

5

Chapter

Mapping Concepts

Now that you have installed MapXtreme and seen the wide variety of features and possibilities, you are probably anxious to get mapping. But, first, take a few minutes to read this chapter, especially for those of you new to MapXtreme Java and MapInfo products. This chapter gives you an understanding of the concepts for successful mapping with MapXtreme.

- **Organizing Your Data and Maps: An Overview of Tables**
 - **MapInfo Geosets**
 - **Map Definitions**
 - **Map Features**
 - **Renditions**
 - **Labeling**
 - **Map Data Analysis**
 - **Putting Your Data on the Map**
 - **The Power of MapXtreme**
-



Organizing Your Data and Maps: An Overview of Tables

To use MapXtreme Java, you need the files that contain your records and maps. Your data can be in MapInfo format or in a format that supports spatial data. This chapter will explain the MapInfo file formats and introduce you to Map Definitions, XML-based text files that can store MapInfo data or data from your JDBC database.

How Files Make Up a Table

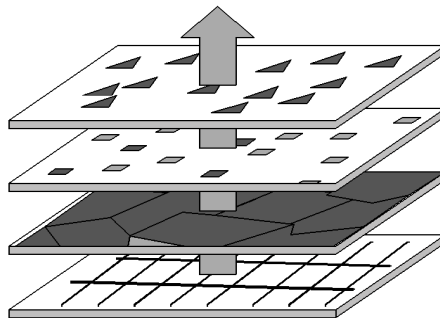
MapInfo .tab files are organized into a group of files to build a map layer. MapInfo tables consist of the following associated files:

- *Somefile.tab*: This file describes the structure of the MapInfo table. It is a small text file describing the format of the file containing the data.
- *Somefile.dat*(.mdb, .dbf, .txt, .xls or .wks): These files contain the tabular data.
- *Somefile.map*: This file describes the graphic objects (it will not exist if the table has no map objects).
- *Somefile.id*: This file is a cross reference file that links the data with the objects (it will not exist if the table has no map objects).
- *Somefile.ind*: This is an index file. The index file allows you to search for map objects (will not exist if the table has no index).

Together these files make up a single layer in MapXtreme Java.

MapInfo Tables and MapXtreme Layers

Each mappable MapInfo table can be displayed in a MapXtreme application as a layer in a map. For example, you can display a layer of customers, a layer of streets, and a layer of county boundaries.



Think of these layers as transparencies where each layer contains a different part of the map. The layers are stacked one on top of the other and allow you to see all aspects of the map at the same time. Layers are drawn from the bottom up. They are listed in the Layer Control dialog with the topmost layer listed first.

MapInfo Geosets

A collection of .tab files is called a geoset. Geosets are similar in concept to a workspace. Instead of opening individual .tab files, you can open the geoset (*somefile.gst*) and all the layers with their particular settings display. MapXtreme Java ships with a wide variety of sample geosets covering world geographic regions. Use these geosets to get familiar with mapping with MapXtreme Java. For a complete description of the sample geosets and geoset metadata, see Appendices F and G.

Geosets are limited in their functionality in MapXtreme Java, however. They specifically refer to MapInfo .tab files which cannot be saved to a remote database. This release of MapXtreme Java provides an alternative, that is far superior to geosets. Called Map Definitions, they are XML-based text files that contain map layer information that can be stored as a file or as a record in a JDBC database. More on Map Definitions is in the next section.

MapXtreme cannot open a MapInfo workspace (.wor file type). If you are using MapInfo Professional, you may save your workspaces as geosets using the MapInfo Geoset Utility, then load them into MapXtreme Java using the Map Definition Manager.

Map Definitions

Map Definitions are not new to MapXtreme Java, but their format is new. In previous releases, Map Definitions were stored as serialized Java objects in a binary format. By changing to XML-based text in this release, Map Definitions are now more forward compatible and easily editable.

When a Map Definition is loaded, MapXtreme automatically reads the metadata in the Map Definition and sets the display properties. The developer can change the default display to meet her individual needs. A Map Definition's settings include projection, default zoom, auto-labeling of objects, zoom layering, and whether the table is visible when opened.

Map Definitions are created using the Map Definition Manager that ships with MapXtreme Java. Instructions are provided in Chapter 14: Managing Your Data. When saved, they can be stored as a file, using the extension .mdf, or stored as a record in a JDBC database.

Map Definitions created in previous versions of MapXtreme will not work with this release of MapXtreme Java. You will need to create new Map Definitions either by using the Map Definition Manager or programmatically through the new MapDefContainer interface, explained on page 239.

Because you have more flexibility and control when working with Map Definitions, we strongly recommend that you save your map settings as Map Definitions rather than geosets. You can even turn your geosets into Map Definitions. The Map Definition Manager provides a Save As feature.

To get familiar with maps and MapXtreme, start by opening a sample geoset in Map Definition Manager using the command Map > Load Geoset. Experiment with different layer settings (described in Chapter 14). When you are ready to save your work, save it as a Map Definition file (extension .mdf) or as a table or query in a remote database.

Map Features

We mentioned earlier that maps in MapXtreme are made up of layers of map objects. These map objects are accessed in MapXtreme through the Feature object. There are three basic types of features:

- **Regions:** closed objects that cover a given area. These include polygons, ellipses, and rectangles. Regions include country boundaries, postal code boundaries, sales territories, among others.
- **Point objects:** represent single locations of data. Some examples include, customer locations, restaurants, parking meters.
- **Line objects:** open objects that cover a given distance. These include lines, polylines, and arcs. Examples are streets, rivers, power lines.

You can have each type of object in a separate layer (most common), or you can combine objects in the same layer. MapXtreme lets you customize and display these objects to make maps that meet your needs.

Renditions

All features in a MapXtreme Java map display with certain visible characteristics, such as the color or symbol type. You have complete control over how you would like your map features to look. This release of MapXtreme Java includes many new display properties to take better advantage of the rendering capabilities of the Java2D API, among them symbol paint for lines and regions, dashed and parallel lines, and vector symbols.

Renditions can be set programmatically or through the Map Definition Manager's Layer Control dialog. See Chapter 12 for the Rendition API. See Chapter 14 for more on the Map Definition Manager.

Labeling

Related to renditions in that it is a display characteristic, labeling is a powerful feature in its own right to enhance your map and impart the proper message to the viewer. Labeling is more than just adding text to describe the map feature. With MapXtreme Java you can control the font type, size, color, position, and use creative effects such as haloing and outlining, to create labels that are distinctive for each layer of your map.

Labeling is covered in Chapter 12: Labeling and Renditions.

Map Data Analysis

A powerful feature of mapping with MapInfo Corporation's family of mapping software is the ability to analyze the map to learn more than you could if you only saw the data in row and column format. By displaying the data on a map you can make visual comparisons of your data which lead to better business decisions.

MapXtreme Java provides a number of ways to analyze the map, from using map tools to click on features, to creating thematic maps that display relationships among the features, to searching underlying data according to your own criteria.

Much of this Developer's Guide provides information on how to get the most out of your data. See Chapter 12 to learn more about searching map layers. Chapter 13 covers the types of thematic maps you can create in MapXtreme Java. Chapter 14: Managing Your Data, provides a discussion on map tools. Map tools are also covered in the MapXtreme JavaBeans chapter.

Putting Your Data on the Map

Data Providers enable you to add data from other sources to your maps. For example, if you have an Oracle8i database of sales by county, you could open that data and display it on a map in order to spot trends in sales patterns by county not easily highlighted in the tabular data alone.

There are many different types of databases in businesses today, therefore, MapXtreme lets you access several different types of data sources. You can access:

- Oracle8i
- SpatialWare for Oracle
- Informix Universal Server SpatialWare DataBlade
- DB2 SpatialWare Extender
- JDBC Compliant databases (spatial data stored in X,Y columns)

For information on adding data from these data sources, see Chapter 9.

The Power of MapXtreme

Now that you have an overview of tables, layers, Map Definitions, map features, and data sets, you are ready to bring the full capabilities of MapXtreme Java into play. With MapXtreme, you can search a layer in a map and locate a specific feature within the layer. For instance, you want to find the closest dealer to Lackawaxen, PA. Or, you need to calculate distances between health care providers and their patients and get counts on how many patients live within a given radius of a particular hospital. Or, you want to thematically shade boundaries (counties, towns, states, countries) according to the number of customers who purchased from you within the last year.

If you would like to take advantage of some of the pre-built components included in MapXtreme Java, to jumpstart your application development, see the next two chapters on MapXtreme JavaBeans and Writing Your Own Servlets. To begin programming using the MapJ API, start at Chapter 8: MapJ. Other elements of the API are covered in chapters 9-13.

Be sure to review Chapter 14: Managing Your Data, for a full discussion on creating Map Definitions and controlling the display and behaviour of your map layers with the Map Definition Manager.