroid matching (Fallback dialog items unchecked)
- do not match if there are multiple match candidates (Multiple Match dialog items unchecked).



To set stricter matching conditions, require an exact match on house number, street name and ZIP Code and do not to fall back to any ZIP Code centroid matching. MapMarker, in essence, is looking for an exact street address match within the ZIP Code named in your input address.

Relaxing the conditions tells MapMarker to broaden the area in which it searches for a match. For example, by relaxing the ZIP Code, MapMarker will search for candidates outside the ZIP Code but within the city for your input address.

To maximize the potential for a match, tell MapMarker what to do if it does not find a match. These settings include Fallback to ZIP Code centroid matching and matching to the first of multiple matches.

For P.O. Box and rural route addresses, MapMarker automatically geocodes them to a ZIP Code centroid, as required by CASS standards.

Each of the settings above affect the match rate, performance and the positional accuracy of the geocoded point. You must decide the best settings for your needs.

Choosing Match Strategies

Now let's look at some matching strategies, including:

- Striking a balance between match rate and false positives (erroneous matches)
- Maximizing match rate
- Maximizing geographic accuracy

You can choose one method or use them in combination. Each is discussed below.

Striking a Balance Between Match Rate and False Positives

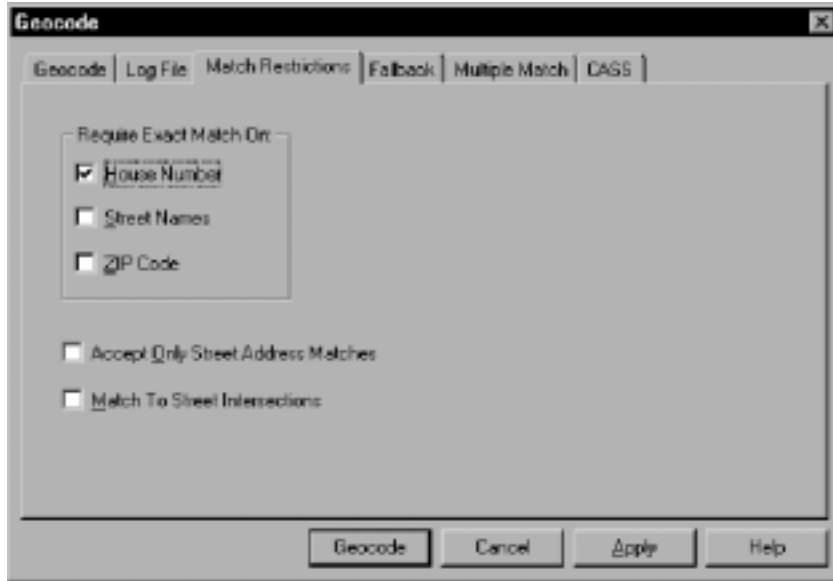
This strategy is best used when you want to geocode as many records as possible automatically, but want to minimize the number of erroneous matches (false positives). You would rather have those records remain ungeocoded so that you can geocode them interactively later. It is likely that you want to analyze your data at a local level where the location of each record on a map is important.

False positives are matches MapMarker makes when it finds a street that sounds like the input street; or finds the same street in another town (if ZIP Code is relaxed), or finds the street but with a different house number (if House Number is relaxed). MapMarker, however, contains a sophisticated matcher that will usually find the correct address.

MapMarker's default settings reflect this strategy— exact match is required for house number, but is relaxed for street name and ZIP Code. P.O. box and rural route addresses are automatically geocoded to the ZIP Code centroid. You do not need to set any conditions specific to these addresses.

Maximizing the Match Rate

To generate the highest match rate possible, set MapMarker to relax house number, ZIP Code and street names in the Match Restrictions dialog. In addition, check Accept First (Multiple Match dialog) and Fallback to ZIP Centroid (Fallback dialog).



While this scenario will likely yield false positives, this is the best match solution when you have large databases that need to be geocoded. You should decide if the percentage of false positives will affect your analysis.

- ✔ **Note** You can evaluate false positives if you have MapMarker return output street, city, state and ZIP Code values. Look for false positives by comparing the input address with the output address.

To further decrease the number of false positives without sacrificing hit rate, analyze the result codes after a geocoding session and re-geocode them interactively. More on result codes is found later in this chapter.

Maximizing Geographic Accuracy

If positional accuracy of the geocoded records is the most important aspect for you, choose the third strategy in which MapMarker returns the lowest number of erroneous matches. To do this, you must require MapMarker to find exact matches for all portions of the address – house number, street name and ZIP Code. Also, do not accept the first of multiple matches or fall back to ZIP centroid. These records will remain ungeocoded.

Choose this scenario when your analysis requires precise locations for your records. To increase the number of matches, your next step might be to use interactive geocoding. Interactive geocoding is discussed later in this chapter.

Picking Your Strategy: An Example

The following scenario outlines a three-pass geocoding session based on the first strategy: the user is looking to generate the highest match rate for a large database while minimizing false positives. The goal during the session is to set the conditions for that strategy in the first pass, then follow with an interactive session to pick up additional matches, and finally, a third pass to ZIP Code centroids to geocode any remaining records.

First Pass

Geocode the table **automatically** with the settings that will generate the highest match rate with the fewest false positives:

- Require exact match on house number, relax street name and ZIP Code (default settings). For even stricter matching, require exact match on street name and ZIP Code as well.
- Do not accept first of multiple matches.
- Do not fall back to ZIP Code centroid matching.

All records geocoded during this pass will be exact or close matches that spot at the street level. Depending on the cleanliness of your data, the majority of your table will be processed during this pass. This pass will generate the least number of false positive matches.

Second Pass

For this pass, geocode all unmatched rows **interactively** with the following settings:

- Relax the settings for house number, street name and ZIP Code
- Do not accept first of multiple matches.
- Do not fall back to ZIP Code centroid matching.
- Edit the input address where necessary to increase the likelihood of a match.

Each match in this pass is still at street level accuracy. Since you are in interactive mode, you control every match, thus, minimizing the number of false positives.

You can also change the match settings (Options button), expand the Search, and edit the record to find additional match candidates.

Third Pass

Perform a final geocoding pass on unmatched rows by geocoding to **ZIP Centroid**. Set the geocoding precision in the Geocode dialog to To ZIP Centroid.

This pass will pick up virtually all remaining records and geocode them to ZIP Code centroid accuracy. If your data includes ZIP+4 information, some of these ZIP Code centroid matches could spot at the street level in as nearly the same location as exact and close street matches from the first pass.

One Pass Settings

If you do not want to geocode your table interactively, combine the first and third passes using the following settings in the Options dialog:

- Require exact match on house number, relax street name and ZIP Code.
- Do not accept first of multiple matches.
- Choose Fallback to ZIP Centroids.

Impact of Relaxing Match Conditions

Consider the following when you change the settings in the Geocode dialog.

Relaxing ZIP Codes

When ZIP Codes are relaxed, MapMarker will search a wider area for a match. While this will result in slower performance, the match rate will be higher because MapMarker does not need to match exactly when it compares match candidates.

Relaxing Street Name

MapMarker will look at all candidates with names that sound like the input address. This will slow down MapMarker's performance. On the plus side, since more candidates are examined, the match rate increases. If your table is indexed, the time difference between performance and match rate will be reduced.

Relaxing House Number

Performance is not significantly affected when the house number setting is unchecked. It does, however, affect the type of match if the candidate address corresponds to a TIGER segment that does not contain any ranges. The type of match can also be affected when the house number range for a candidate does not contain the input house number.

Matching to Street Intersections

If you select the option Match to Street Intersections you will notice a significant drop in geocoding speed since MapMarker is now comparing two street names for every record. Choose this option only if your table contains a large number of street intersection records.

Geocoding to ZIP Centroids versus Falling Back to ZIP Centroid

When MapMarker matches to ZIP centroid it only looks at the ZIP Code in the input address. If your data has a 5-digit ZIP Code, MapMarker can only match to the 5-digit ZIP centroid. If, on the other hand, you geocoded the same record at street level with a fallback to ZIP centroid, MapMarker takes into account both the 5-digit ZIP Code and street name and can likely match it to a more precise location than the 5-digit ZIP centroid.

Re-Geocoding a Table

You may often find that once a table is geocoded, records become updated or new ones added and you need to geocode again. New or previously ungeocoded records (those that do not have result codes) can simply be re-geocoded. By default MapMarker automatically geocodes only unmatched records. This means that for a large database, previously matched records are not re-matched, thus saving time.

Records that already contain coordinates, but are incorrect due to a change in address for example, must be identified and the result code deleted. MapMarker will treat any record without a result code as an unmatched record and attempt to find a match. MapMarker will also re-geocode records with an “N” or “ND” codes.

If your table was geocoded previously with MapInfo, keep in mind that MapMarker will only replace the points for matches it makes. It does not touch the points for records it cannot match.

MapMarker saves the columns settings and geocoding options you set the first time you attempt to geocode a table. Re-geocoding the same table is simple. Open the table, press enter twice and geocoding begins. To view the settings, open the .tab file in a text editor and find the section called “Metadata.”

Making the Most of Interactive Geocoding

Interactive geocoding allows you to control the matching process because you choose to accept or ignore a potential match from the list of suggestions that MapMarker has determined best fit your record.

Change Matching Conditions

MapMarker follows the settings in the Geocode dialog or responds to your changed settings as you progress through the geocoding pass. To increase the list of suggested matches, relax

street name. You can do this at the beginning of an interactive pass or change the settings for each record by accessing the Options button from the Interactive dialog.

You can also return to a previous record in the table with the Back button and change settings as needed. This feature can help you maximize the potential for a match.

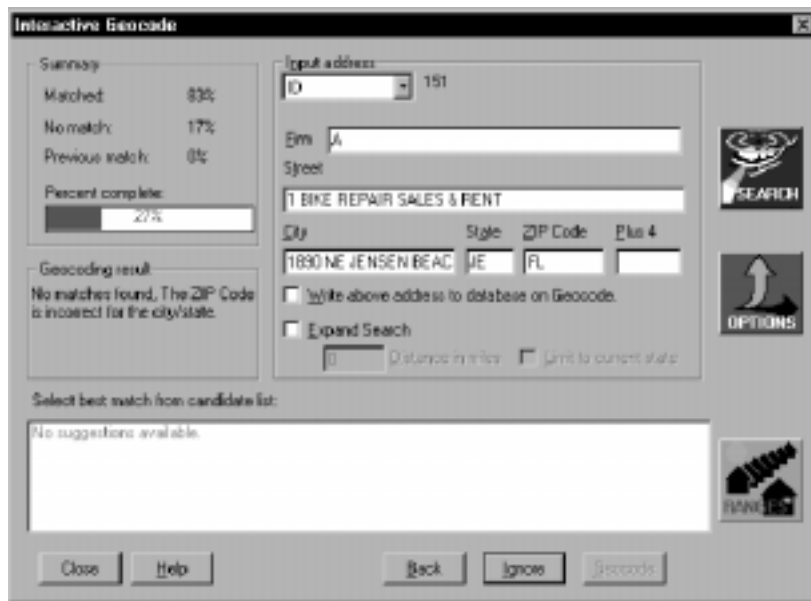
Expand Your Search

Once at the Interactive Geocode dialog, you can further increase the likelihood of a match by expanding the search area in which MapMarker will check for a match. Check the box Expand Search and set the radius in miles (up to 99) around which your record lies. You can contain the search to the state by checking the Limit to current state. MapMarker checks all finance areas whose centroids lie within the radius.

Edit Your Records

Interactive geocoding also allows you to correct your record to improve the possibility of a match. If you choose, your table can be updated to reflect the newly edited record, all from within the interactive dialog.

For example, the following illustration shows the interactive dialog displaying a record with address fields shifted into the wrong fields. MapMarker is unable to find a suitable match for the information as is.



Chapter 4: Making the Most of MapMarker

By manually correcting the fields to put the firm, street, city and state into their proper locations, the record is more logical and MapMarker can make better suggestions on possible matches. After editing the record click the Search button. A new list of match candidates displays:

The screenshot shows the 'Interactive Geocode' window. On the left, a summary box shows 'Matched: 83%', 'No match: 17%', and 'Previous match: 0%'. Below this, a 'Percent complete' progress bar is at 27%. The 'Geocoding result' section states 'Unique close match found'. The main input area has 'Input address' set to '151'. The 'Firm' field contains 'A1 BIKE REPAIR SALES & RENT'. The 'Street' field contains '1850 NE JENSEN BEACH BLVD'. The 'City' field contains 'JENSEN BEACH', 'State' contains 'FL', and 'ZIP Code' contains '34957'. There are checkboxes for 'Write above address to database on Geocode.' (checked) and 'Expand Search'. A 'SEARCH' button is on the right. Below the input fields, a table lists candidate matches:

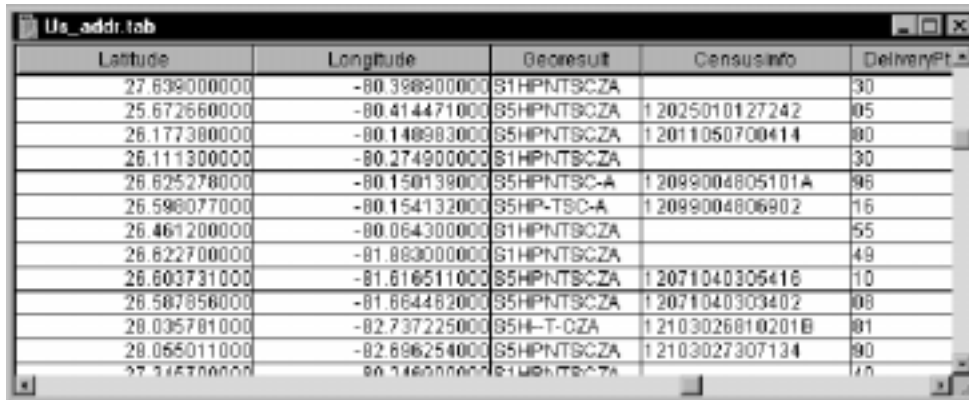
Select best match from candidate list:		
1850	NE JENSEN BEACH BLVD	JENSEN BEACH, FL 34957-7278
	JENSEN BEACH BLVD	
1499	NW JENSEN BEACH BLVD	JENSEN BEACH, FL 34957-4425

At the bottom, there are buttons for 'Close', 'Help', 'Back', 'Ignore', and 'Geocode'.

To write these changes to the base table, check the box “Write above address to database on Geocode” and click Geocode to confirm the match. **Important:** your edits will overwrite the address data in the respective fields, so be sure not to alter the fields containing the correct address data.

Understanding Result Codes

As an output option, MapMarker returns a result code for every record it attempts to match. The code represents the success or failure of the geocoding operation as well as conveys information about the quality of the match. Each character of the code tells how precisely MapMarker matched each address component.



Latitude	Longitude	Georesult	Censusinfo	DeliveryPt
27.639000000	-80.398900000	S1HPNTSCZA		30
25.672660000	-80.414471000	S5HPNTSCZA	12025010127242	05
26.177380000	-80.148983000	S5HPNTSCZA	12011050700414	00
26.111300000	-80.274900000	S1HPNTSCZA		30
26.625278000	-80.150138000	S5HPNTSC-A	12089004805101A	06
26.598077000	-80.154132000	S5HP-TSC-A	12089004806902	16
26.461200000	-80.064300000	S1HPNTSCZA		55
26.622700000	-81.883000000	S1HPNTSCZA		49
26.603731000	-81.616511000	S5HPNTSCZA	12071040305416	10
26.587856000	-81.664482000	S5HPNTSCZA	12071040303402	08
28.035781000	-82.737225000	S5H-T-CZA	121030268102018	01
28.065011000	-82.896254000	S5HPNTSCZA	12103027307134	00
27.715700000	80.716200000	S1HPNTSCZA		10

The result codes are written to a result column that you specified in the Select Output Columns dialog at the beginning of the geocoding process. You can either create this column before geocoding or have MapMarker create it automatically (default name GeoResult) when you open your table.

The code is an alphanumeric code of 1-10 characters. The codes fall into four major categories:

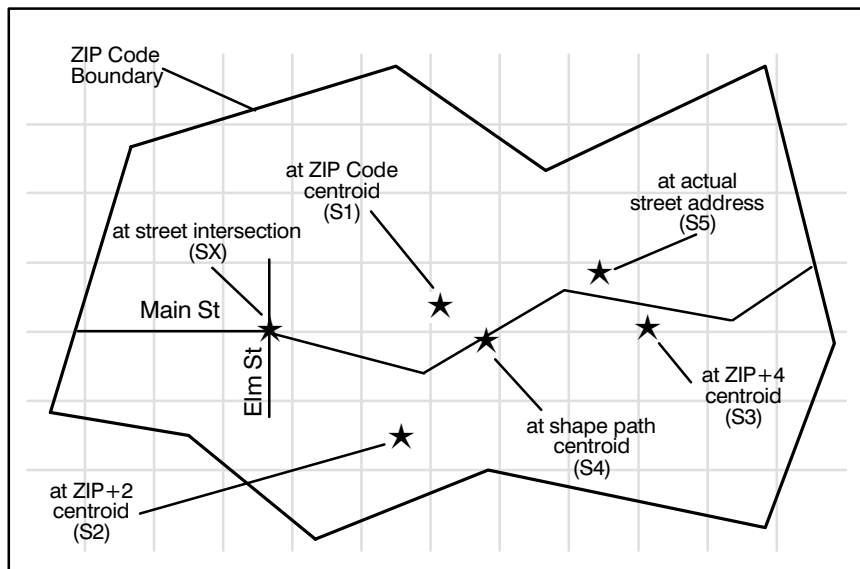
- Single close match
- Best match from multiple candidates
- ZIP Code centroid match
- Non-match

The codes reflect the settings you choose or accept in the Geocode dialog. They also reflect the quality of the data in both your table to be geocoded and the Address Dictionary. Each category is explained below.

Single Close Match (S category)

Matches in the S category indicate that the record was matched to a single address candidate. The first character (S) reflects that MapMarker found a street address that matches your record. The second position in the code reflects the positional accuracy of the resulting point for the geocoded record, as indicated below.

- S1 single close match, point located at ZIP Code centroid
- S2 single close match, point located at ZIP+2 centroid
- S3 single close match, point located at ZIP+4 centroid
- S4 single close match, point located at the center of the shape point path (shape points define the shape of the street polyline)
- S5 single close match, point located at the street address position (highest accuracy available)
- SX single close match, point located at street intersection
- S0 single close match, no coordinates available (very rare occurrence).



Best Match From Multiple Candidates (M category)

Matches in the M category indicate that there is more than one close match candidate for the record and MapMarker has chosen the best one of those candidates. This category is used only when you check the box marked Accept First in the Multiple Match dialog tab and MapMarker finds more than one strong match candidate. Like the S category, the second position in the code reflects the positional accuracy of the resulting point object.

M1 multiple close matches, point located at ZIP Code centroid

M2 multiple close matches, point located at ZIP+2 centroid

M3 multiple close matches, point located at ZIP+4 centroid

M4 multiple close matches, point located at the center of the shape point path (shape points define the shape of the street polyline)

M5 multiple close matches, point located at the street address position (highest accuracy available)

MX multiple close matches, point located at street intersection

M0 multiple close matches, no coordinates available

For either S or M category result codes, eight additional characters describe how closely the address in your table matches an address in the Address Dictionary. The characters appear in the order given. Any non-matched components are represented by a dash.

Result Code	Description	Example
Component		
H	House Number	110
P	Street Prefix	North
N	Street Name	Fletcher
T	Street Type	Place
S	Street Suffix	SE
C	City Name	Boulder
Z	ZIP Code	80303
A or U	Address Dictionary or User Dictionary	

For example, the result code S5- -N-SCZA represents a single close match that matched the street name, street suffix direction, city and ZIP Code exactly, but could not match the house number, street prefix direction or the street type. The match came from the MapMarker Address Dictionary. This record would spot at the street address position of the match candidate.

Postal Centroid Matches (Z category)

Matches in the Z category indicate that no street match was made, either 1) because there is no close match and you allowed MapMarker to fall back to ZIP Centroid; 2) the address is a P.O. Box or rural address or, 3) you were matching to ZIP Centroids. The resulting point is located at the ZIP Code centroid with three possible accuracy levels.

- Z1 ZIP Code centroid match
- Z2 ZIP+2 centroid match
- Z3 ZIP+4 centroid match (highest accuracy available)
- Z0 ZIP code match, no coordinates available (very rare).

Non-match Codes

The following result codes indicate no match was made:

- N No close match. These records can be re-geocoded interactively or during subsequent automatic passes under different matching conditions.
- NX No close match for street intersections.
- ND MapMarker could not find the Address Dictionary for the given ZIP Code or city/state. These records can also be re-geocoded once the Address Dictionary is available.
- NG The user marks these records during interactive geocoding as non-geocodable. MapMarker will not attempt to match these records again until the code is removed.

GeoResult Dialog

To view a breakdown of the result code for each record, double-click on the code in the result column in the Browser. This behavior is available after you have set the columns for geocoding or completed a geocoding pass. It is not available at the initial opening of the table.



Information about the record displays showing the address, result code, match type and match details by address component. The dialog is modeless; that is, you can continue to click on result codes in your table while this dialog is open and it will display the result code information for that record.

- Note** You may notice that in the above illustration, the street prefix and suffix directions are checked, indicating the record matched on those components. The address 280 NW 41st St only contains a prefix direction. MapMarker considers it a match when both the record to geocode and the match candidate from the Address Dictionary have the same components. In this case, although neither address contain a suffix, MapMarker considers it a match on that component.

To increase the number of successful matches, using MapInfo, search your table for records containing the same code. You can then edit and re-geocode them to increase the likelihood of a match.

S3 and Z3 Result Codes: What's the Difference?

One of the most often asked questions of our Tech Support staff is what is the difference between S3 and Z3 result codes.

An S3 match is defined as a single close match with the point located at the ZIP+4 centroid. The Z3 match is also spotted at the ZIP+4 centroid. The difference lies in how the MapMarker arrives at the results.

For the S3 record, MapMarker found a street address that matches, but the match record did not contain any street geometry (line points). MapMarker is, therefore, unable to interpolate where along the segment to place the record. The best it can do is to spot it at the ZIP+4 centroid.

A Z3 match, on the other hand, is a direct match to the ZIP+4 centroid. MapMarker could not find a street address match for one of several reasons: 1) you set the match setting to ZIP Codes, 2) there was no close match and you set your fallback criteria to ZIP Centroid, or 3) the address is a P.O. Box or rural route.

On a map, the S3 and Z3 records display at the same location, assuming the input ZIP Codes were the same for both records. However, the matching process for a S3 record has an extra step that could place the point at a different, and more accurate, ZIP+4 centroid. During a street address match, MapMarker will correct the input ZIP Code if the match record includes different information. Z3 matches do not get corrected. Because of this correction step, the S3 match is considered the more accurate match.

Census Block Codes

MapMarker returns a Census Block ID for matched records if you identify an output column prior to geocoding. This code is a unique identifier from the U.S. Census Bureau that defines the smallest geographic area for census reporting. The 15-digit alphanumeric code is interpreted as follows:

SSCCCTTTTTTGBB(B)

Where S = State FIPS code (Federal Information Processing Standard) (2 characters)

C = County FIPS code (3 characters)

T = Census Tract (6 characters)

G = Census Block Group (1 character)

B = Census Block (typically 2 characters. The third character means the block was split from another block).

Performance Tuning

To get the most out of MapMarker's top geocoding performance, consider these tips for optimizing your system.

- Use 32-bit MapMarker on a 32-bit operating system (Windows NT or Windows 95)
- Use the faster processor available to you.
- Have enough memory so that the operating system can allocate some memory to your disk cache.
- Copy the Address Dictionary to your hard drive.
- Sort your table by ZIP Code.
- Choose exact match criteria for all (house number, street name, ZIP Code).
- Do not create points automatically.

Creating a Customized User Dictionary

A user dictionary is a file that you can add to your MapMarker dictionary search path to override or extend the address information in the purchased MapMarker Address Dictionary. The user dictionary is similar in concept to the user dictionaries that many spell checkers use to allow you to add special cases to the main spelling dictionary.

You create a user dictionary from a MapInfo table by specifying fields of address information which MapMarker converts into a dictionary format. The MapInfo table must contain certain required fields.

Creating a user dictionary in MapMarker starts with a MapInfo table of street segments or points that represent addresses you wish to geocode to. The chart below identifies the required and optional fields in the MapInfo table.

Required Fields	Optional Fields
Left Start Address	City
Left Start Address	Left ZIP+4 code
Right Start Address	Right ZIP+4 code
Left End Address	Left Census Block
Right End Address	Right Census Block
Street Name	Left Odd/Even indicator
State Abbreviation	Right Odd/Even indicator
Left and Right ZIP Code	Place Name

Effective with MapMarker 3.1, the source table for your User Dictionary must contain left and right ZIP Code columns. The City column is now an optional field.

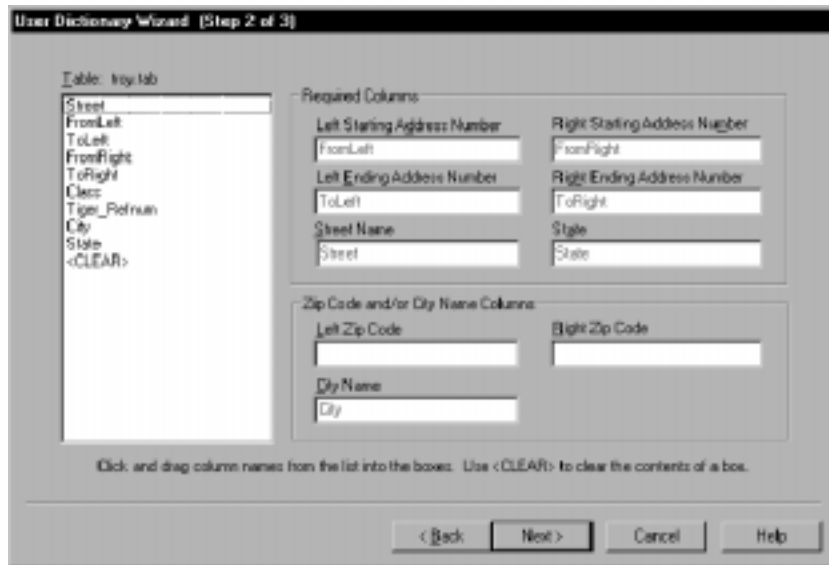
Note: MapMarker will only use a user dictionary if you have licensed the areas covered by that dictionary. For example, if you create your own New York City user dictionary, you must have a license file for NY state or the entire US.

The more user dictionaries you use, the more geocoding performance will degrade. We recommend not using more than five user dictionaries. If you need to use more than five, use Append.mbx (provided with MapMarker) to append some of your MapInfo tables together. Append.mbx is found in the MapMarker program directory. The Append utility is discussed in the Appendix.

MapMarker provides a wizard to walk you through the setup phase of the user dictionary creation process.

To create a user dictionary:

1. Choose File > Create User Dictionary. The User Dictionary Wizard (Step 1 of 3) dialog displays.
2. Specify the path and filename of the MapInfo table from which address information will be derived. Alternatively, click the Browse button to reach the desired location of the table.
3. Specify how much of the table you wish to include in the dictionary by specifying the row numbers or leave All Rows selected.
4. Click Next to continue. The User Dictionary Wizard (Step 2 of 3) dialog displays.



5. Specify the required fields in the source table by highlighting a field name in the list box and dragging it to the appropriate box in the address groups. Note: you must fill in either the City and State boxes or the Left ZIP Code, Right ZIP Code and State boxes.
When selecting the state field choose the field that contains the two-letter state abbreviation. Do not use the 2-digit FIPS code field as the state field.
6. To clear any box, highlight <CLEAR> in the list and drag it over the column name you wish to delete.
7. Click Next to continue. The User Dictionary Wizard (Step 1 of 3) dialog displays.
8. Specify any optional fields for ZIP+4, Census Block or place name columns you wish to include in the dictionary. A place name is a point object, rather than a street segment. Examples include Sears Tower, Wrigley Field, or City Hall.
9. Click Finish to proceed or Back to revisit the previous dialogs. When you click Finish the Save User Dictionary As dialog displays.

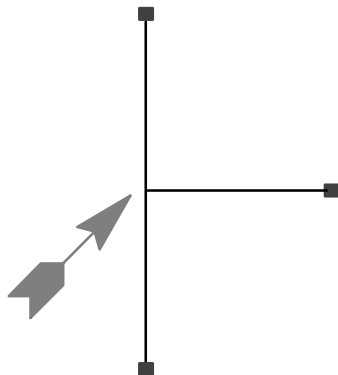
10. Specify the location of the user dictionary. Note: it must be in a different location than the MapInfo source table.
11. Choose Options > System Preferences > Dictionary. The Dictionary dialog displays. Check the User Dictionary checkbox and provide the path and filename in the content box.

You can geocode with the MapMarker Address Dictionary and your custom user dictionary at the same time. Matches made from the user dictionary will take precedence over matches from the MapMarker dictionary.

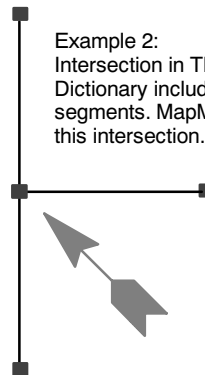
Note: You cannot geocode to CASS standards with a user dictionary. The Use User Dictionary option in the Dictionary dialog will be grayed out.

Street Intersections and Customized User Dictionaries

When geocoding to street intersections with a custom User Dictionary, MapMarker cannot recognize the intersections if one or more of the segments that make up the intersection does not have an endpoint at the intersection. This can happen when you create the User Dictionary from a customized street table where intersections were not constructed with every segment ending with an endpoint at the intersection (Example 1).



Example 1:
Intersection in custom User Dictionary does not have endpoints for all segments. MapMarker does not recognize this as an intersection.



Example 2:
Intersection in TIGER-based Address Dictionary includes endpoints for all segments. MapMarker will geocode to this intersection.

This is not a problem when geocoding to street intersections with the MapMarker Address Dictionary or a custom User Dictionary based on TIGER 95 street geometry (Example 2).

Understanding Datums

A datum is a reference point from which geographic coordinates are measured. Recently, the U.S. Census Bureau released the latest version of its TIGER Line data using a new datum reference. TIGER 95 uses as its standard the North American Datum of 1983, known as NAD83.

NAD83 represents the most accurate and comprehensive geodetic survey for the U.S. to date. It is based on a model that calculates the earth's ellipsoid slightly differently than NAD27. It supersedes NAD27 and provides a consistent measurement across North America at a higher degree of accuracy.

NAD83 and the MapMarker Address Dictionary

MapMarker's Address Dictionary was created with TIGER 95 source data that uses NAD83 as its datum reference. This means that the longitude and latitude coordinates that MapMarker assigns to geocoded records will use the NAD83 datum and will differ slightly from records previously geocoded against NAD27-based Address Dictionaries (such as MapMarker 2.1 or earlier). Coordinate pairs in NAD83 and NAD27 differ on average by +/- 5 meters.

Because of this transition from one datum standard to another, MapMarker 3.x allows you to control which datum you want for your geocoded records. Your decision to use NAD83 or NAD27 should rest with how you intend to display your geocoded records and whether the point differential will affect your analysis.

For example, if you will display your records against street data based on NAD27 datum, then you should geocode your table to that datum. If your display data is in NAD83, geocode to NAD83 so that your points and the street data will match.

Here are the datums for MapInfo's street products:

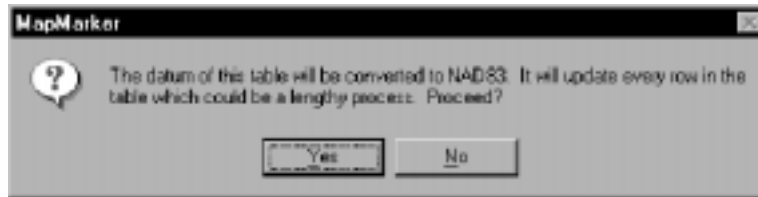
Street Product	Datum
StreetInfo V 2.0/3.0	NAD27
StreetInfo V 4.0 *	NAD83
StreetWorks 1.0/2.0	NAD27
StreetWorks 3.0	NAD83

* Future release

A general rule for TIGER-based data products is TIGER 90, 92 and 94 use NAD27 and TIGER 95 uses NAD83.

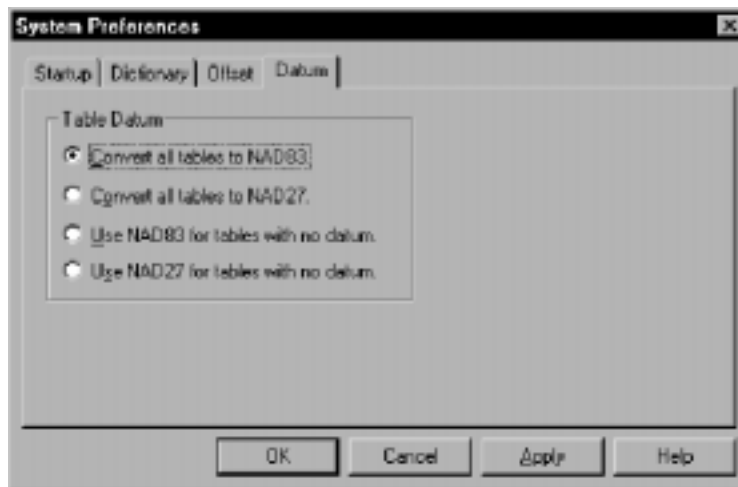
How MapMarker Determines Which Datum to Use

MapMarker looks in the table's metadata for a datum key and in the .map file (if one exists) and compares them against the setting in the System Preferences > Datum dialog. If there is a datum mismatch, MapMarker displays a "prescan" message, prompting you to resolve the conflict.



For example, if your table has a metadata datum key of NAD27 and the Datum Preferences dialog is set to Convert all tables to NAD83, the prescan message box displays. If you answer Yes to convert to NAD83, MapMarker will geocode records to that datum. If you answer No, MapMarker will override the Datum Preference setting and geocode your table to the original NAD27 datum.

If MapMarker cannot determine the datum for a table, it follows the setting in the Datum Preference dialog. The default is to Convert tables to NAD83.



The following will help you determine what datum will be used for your table.

Your table will be geocoded with a NAD83 datum if:

- it has a metadata datum key of NAD83 or map file in NAD83 and you do not tell MapMarker to convert it to NAD27, or
- you choose to convert it to NAD83 by saying Yes at the prescan dialog or accepting that setting in Datum Preferences dialog, or
- your table is in a coordinate system that is neither long/lat NAD83 nor NAD27¹

Your table will be geocoded with a NAD27 datum if:

- it has a metadata key of NAD27 or map file in NAD27 and the Datum Preferences dialog is set to convert to NAD27, or
- you choose to convert it to NAD27 by saying Yes at the prescan dialog or accept that setting in the Datum Preferences dialog.

To view the metadata for your table, open the .tab file in a text editor and look under the section called "begin_metadata." The datum key is listed as MapMarker\Preferences\Datum with possible values of NAD83 and NAD27.

Setting datum preferences is discussed in Chapter 3.

Datums and ODBC Tables

MapMarker is unable to store metadata in ODBC tables so you will be prompted to choose the datum setting every time you open the table in MapMarker.

¹ Points will be created in the original system. Long/lat values will be returned in NAD83. Examples of these tables are long/lat datumless or State Plane system.

Chapter 5: MapMarker Client/Server Toolkit

Overview

The Client/Server Toolkit allows developers to add MapMarker geocoding to their client/server applications. These applications run the gamut from roadside assistance, where-is-the-nearest-store, visual multiple listing services for real estate, and telecom call-before-you-dig applications. These applications can be deployed in traditional client/server configurations as well as Internet and intranet environments.

The Client/Server Toolkit consists of two components, the MapMarker Server and the MapMarker Client OCX, each of which is discussed here.

Note: The OCX is also referred to as the MapMarker Geocoder Control in the Supplemental Guides that shipped with versions 3.1 through 3.4.

In This Chapter . . .

Client/Server Geocoding	73
MapMarker Server	73
Running MapMarker Server	74
MapMarker Client OCX	76
Adding MapMarker OCX to Your Client Application	77
Setting MapMarker OCX Properties	80
Geocoding via the Client OCX	81
Falling Back to ZIP Code Centroids	81
MapMarker OCX Events and Methods	82
The Toolkit and the Internet	84

Client/Server Geocoding

MapMarker 3.x comes complete with a Client/Server Toolkit that allows MapMarker to geocode records from multiple users using a single geocoding engine. The Toolkit components include MapMarker Server and MapMarker Client Geocoder Control (OCX). MapMarker Server can be run as a Windows NT Service or Windows 95 console application.

If you wish to geocode in a client/server environment, be sure to choose the Client/Server Toolkit component during installation. You can install this component at any time by running **setup.exe** from the MapMarker CD #1. At the Software Installation dialog choose the Client/Server Toolkit option (clear all other component checkboxes if they are already installed) and proceed through the installation. The installer will install the MapMarker Server and Client OCX and register them properly on your machine.

Installing MapMarker Server and Client OCX Manually

If you need to install MapMarker Server manually:

- ▶ At the DOS prompt, run `mm_serve -install`

To register the OCX control manually:

- ▶ At the DOS prompt, run `regsvr32 mapmarkr.ocx`

Regsvr32.exe is found in the MapMarkr program directory on the target machine or in the main network installation directory.

If you cannot register the OCX, make sure you have the following .dlls in your windows\system subdirectory:

```
msvcrt.dll
msvcrt40.dll
```

Setting up Client Sites

Once MapMarker Server is installed, run **setup.exe** from the server machine to set up a client. This applies to both Windows NT and 95 client sites.

MapMarker Server

MapMarker Server is a layer on top of the MapMarker Geocoding Engine, and extends the engine's functionality by providing a Remote Procedure Call (RPC) facility and a

queuing/multi-threading function. The RPC interface of the MapMarker Server allows it to receive via TCP/IP geocoding requests from remote clients (such as the MapMarker OCX). Since the Geocoding Engine handles one request at a time, the MapMarker Server queues multiple requests until they can be fulfilled by the Geocoding Engine.

The MapMarker Server can handle a maximum of 1,024 simultaneous requests. In a typical customer service environment, a request might consist of an address being sent to the MapMarker server, displaying the candidate list and then choosing the best candidate.

Even though the request may span 20 seconds or even 20 minutes, the Geocoding Engine is only occupied for an extremely short period of time. Depending on your hardware and network environment, the Geocoding Engine will geocode as many as 25 to 100 records a second. So even though there is one Geocoding Engine, many users may be served "simultaneously".

The MapMarker Server RPC interface is defined in Chapter 12 (not included in this abridged version of the Product Guide). Most applications will not have to use this interface. Instead, we recommend that you use the MapMarker OCX. Application development should be easier with the OCX since you will probably have to use C or C++ to access the RPC interface.

Note: The current implementation of the MapMarker Server does not support the Browse or GetNextRange features of the MapMarker Geocoding Engine. If you need these functions, you should use the MapMarker API directly.

Running MapMarker Server

To run MapMarker Server as a Windows NT Service or console application in Windows NT or Windows 95, see the appropriate section below.

NT Service

To configure MapMarker Server as a Windows NT Service:

1. From the Control Panel in Windows NT, choose the Services icon. The Services dialog displays.
2. Highlight the service called MapMarker Server.
3. Choose the Startup button. Configure the MapMarker Server to run as the user who installed MapMarker. Be sure the user has permission to start a service at start up time. Click OK to return to the Services dialog.

4. Click Start. The MapMarker Server service starts.
5. To leave the Services dialog, choose Close.

MapMarker Server is now running in the background. All log information is sent to the Event Viewer (found in the Administrator) under Log > Application. As with any NT service, you can configure MapMarker Server to start whenever the NT Server is booted.

To stop the MapMarker service:

1. Open the Control Panel > Service application and highlight MapMarker Server.
2. Click STOP.

To remove the MapMarker service from the Windows NT Service Control Manager,

3. At the DOS prompt, type **mm_serve -delete**.

Configuring MapMarker Server to Run When NT Server Boots

If you wish to start the MapMarker Server whenever your NT server machine is started, you will need to modify the NT Services feature, as described below.

1. In Windows NT Control Panel, double-click on Network.
2. Under the Services tab, double-click on RPC Configuration and change the Name Service Provider to DCE Cell Directory Service.
3. In the Network Address box, enter the machine's IP address. If the machine is not on a network, leave this box blank.

Console Application

To start MapMarker Server as a console application on Windows 95 or Windows NT:

1. At the DOS prompt, type **mm_serve -console**

When the application is started, the server will send the message, "The MapMarker Engine Initialized with Database Path = [c:\mapinfo\mapmarker\data;d:\data]." If the data path is incorrect, you can change it in regedit. In Windows 95/NT 4.0 Registry Editor (regedit.exe) or Windows NT 3.51 Registry Editor (regedt32.exe), find the settings under HKEY_LOCAL_MACHINE\SOFTWARE\MapInfo\MapMarker\SYSTEM.

To stop the application:

1. From the console window, press Ctrl-C.
2. When the DOS prompt displays, type exit. The console window will close.
3. Alternatively, press Ctrl-Alt-Del and select mm_server from the Close Program dialog.

Any error information that is generated when MapMarker Server is running is displayed in the console window. There is no log file when running MapMarker Server as a console application.

For details on MapMarker's geocoding operations, refer to Chapters 3 and 4 of this abridged version.

Geocoding Request Timed Out

If MapMarker Server receives more than 1,024 requests, (i.e., all 1,024 available threads are occupied with other requests waiting for the Server), it may block your geocoding request and cause your request to time out. At that time, MapMarker Server will display an error 32104: Mutex timed out. In this case, simply resubmit your request.

This error may also occur if you have started two instances of MapMarker Server on the same machine. This commonly happens when you start the server as an NT Service and as a console application. This error will display in the NT Event View log under Log > Application.

MapMarker Client OCX

The MapMarker Client OCX is a ready-made OLE control object for MapMarker Server. In the simplest case, you can use the MapMarker OCX without any programming. The OCX is designed for any application programming environment that can call OCXs such as Visual Basic, Delphi, PowerBuilder, Oracle PowerObjects or even C++. To use the OCX you just drag and drop the control onto your form. When you run your application, the MapMarker control will appear as shown below. Through the control an end-user will be able to connect to the MapMarker Server and send geocoding requests.



MapMarker OCX uses the RPC API to communicate with MapMarker Server. The Client OCX connects, geocodes and disconnects for each geocode request (e.g., when the user clicks on the Geocode button). This ensures that the geocoding request grabs one of the MapMarker Server's 1,024 threads for the shortest time period. This reduces the time other geocoding requests will wait for a thread.

If you want more control over the MapMarker OCX or want to provide a different user interface you can drive the control via the MapMarker OLE Automation interface. Since the OCX uses the same OLE Automation interface, a direct user of the Automation interface has access to the same methods and properties of the OCX. The OLE Automation API is discussed in the next chapter.

Adding MapMarker OCX to Your Client Application

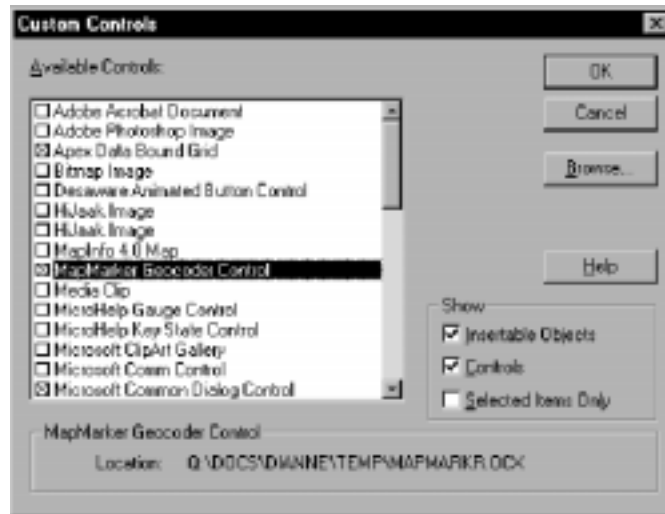
When you install MapMarker Server, the MapMarker OCX is installed and registered as well (for installation instructions, see Chapter 3). The OCX component files include:

- mapmarkr.ocx
- mapmarkr.oca

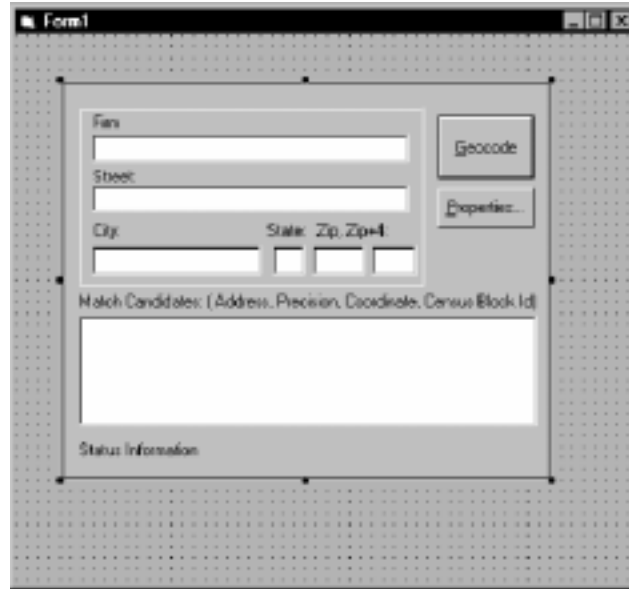
These files will be located in the OCX directory following installation.

With MapMarker OCX installed, you are ready to add it to your application. The steps are outlined below. This process assumes you are using Visual Basic as your development tool. However, any tool that supports OLE Automation can be used.

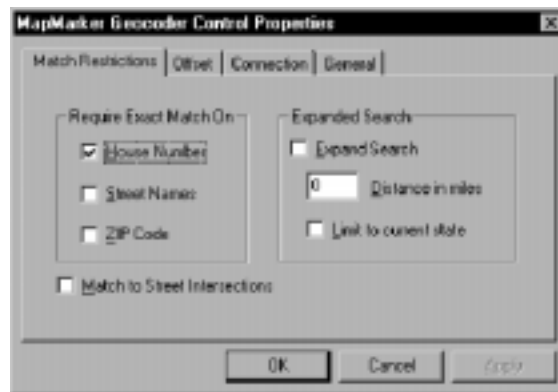
1. In Visual Basic, add MapMarker OCX to the Toolbox by choosing Tools > Custom Control or right-click on the Toolbox and select Custom Controls.



2. Check the box MapMarker Geocoder Control. Click OK. The icon for the MapMarker control displays in the Toolbox.
3. Drag and drop the control onto your design form, or double-click on the icon to add the control to the form.



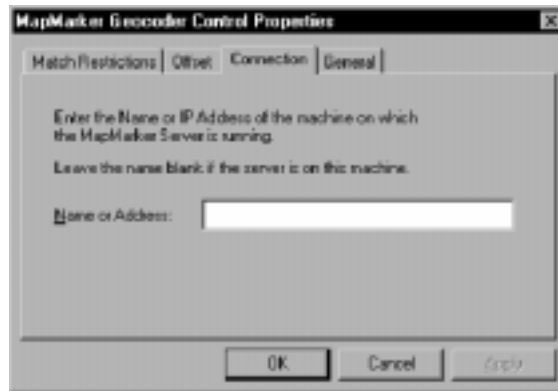
4. To set the properties, right-click on the control. The Control Properties dialog displays.
5. Set the match restrictions for house number, street name, ZIP Code, street intersection and expanded search in the Match Restrictions tab.



6. Continue to set the properties in the Offset, Connection and General tabs.
7. Choose Run > Start or click the Start button to run the application in debug mode.
8. If you wish to create an executable, choose File > Make EXE File.

Setting MapMarker OCX Properties

You can control the properties of MapMarker OCX by specifying information in the Control Properties dialog. The most important property to define is the name of the machine on which MapMarker Server is running. If this is left blank, it assumes MapMarker Server is running locally.



Properties include the following:

- Match restrictions (exact or relaxed match conditions, match to street intersection, and expand the search area if your street address is incomplete)
- Offset conditions for the geocoded point
- Name of machine on which MapMarker Server is running
- General properties where you can specify an address to geocode at design time.

Geocoding via the Client OCX

When the user runs the executable, she sends a request to MapMarker Server to geocode an address. The Server returns zero, one or more match candidates. Each candidate contains the address, match precision (e.g., street level or ZIP+4 centroid match), result code, latitude/longitude coordinates, and Census Block ID. The status line in the interface explains the type of match:

If the Server returns...	the Status line displays...
SINGLE_MATCH	Single close match
MULTIPLE_MATCHES	Multiple possible matches
NO_CLOSE_MATCHES	Candidate(s) found
NO_CANDIDATES	No candidates found
POSSIBLE_INTERSECTION	Possible intersection
NO_INTERSECTION_MATCH	Intersection was not found
SINGLE_INTERSECT_MATCH	Single intersection match
MULTIPLE_INTERSECT_MATCH	Multiple possible intersection matches
GEO_ENG_BAD_INPUT_ADDRESS	Bad Input Address

Falling Back to ZIP Code Centroids

MapMarker OCX attempts to fall back to a ZIP Code centroid match when there are no match candidates or the input address is bad. There are three situations in which MapMarker OCX will fall back to ZIP Code Centroid:

- The operator entered a ZIP Code only
- The operator entered a street and ZIP Code and MapMarker returns a Bad Input Address message
- The operator entered a street and ZIP Code and MapMarker returned a message of No Candidates.

MapMarker OCX Events and Methods

When you use the supplied MapMarker OCX interface, certain events and methods from the OLE Automation API are called. They are included here as a summary.

Event	Purpose
LatLongChanged()	This event is triggered when the Geocode button is pressed, the DoGeocode() method is called, or when a different candidate is selected in the Match Candidates list box. This event updates the latitude, longitude and precision properties.
GeocodeEvent()	This event is triggered when the Geocode button is pressed and the whole geocoding process is completed. Note: The DoGeocode() method does not trigger this event.

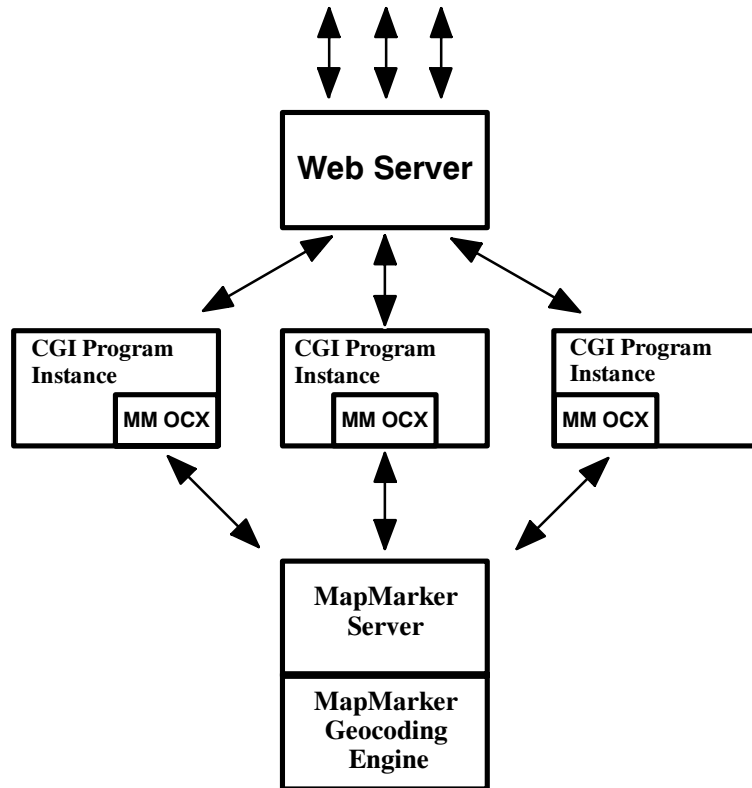
Method	Purpose
ClearDialogText()	Clears all visible text in the dialog, as well as these properties: firm, street, city, state, Zip, ZipPlus4, lastErrorCode, longitude, latitude, and precision.
RefreshDialog()	Tells the dialog to repaint. MapMarker Client Control will usually repaint automatically and you will not need to call it.
DoGeocode()	This method is the same as pressing the Geocode button. It implies that you must initialize these properties: firm, street, city, state, Zip, ZipPlus4, serverName.
GetCandidateAt()	Returns the line of text in the Match Candidate list box.
GetCandidateStreetAt()	Returns the street portion of the address as a string.
GetCandidateCityAt()	Returns the city portion of the address as a string.
GetCandidateStateAt()	Returns the state portion of the address as a string.
GetCandidateZIPAt()	Returns the ZIP portion of the address as a string.
GetCandidatePlus4At()	Returns the ZIP add on of the address as a string.
GetCandidatePrecisionAt()	Returns the precision for the match as a string. Precision refers to the quality of the match: to street level, shape path, intersection, or ZIP centroid.
GetCandidateLongitudeAt()	Returns the longitude as a double.
GetCandidateLatitudeAt()	Returns the latitude as a double.
GetCandidateCensusBlockIDAt()	Returns the Census Block ID as a string.

Method	Purpose
DoSetProperties()	Brings up a set of property pages for the Client Control.
SelectCandidateAt()	Highlights the specified candidate in the Match Candidate List box.
GetLastErrorCode()	Returns the last error code (same as the LastErrorCode property).
GetStringBinding()	Returns a string that describes the connection to the MapMarker Server (same as StringBinding property).
GetName()	Returns the application name.
GetFullName()	Returns the full name and path of the application.

The Toolkit and the Internet

MapMarker Client/Server Toolkit is an excellent way to create geocoding applications via the Internet. The most common configuration for use of the Toolkit and a web server is to create a Common Gateway Interface (CGI) program that uses the MapMarker Client OCX. The MapMarker Client will then call the MapMarker Server. As your web server gets geocoding requests it will execute the CGI program for each request. The MapMarker Server will get the requests via the MapMarker OCX, queue them and then serve each one in turn.

Another solution is to use the MapMarker geocoding engine by itself via a web server API such as the Netscape API or the Microsoft Internet Server API. This alternative will use less system resources when processing geocoding requests. The disadvantage is that it requires more programming as well as a good understanding of multi-threaded programming.



Appendix A: Geocoding Terminology

Here are a few terms that you should understand and become familiar with before working with MapMarker.

Address Columns

Fields in a table containing address information. In MapMarker you must specify which columns in your table contain the addresses on which MapMarker will attempt to match. Address columns also include output columns where MapMarker will store longitude and latitude coordinates and result codes information as a result of a match.

Automatic Geocoding

Records in your table are matched against an Address Dictionary automatically.

CASS

Coding Accuracy Support System. CASS was created by the U.S. Postal Service as a mailing standard for bulk mailing. It offers postage discounts to organization who can prove to the USPS that their mailing addresses comply with CASS requirements. All address lists used to produce mailings for automation rates must be matched and coded to current USPS directories using CASS-certified software, of which MapMarker 3.0 is one.

CD-ROM Unlocking

Technology included with the MapMarker installation program that allows you to access encrypted data stored on the CD-ROM.

Census Block ID

The 15-digit code that defines the U.S. Census Bureau's smallest unit for census information reporting. The code represents the state and county FIPS code (Federal Information Processing Standard), the Census Tract, Block Group and Census Block.

Check Digit

Part of the delivery point barcode on U.S. mail pieces, along with the Delivery Point Code. This address component is a requirement if the mailing table is being considered for CASS certification.

Datum

Geographic reference point on which the coordinates for longitude and latitude in the North American are based.

Delivery Point Code

A 2-digit code that together with Check Digit is used to form the delivery point barcode on mail pieces. This code is a required output component in MapMarker for those tables that are being geocoded and address standardized for CASS certification.

Fallback

To relax a geocoding condition to the next level. For example, fallback to ZIP Centroid means that if MapMarker cannot make a match at the street level, that it should match the record to the appropriate ZIP Code centroid.

Firm Name

Firm records is a USPS designator given to organizations that receive a significant letter mail volume as determined by the local postal district office. Not all organizations have firm records, and firm records do not contain individual names at residential addresses. Firm records may also have a range of ZIP add-on code assignments. This is a required output field for CASS address standardization.

Geocode

The process of matching an address record against a search table of addresses containing geographic coordinates. Once geocoded, that record has coordinates associated with it and can be spotted on a computer map at or near the location of the address (depending on the geographic precision of the match).

Interactive Geocoding

For records that you cannot match automatically, you decide whether there is a match by choosing to accept or ignore a potential match from a list of suggestions.

License Access Code

An unlocking code that allows MapMarker to access the data in the Address Dictionary. This code is required to install MapMarker and is contained on the data CD-ROM or may be obtained by calling the MapMarker License Authorization Desk during installation.

MapMarker Address Dictionary

A search table of address information that is used for geocoding records. In MapMarker 3.0, the Address Dictionary includes information from three sources: address and ZIP+4

information from the U.S. Postal Service ZIP+4 database (vintage February 1997); street geometry and related fields from the U.S. Census Bureau's TIGER 95 files (vintage 1995, release date September 1996); and ZIP+4 centroids from GDT, Inc. (vintage December 1996).

Match

To compare two addresses to determine whether they are the same. In MapMarker, a match does not necessarily mean an exact match, but uses a variety of matching rules to make a judgment that the addresses are the same.

Metadata

Information stored in the table that describes the data itself. MapMarker 3.0 reads the metadata for a table to be geocoded to determine what settings exist for address columns, geocoding and system preferences, if any.

NAD27

North American Datum of 1927. The old datum standard for coordinates from the U.S. Census Bureau prior to TIGER 95.

NAD83

North American Datum of 1983. The new standard for coordinate data from the U.S. Census Bureau's TIGER 95 files. TIGER 95 is one of the sources of the MapMarker 3.0 Address Dictionary.

ODBC

Open Database Connectivity interface. Consists of a driver manager and a set of ODBC drivers that enables an application to access remote data using SQL as a standard language. In MapMarker, ODBC allows users to geocode remote tables. MapMarker supports remote tables in the following formats: Informix, Oracle, Oracle - SpatialWare, MS Access, and Sybase.

Ranges

Address ranges that characterize a street. A range shows the house numbers that cover a particular side of the street - generally, odd numbers on one side of the street, even on the other.

Result Code

MapMarker returns a code for each record describing the results of the geocoding operation. The code represents whether a match was made and the quality of match.

Shape Path

The shape of a street as defined by points that make up each segment of the street. In MapMarker, a record is geocoded to the centroid of the shape path if the matching street address does not contain address ranges.

Urbanization (Urb)

Urbanization is a USPS designation for an area, sector, or development within a geographic area. This URB descriptor, commonly used in Puerto Rican urban areas, preceded the name of the area, is an important part of the addressing format of Puerto Rico, as it describes the location of a given street. This is a required output field for CASS address standardization.

User-Defined Address Dictionary

A search table of address information that you create from a MapInfo table containing geographic objects, such as streets. MapMarker provides a walk-through wizard to create this customized matching dictionary.

ZIP Code Centroid

ZIP is an acronym for Zone Improvement Plan. A ZIP Code is a five-digit code that identifies a specific geographic delivery area. ZIP Codes can represent an area within a state (an area that may or may not cross county boundaries), an area that crosses state boundaries (an unusual condition), or a single building or company that has a very high mail volume. The ZIP Code Centroid is the center point of the ZIP Code boundary.

ZIP+2 Centroid

Center point of the mail delivery area that is more precise than a 5-digit ZIP Code area.

ZIP+4 Centroid

Center point of a ZIP+4 mail delivery location. ZIP+4 is the smallest USPS delivery area, usually covering no more than a street block. It also refers to single mail-drop locations for businesses that receive high volumes of mail.

Appendix B: Append Utility

Append is a utility program included in MapMarker that combines two or more MapInfo tables into a new table or adds them to an existing table. These tables must have the same number of columns and data types for each column. We recommend using Append for making one large MapInfo table when creating MapMarker User Dictionaries.

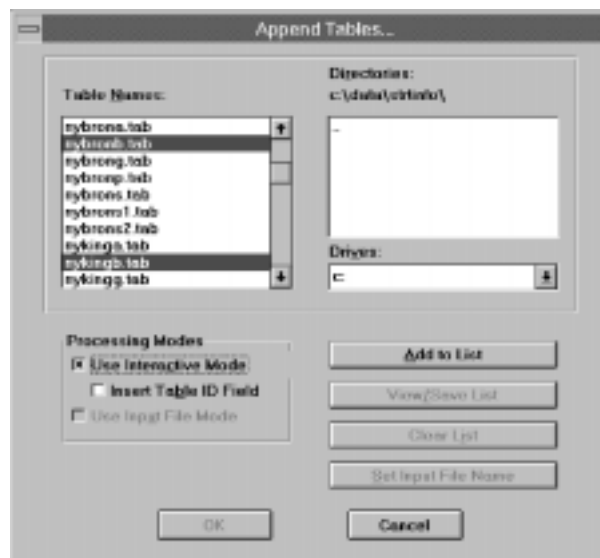
Append runs from within MapInfo as an add-on application. It displays a series of dialogs where you specify the starting drive and directory of the search and select any available options. Append.mbx is located in the mbx directory after installing MapMarker.

If you are running the 16-bit version of MapInfo for Windows (3.x or 4.x), Append uses two files, append.mbx and iputil16.dll. These two files must be in the same directory. For users of 32-bit MapInfo (4.x) append.mbx and iputil32.dll must be in the same directory.

Running Append

To run the Append program from within MapInfo:

1. In MapInfo, choose File > Run MapBasic Program. The Run MapBasic Program dialog displays.
2. Choose APPEND.MBX from the directory where MapMarker is installed.
3. Click OK. The DataKit menu is added to the menu bar.
4. Choose DataKit > Append > Append Tables to display the Append Tables dialog, as shown below:



5. Choose a processing mode. Select Use Interactive Mode to choose individual groups of tables to append. The tables will be placed in a queue for Append to concatenate. If you want Append to use a text file to determine what tables it should append and how it should append them, choose Use Input File Mode. For more instructions on Interactive Mode and Input File Mode see the sections following these steps.
6. Click the OK button in the Append Tables dialog to start processing.
While Append runs, the program appends tables and writes out the new appended table with or without the table ID field, depending on the options you have chosen. When the program finishes, Append records a summary of the results in the message file and a dialog displays the elapsed processing time. You may open the message files in any text editor such as Notepad.
7. When you have finished, choose DataKit > Append > Exit Append to close the program. If the Append command is the last item left in the DataKit menu, exiting the program will also remove the DataKit menu.

Interactive Mode

Interactive Mode allows you to specify each process one at a time. This allows you to build an interactive list of tables that Append will process. Follow these guidelines to use Append in interactive mode. Follow these steps for each group of tables you wish to append.

1. Click on the tables you want to select in the Table Names list. You may change the directory and drive to display other tables. Keep the following in mind when choosing your tables:
 - Append will not let you concatenate tables if the numbers of columns in all of the tables are not the same. The program will return you to the Append Tables dialog box if you attempt to do this.
 - If the number of columns in all of the tables is the same but they do not share the same data types between column numbers, Append will try to resolve the differences in the column types. When this happens, Append may change some of the data types of your columns. If there are changes, Append records these changes in the message file. If Append cannot resolve the data type differences, the program returns you to the Append Tables dialog.
 - If the number and types of columns in the appended tables are the same but the names are different, Append names the resulting columns according to the following rules. If you are appending tables to an existing table, the column names of the existing table will not change. If you are creating a new table with Append, the table will have the same name as the first table that you selected in the Table Names list.
2. To insert a table identifier field into the output table, check the Insert Table ID field box. The table identifier field contains the source table name for each row.
3. Click on Add to List. Append prompts you for the output name of the appended table. Enter the table name and click OK.
4. Click on View/Save List to inspect or save the current interactive list. You may want to do this if you intend to process the same tables later in Input File Mode. To remove all the items in the current interactive list, click Clear List.

Input File Mode

Input File Mode allows you to specify an input text file that Append uses to process many tables without having to add them into a list one at a time. The input file is a formatted file that contains command lines for Append. The command lines list the files that will be joined and tell Append what options to use when processing. This method is most convenient when you have a large number of tables that must be appended and you do not want to go through the tedious process of selecting the tables through the graphical interface. Below is the format of the command line that you must use and some examples that use the format.

```
append <input_table_name_list> [/table-id] /output <output_table_name>
```

```
E.g., append c:\data\test1.tab test2.tab /table-id /output newtest.tab
```

```
append test1.tab test2.tab /output c:\data\newtest.tab
```

The first example creates a table called newtest that is a concatenation of two other tables: test1 and test2. This example also creates a new column in the newtest table (Table_ID) that will hold the source table for each record.

The second example is similar to the first except that Append will not create a Table_ID column; the paths for the source and output tables are also different.

To use Append in Input File Mode, do the following:

1. Check the Use Input File Mode box.
2. Click the Set Input File Name button to specify the name of the input file that Append will use.

The APPEND.MSG File

Append produces a message file for each processing session that contains information about the tables that were appended. Append names the message file APPEND.MSG and places the file in the same directory as the output tables. If the file APPEND.MSG already exists, Append will add the new information to the end of the existing file.

Considerations When Running Append

If you append a file that currently is open, Append will close the table before the program processes it. If the table has pending edits, Append will bring up a dialog that asks you whether you want to save or discard those edits before appending.

If you are using input file mode, be sure that none of your command lines conflict or overwrite files produced by a previous line in your input file.