

MapX 4

Руководство разработчика
русская версия

MapInfo Corporation
Troy, NY

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.

©1992-1999 MapInfo Corporation. ALL RIGHTS RESERVED.

MapInfo Help ©1992-1999 MapInfo Corporation. ALL RIGHTS RESERVED.

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

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

MapInfo Corporate Headquarters: MapInfo Europe Headquarters:

Voice: (518) 2856000

England

Germany

Fax: (518) 2856060

voice: +44 (0)1753 848 229

voice: +49 6196 6700 0

Sales Info Hotline: (800) 3278627

fax: +44 (0)1753 621 140

fax: +49 6196 6700 11

Federal Sales: (800) 6192333

Technical Support Hotline: (518) 2857283

Technical Support Fax: (518) 2856080

For international customers, please use the Technical Support Fax number.

WARNING: This software uses patented LZW technology for .GIF image compression and/or decompression. (Unisys United States patent No. 4,558,302 and corresponding patents in Canada, France, Germany, Italy, Japan and the United Kingdom). GIF images compressed or decompressed for transmission via the Internet or via any other on-line communication capability may not be sold or licensed for revenue, or used by an Internet Service Provider or in paid advertisements unless the user first enters into a written license agreement with Unisys. For information concerning licensing, please contact: Unisys Corporation Welch Licensing Department C1SW19 Township Line & Union Meeting Roads P.O. Box 500 Blue Bell PA 19424 Fax: 215-986-3090

HAHTsite is a registered trademark of HAHT Software Inc. in the United States.

Portions of the data are the proprietary information of Roadnet Technologies, Inc., a United Parcel Service Company, and are Copyright 1993. Roadnet Technologies, Inc.

Monotype and Century Gothic are trademarks of Monotype Topography Limited registered in the U.S. Patent and Trademark Office and certain other jurisdictions.

Portions of this publication are Copyright 1998, HAHT Software Inc. All Rights Reserved.

Portions of the software are derived from the *Standard C Library*, ©copyright 1992, by P.J. Plauger, published by Prentice-Hall, and are used with permission.

HyperHelp copyright© Bristol Technology Inc. 1991, 1992, 1993

HIL©Media Cybernetics, Inc. 1993, Halo Imaginf Library is a trademark of Media Cybernetics, Inc.

EHelp® is a registered trademark of Foundation Solutions, Inc., ©copyright, 1992, 1993.

MrSID® is a trademark of LizardTech, Inc. and is used under license.

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.

This documentation reflects the contributions of almost all of the women and men who work for MapInfo Corporation. It was specifically produced by Tony Maritato and Max Morton , with the help of Jim Regan, David Smith, Brian Bloniarz, and Larry Strianese. Colleen Cox, Editor. These members of the Documentation Department are indebted to MapInfo's Quality Assurance Department and, of course, to all the members of the Engineering team who labored on this project.

MapInfo welcomes your comments and suggestions.

MapX v4.0
September 1999

Содержание

Глава 1: Введение	1
Введение.....	1
Знакомство с Картографией.....	2
Заставьте MapX работать на Вас.....	3
Обзор основных возможностей	4
Для кого эта книга	6
Глава 2: Начинаем работу с MapX	8
Начинаем работу с MapX.....	8
Что такое MapX?.....	9
Системные требования	9
Что содержит пакет MapX.....	9
Комплектность MapX.....	10
Установка MapX	12
Подключение компоненты Map	14
Для пользователей Visual Basic.....	14
Обновление приложений C++ использующих более ранние версии MapX.....	16
Работа с Visual Basic	17
Работа с Visual C++	25
Работа с PowerBuilder	26
Работа с Delphi	27
Работа с Lotus Notes.....	28
Документация MapX	33
Что читать дальше.....	36
Глава 3: Новое в MapX.....	37
Новое в MapX.....	37
Новое в MapX 4.0.....	38
Улучшения и добавления в MapX 3.5.1	47
Улучшения и добавления в MapX 3.5	49
Улучшения и добавления в MapX 3.0	54
Обновление приложений MapX 2.0 до MapX версии 3	58

Глава 4: Основы картографии	61
Основы картографии.....	61
Организация Ваших данных и карт: Обзор таблиц	62
Что такое GeoSets?	64
Объекты карты.....	72
Помещаем Ваши данные на карту	73
Мощь MapX.....	76
Глава 5: Основы MapX.....	77
Основы MapX.....	78
Объект Map	78
Страница свойств.....	80
Слои	82
GeoSets.....	84
DataSets	89
Аннотации	90
Создание объектов.....	92
Глава 6: Картографирование по слоям	94
Картографирование по слоям	94
Карта как набор слоев	95
Коллекция слоев: строительные блоки Вашей карты	96
Некоторые свойства коллекции слоев.....	99
Некоторые методы коллекции слоев	101
Объект Layer.....	107
Порядок слоев	109
Информация о слоях.....	110
Проверка типов объектов на слое.....	111
Масштабный эффект.....	112
Создание подписей для слоя	115
Аннотации	119
Растровые изображения	120
Анимационный слой.....	121
Собственные слои пользователя.....	123
Глава 7: Объекты и выборки.....	125
Объекты и выборки	125

Использование коллекции Features	127
Коллекция Selection	133
Редактирование объектов.....	137
Глава 8: Поиск объектов на карте	147
Поиск объектов на карте	147
Объект Find	148
Объект FindFeature	150
Глава 9: Инструменты	153
Стандартные инструменты.....	154
Создание собственных инструментов.....	158
Создание инструмента рисования полигона.....	164
Глава 10: Размещение данных на карте	167
Размещение данных на карте	167
Что такое связывание данных?	168
Преимущества помещения данных на карту.....	170
Как добавлять данные к карте.....	171
Объект DataSet.....	172
Коллекция DataSets.....	173
Метод DataSets.Add.....	174
Метод Fields.Add	181
Отображение Ваших данных в виде слоя точек.....	185
Превращение нового слоя с точками в постоянный слой	192
Использование геословаря при связывании данных	193
Различные типы источников данных.....	197
Глава 11: Доступ к данным СУБД (DBMS)	208
Доступ к данным СУБД (DBMS)	209
Доступ к удаленным пространственным данным	210
Применение метода Layers.Add с использованием объектов LayersInfo	212
Доступ к удаленным таблицам через .tab файл.....	213
Создание точечных объектов для данных СУБД по значениям в колонках X/Y	215
Поддержка Oracle8i.....	217

Параметры DBMS LayerInfo.....	221
Доступ к атрибутивным данным	225
О производительности	227
Кэширование	228
Каталог карт MapInfo.....	229
Загрузка пространственных данных в СУБД	230
Создание каталога карт MapInfo вручную	231
Геокодирование таблицы DBMS.....	233
Синтаксис определения стиля Symbol, Pen, Brush	236
Устранение неисправностей	238
Глава 12: Тематическая картография и анализ	240
Тематическая картография и анализ	240
Что такое тематическая картография.....	241
Разработка собственной тематической карты	243
Типы тематических карт	251
Работа с тематическими картами	271
Настройка тематической легенды	275
Глава 13: Использование систем координат	276
Использование систем координат.....	276
Концепции систем координат	277
Получение объекта Coordinate System	278
Запрос свойств объекта CoordSys.....	279
Отображение Карты в разных системах координат.....	280
Определение координат X-Y в разных системах координат.....	282
Диалог Выбор Проекции	284
Использование настроек из MAPINFOW.PRJ	285
Применение аффинных преобразований.....	290
Определение собственных картографических регионов.....	293
Преобразование картографического региона	297
Дополнительная информация.....	298
Глава 14: Использование сборочных слоев	299
Использование сборочных слоев	299
Что такое Сборочный слой?	300
Термины и Концепции	303

Как делать приложения со сборочными слоями	306
Подготовка сборочного слоя	308
Создание инструмента Сборки	313
Переустановка сборочного слоя	328
Ограничения и требования для сборочного слоя	329
Дополнительная информация... ..	330
Глава 15: Экспорт карт	331
Экспорт карт	332
Свойство ExportSelection	334
Печать карт	335
Глава 16: Работа с Visual C++	336
Работа с Visual C++	336
Знакомство с примерами приложений	337
Обновление приложений на C++ из ранних версий MapX	338
Доступ к свойствам и методам MapX в C++	339
Включение файла MapX.cpp в Ваш проект	344
Создание управления MapX, с помощью C++	345
Создание разделов меню с помощью C++	347
Обработка событий MapX с помощью C++	350
Применение пользовательских инструментов (C++ пример)	353
Связывание данных с использованием C++	356
Добавление быстрого меню с использованием C++	360
Использование встроенных диалогов справочной системы из C++ ..	362
Обработка исключительных ситуаций MapX с помощью C++	364
Создание карты в диалоге C++	366
Глава 17: Распространение Ваших MapX приложений	371
Распространение Ваших MapX приложений	371
Установка MapX приложений	372
Установка MapX ОСХ	373
Установка обработчиков растровых форматов	378
Установка карт и наборов данных Geosets	381
Добавление ключей в реестр Windows	382
Передача строки лицензии в MapX	384

Приложение А: Управление данными MapX.....	387
Приложение В: Утилита Geoset Manager	398
Приложение С: Использование утилиты Geodictionary Manager	418
Приложение D: Поддержка собственных наборов данных Dataset	430

Введение

- ◆ Знакомство с Картографией
- ◆ Заставьте MapX работать на Вас
- ◆ Обзор основных возможностей
- ◆ Для кого эта книга

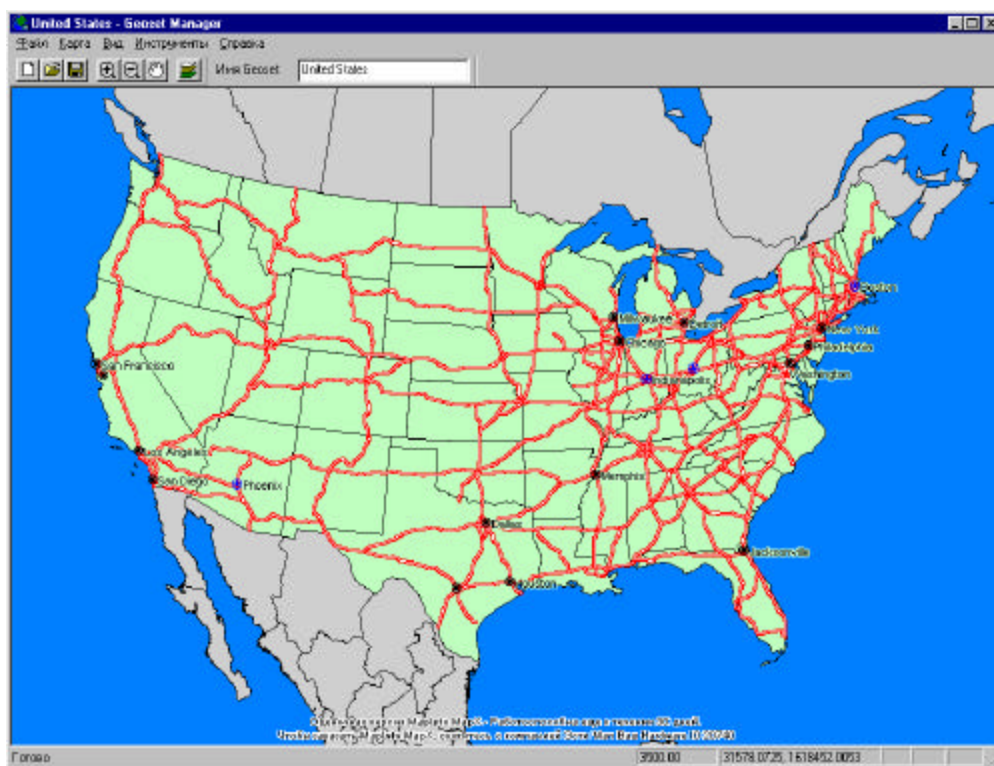
Добро пожаловать в семейство продуктов MapInfo. Поле применения электронной картографии продолжает расширяться и MapInfo продолжает создавать новые продукты, дополняющие основной продукт корпорации - MapInfo Professional.

MapInfo MapX это картографический ActiveX (OCX) компонент, который позволяет легко добавить мощные картографические функции в Ваши приложения. Используя карты, Вы можете отображать информацию в виде, легко понятном каждому. Карты более информативны, чем диаграммы и графики, и их интерпретация более наглядная и быстрая по сравнению с таблицами.

MapX основан на тех же картографических технологиях, которые используются в других продуктах MapInfo, таких как MapInfo Professional и Microsoft Map. Если Вы уже создавали или приобретали карты в формате MapInfo для использования в MapInfo Professional или Microsoft Map, Вы можете использовать эти же файлы с MapX.

Знакомство с Картографией

Сегодня доступно огромное количество информации, гораздо больше, чем раньше. Данные накапливаются в виде таблиц, записей о продажах и маркетинговых файлов. Бумаги и компьютеры хранят множество информации о покупателях, магазинах, персонале, товарах и ресурсах.



Почти вся эта информация имеет географическую составляющую. По оценкам 85 процентов информации в базах данных содержит некоторый вид географической информации, такой как улицы, города, районы, почтовые индексы, телефонные номера с кодами территорий и т.п.

МарХ может помочь отсортировать всю эту информацию и, используя географическую составляющую в Ваших данных, отображать их на картах. Это позволит увидеть тенденции и взаимоотношения между различными объемами информации быстро и надежно.

Заставьте МарХ работать на Вас

С помощью МарХ возрастает мощь картографических возможностей в ваших приложениях. Вы можете отобразить Ваши данные как точки, тематически раскрашенные области, круговые и столбчатые диаграммы и др.. Применяйте аналитические объекты МарХ при группировании и организации данных, поиске или выборке на карте в заданных областях.

Например, МарХ может показать какой из сети магазинов ближе всего к Вашим самым крупным покупателям. Программа может рассчитать расстояние между покупателями и магазинами; может показать покупателя потратившего больше всего денег за последний год; можно разными цветами представить объемы продаж магазинов. Все это вместо и есть визуальное представление ваших данных на карте.

Обзор основных возможностей

MapX это больше чем "просмотрщик карт". Используя MapX, Вы можете анализировать и отображать на карте Ваши данные, создавать или редактировать картографические объекты и отображать результаты географически. Основные функциональные возможности MapX описаны ниже.

- **Тематические карты** Визуализация Ваших данных на тематических картах. Связь данных с каждым объектом на карте, использование цветового и стилевого тематического оформления для визуального представления данных. С помощью тематической картографии можно увидеть данные в шести типах тематических карт (цветные диапазоны, плотность точек, индивидуальные значения, размерные символы, круговые диаграммы или столбчатые диаграммы).
- **Сборочная картография** Подведите курсор к интересующей Вас области и нажмите кнопку мышки - этот регион отобразится более подробно. Для OLAP/DSS, можно позволить пользователю просматривать сборочные слои.
- **Связывание данных** Ваша карта может инкорпорировать данные из контейнера в который встроен ODBC, источники данных ODBC или DAO, такие как MSAccess. MapX имеет несколько различных типов привязки данных к карте, включая уровень кодирования ZIP Code (почтовые индексы).
- **Аннотации** Обеспечивается ориентация, выделение указанных данных, и Ваши карты будут более информативными с добавлением текста, символов и подписей.

- **Слои** Отображение и управление слоями, видимый=невидимый, редактируемый-нередитируемый и масштабный эффект отображения слоя. Возможно использовать сшитые слои из нескольких обычных таблиц. Специальные типы приложений поддерживаются для специальных типов слоев, таких как Слой Анимации (для контроля движущихся объектов в режиме реального времени) и Косметический (UserDraw) слой (для рисования специализированных картографических элементов, таких как логотип).
- **Растровые изображения** Имеется возможность добавить растровую подложку, чтобы сделать карту более привлекательной и информативной.
- **Автоматическое подписывание** Автоматически подписывает Вашу карту, с настройкой их атрибутов и настройкой изображения.
- **Выборки** MapX усилен аналитическими объектами для группирования и организации данных. Выбор объектов карты в указанных областях (в круге, в прямоугольнике и др.).
- **Feature Factory** Объект FeatureFactory позволяет Вам создавать, комбинировать, создавать буферные зоны, или стирать точки, линии и полигоны (объекты).
- **Инструменты** Ваш пользователь сможет работать с картой непосредственно, с помощью мышки и переноски. Используйте встроенные инструменты MapX навигации, выбора, подписывания или создания пользовательских инструментов, что бы усовершенствовать Ваше приложение.
- **Редактирование карт** Вы можете дать Вашим пользователям возможность добавлять, модифицировать или удалять объекты карты.
- **Проекции и координатные системы** Полная поддержка систем координат и проекций карт, MapX позволяет тонко настраивать отображение карт и преобразовывать X-Y данные в координаты.

- **Связь с удаленным сервером пространственных данных**
“Живое” соединение возможно в Oracle8i Spatial и MapInfo SpatialWare запускаемым под Oracle 8.0.5, Informix или другой поддерживаемой базой данных SpatialWare. Пространственные серверы позволяют хранить на нем картографическую информацию, что крайне удобно для безопасности и экономии машинных ресурсов. Пространственные серверы типа SpatialWare и Oracle8i Spatial предлагают совершенный механизм обработки запросов и увеличивают преимущества сервера в организации пространственных данных. Хранение пространственных данных в RDBMS так же необходимо для приложений, которые работают с редактированием карт и большими объемами данных.

Для кого эта книга

Это книга для всех кто хочет добавить картографическую функциональность к их приложениям с использованием MapX и объектно-ориентированных языков программирования типа Visual Basic, Visual C++, PowerBuilder, Delphi, или в Lotus Notes используя LotusScript.

Изучение MapX

Если Вы уже использовали более ранние версии MapX, посмотрите главу: *"Что нового в MapX"*.

Если Вы используете MapX первый раз, посмотрите как обучаться MapX:

- Прочтите главу: *"Начинаем работу с MapX"*.
- Попробуйте примеры приложений и используйте их как шаблоны для создания собственных программ. Когда Вы установите MapX, примеры приложений будут установлены на Вашем компьютере, <Путь к MapX>\ [Samples40](#).

Примечание: Диск MapX CD включает дополнительные примеры программ, которые не устанавливаются на компьютер

автоматически. Вы можете найти эти примеры в на CD, в папке \Samples. Вы также можете найти самые последние примеры приложений на сайте MapX
<<http://www.mapx.com>>.

Особенности использования MapX в различных средах разработки ознакомьтесь с соответствующими разделами главы "*Начинаем работу с MapX*":

- Начинаем работать с Visual Basic
- Начинаем работать с Visual C++
- Начинаем работать с Lotus Notes
- Начинаем работать с PowerBuilder
- Начинаем работать с Delphi

2

Начинаем работу с MapX

- ◆ Что такое MapX?
- ◆ Системные требования
- ◆ Что содержит пакет MapX
- ◆ Комплектность MapX
- ◆ Установка MapX
- ◆ Подключение компоненты Map
- ◆ Для пользователей Visual Basic
- ◆ Обновление приложений C++ использующих более ранние версии MapX
- ◆ Работа с Visual Basic
- ◆ Работа с Visual C++
- ◆ Работа с PowerBuilder
- ◆ Работа с Delphi
- ◆ Работа с Lotus Notes
- ◆ Документация MapX
- ◆ Что читать дальше

В этой главе содержится информация необходимая Вам для успешной установки MapX и краткий обзор MapX. Прочитайте эту главу, если это Ваша первая встреча с MapX.

Что такое MapX?

MapX это инструментарий для разработчиков приложений. Он предлагает легкий и экономичный путь внедрения картографической функциональности в новое или существующее приложение. MapX это ОСХ компонент, который может быть быстро интегрирован в клиентскую часть приложения написанного на таких объектно-ориентированных языках как Visual Basic, PowerBuilder, Delphi, Visual C++ или в Lotus Notes (v4.5) используя Lotus Script. Разработчик таким образом имеет возможность работать в той среде которая ему более знакома, а пользователи в свою очередь получают доступ к картографическим функциям в привычном им приложении.

Системные требования

MapX это 32-bit ОСХ, и требует 32-bit версии Windows (Windows 95/98 или Windows NT 4.0). Приложения MapX не будут работать под Windows 3.1.

MapX работает с объектно-ориентированными языками программирования (такими как Visual Basic, Visual C++, PowerBuilder, или Delphi) или с Lotus Notes используя Lotus Script.

Что содержит пакет MapX

Комплект программного обеспечения MapX включает компоненты перечисленные ниже.

Примечание: При установке MapX Вы могли выбрать отмену установки

некоторых компонент программы и они могут быть не установлены на вашей системе. По умолчанию файлы устанавливаются в следующую папку: \<Путь к MapX>\MapInfo MapX 4.0\

Комплектность MapX

Файлы программы

Собственно компонент MapX OSH, с набором DLL и другими дополнительными файлами.

Примеры карт

Коллекция карт (в формате таблиц MapInfo) на различные регионы мира. Число карт включенных в поставку MapX зависит от того какую версию MapX вы установили; полная версия MapX включает больше карт чем оценочная версия функционирующая в течении 30 дней.

Примеры данных

База данных в формате Microsoft Access содержащая примеры демографических данных.

Примеры приложений

Примеры программ написанных на различных языках программирования, и демонстрирующие MapX в действии. Самые последние версии примеров приложений Вы можете посмотреть на сайте MapX: <http://www.mapx.com>

Утилиты

Утилита [MapInfo Geodictionary Manager \(GeoDictionaryManager40.exe\)](#), которая позволяет Вам зарегистрировать таблицу для использования с MapX, и GeosetManager, позволяющая Вам легко создавать геонаборы Geosets.

Документация MapX

Покупатели MapX получают *Справочник*, описывающий все объекты, методы, свойства, события и константы используемые в MapX. (Не включен в 30-дневную оценочную копию.)

Покупатели MapX также получают *Руководство разработчика MapX*. В нем даны основы картографии и подробное описание приемов работы с использованием MapX.

Справочная система MapX это комбинированная версия *Справочника* и *Руководства разработчика*, представленная в электронном виде. Она позволяет Вам получить быстрый доступ к информации необходимой для изучения и использования MapX.

Spatial Server Access

Используя [Spatial Server Access](#) (доступ к серверу пространственных данных), Вы можете отображать картографические данные их удаленного источника, такого как Oracle и Informix.

Прежде чем установить MapX

Прежде чем установить MapX, запишите Ваш серийный номер в легко запоминаемом месте, таком как обложка Справочника. Также заполните регистрационную карту и отправьте ее в корпорацию MapInfo.

Примечание: Мы настоятельно рекомендуем удалить все предыдущие версии MapX и закрыть все выполняющиеся Windows приложения прежде чем начать установку.

Удаление старой версии MapX

Удаление старой версии MapX достаточно простая процедура. Сделайте следующие шаги:

1. Выберите [Start > Programs > MapInfo MapX v.X > Uninstall MapX v.X](#)

2. Появится экран **Uninstall Shield** и диалог запрашивающий Ваше подтверждение на удаление MapX и всех его компонент. Нажмите **ОК**.

Установка MapX

Процедура установки MapX описывается ниже.

Если конфигурация Windows не содержит программной группы MapX то процедура установки создаст такую группу. Если Windows уже содержит программную группу MapX, в процессе установки в этой группе будет создана новая пиктограмма MapX.

Чтобы установить MapX:

1. Вставьте MapX CD в CD дисковод (такой как D:). Нажмите кнопку Start и выберите Run.
2. Напечатайте или выберите [CD дисковод]\Setup.exe (т.е., D:\Setup.exe) и нажмите **ОК**.
3. Появится экран **Welcome (добро пожаловать)**. Выберите Next, чтобы продолжить процесс установки.
4. Появится экран **Лицензионного соглашения**. Выберите YES, чтобы принять условия соглашения и продолжить установку.

Появится экран выбора папки назначения. Укажите директорию в которую Вы хотите установить MapX. Если MapX еще не был установлен на компьютере, по умолчанию будет предложено следующее размещение: **Program Files\MapInfo MapX 4.0** .Если MapX уже был установлен, то по умолчанию будет предложена папка в которой уже установлен MapX.

Примечание: Мы настоятельно рекомендуем, чтобы вы удалили все предыдущие версии MapX и закрыли все выполняющиеся Windows приложения, прежде чем начать процесс установки.

Чтобы выбрать другую папку назначения нажмите кнопку **Browse**, и укажите новый путь. Нажмите кнопку **Next**, чтобы продолжить процесс установки.

5. Укажите какие компоненты продукта Вы хотите установить. Будет отображено требуемое дисковое пространство для устанавливаемых компонент. Когда выбираете компонент, будет появляться его описание. Если кнопка **Change** доступна, это значит что компонент имеет подкомпоненты. Нажмите кнопку **Change**, чтобы посмотреть список подкомпонент и требуемое для них дисковое пространство. Поставьте подкомпоненты которые Вы хотите установить. Например, если Вы выберете форматы **Экспорта/Импорта**, будет показано пять подкомпонент (GIF, JPG, TIF, PSD, и PNG) с требуемым дисковым пространством для каждого. Вы можете выбрать установку любого или всех имеющихся подкомпонент.


Если Вы хотите установить поддержку базы данных Lotus Notes, убедитесь что опция Lotus Notes помечена, при установке MapX. (Опция Lotus Notes это подкомпонент в разделе Драйверов (Data Drivers)). Смотрите раздел *Установка и настройка драйверов для Lotus Notes*, далее в этой главе.

6. Появится экран **Выбора программной группы**; укажите программную группу.
7. Появится экран **Начало копирования файлов**. Ознакомьтесь с представленной информацией, и если она корректна, нажмите кнопку **Next**, чтобы установить MapX. Появится индикатор, отображающий процесс установки.

Подключение компоненты Map

После установки MapX, Вам потребуется добавить элемент управления Map в среду программирования.

Для пользователей Visual Basic

Установка объекта Map в **Visual Basic toolbox** . Откройте проект **Visual Basic project** и проделайте следующие шаги:

Если Вы используете Visual Basic 4:

1. Нажмите правую кнопку в **Visual Basic toolbox**, и выберите **Custom Controls** из меню.
2. В списке диалога **Custom Control** найдите "MapInfo MapX V4". Если этот пункт еще не выбран, выберите его. Нажмите **OK**.

Если Вы используете версию Visual Basic 5 или новее:

1. Нажмите правую кнопку в **Visual Basic toolbox**, и выберите **Custom Controls** из меню.
2. В диалоге **Components**, нажмите на закладке **Controls** и найдите "MapInfo MapX V4" в списке. Если этот пункт еще не выбран, выберите его. Нажмите **OK**.

Объект Map появится в toolbox. Чтобы поместить карту на Вашу Visual Basic форму, выберите объект Map и нарисуйте прямоугольник на форме.

Если Вы сохраните Ваш проект, следующий раз когда Вы снова загрузите проект, пиктограмма Map будет появляться в toolbox автоматически.

Для пользователей C++

Вам понадобится включить MapX.cpp и MapX.h в Ваш проект. Они содержат **class definitions** и **method implementations** для доступа к объекту MapX. Файлы MapX.h и MapX.cpp находятся в поддиректории **Samples40\CPP** папки в которую Вы установили MapX.

Использование Visual C++ версии 4

В меню **Insert**, выберите **Files Into Project**. Выберите MapX.cpp как файл для вставки.

Примечание: Не используйте **Insert > Component command**. Как результат Вы получите **.cpp** файл, но он будет не полным.

Использование Visual C++ версий 5 и 6

Из меню **Project**, выберите **Add To Project > Files**. Выберите для добавления файл **MapX.cpp**.

Примечание: Не используйте команду **Project > Components And Controls**. Как результат Вы получите **.cpp** файл, но он будет не полным.

Обновление приложений Visual Basic использующих более ранние версии MapX

Если Ваш проект на Visual Basic использует старую версию MapX, используйте следующую процедуру для трансляции Вашего приложения.

1. Откройте проект приложения которое Вы хотите обновить до текущей версии.
2. Удалите объект map из формы. Запомните имя компоненты **map control** и все свойства которые были изменены у объекта по сравнению с принятыми по умолчанию.
3. Из меню **Tools**, выберите **Custom Controls**.
4. В списке **Available Controls**, снимите флажок у компонента **MapInfo MapX Control** и нажмите **OK**.

5. Вернитесь к диалогу **Tools > Custom Controls** и выберите объект **MapInfo MapX Version x Control** (где x, это номер текущей версии) и нажмите **ОК**.
6. Поместите новый компонент MapX Map на Вашу форму, и дайте ему тоже имя которое использовали в вашем проекте раньше.

После того как новый ОСХ добавлен в проект, восстановите также все его свойства, если вы изменяли их в прошлом проекте.

Обновление приложений C++ использующих более ранние версии MapX

Если Вы разработали приложение на C++ используя ранние версии MapX, Вам надо использовать новые классы **MapX wrapper classes** (**mapx.h** и **mapx.cpp**) чтобы обновить Ваше приложение до текущей версии MapX.

Вам может понадобиться изменить строки, передаваемые в **CreateDispatch**. В предыдущих версиях MapX, Вы могли использовать следующую строку для указания имени объекта, например:

```
Flds.CreateDispatch("MapX.Fields")
```

В текущей версии MapX, Вам требуется указать строку:

```
Flds.CreateDispatch("MapX.Fields.4")
```

Чтобы сделать Ваш код более совместимым с будущими версиями MapX, Вы можете использовать **GetClsid** вместо строк. Результат, возвращаемый по **GetClsid**, будет работать в текущей и будущих версиях MapX. Например:

```
Flds.CreateDispatch(Flds.GetClsid())
```


Эти изменения приложимы ко всем создаваемым (creatable) объектам.

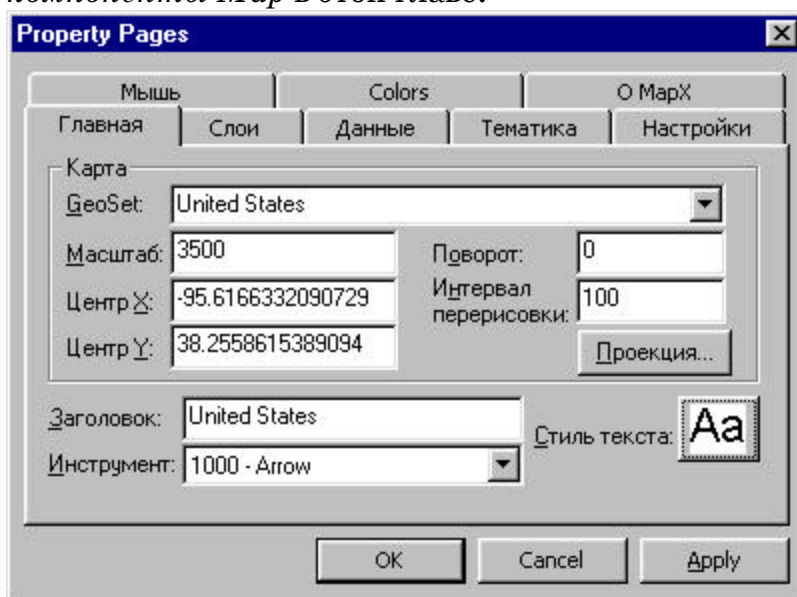
Что бы узнать побольше о создаваемых объектах, ищите в *Электронной справке* индексы "Create, Objects" в подразделе *Создание объектов* (Creating Objects), раздела *"Основы MapX"*.

Работа с Visual Basic

Создание простой карты

Используя MapX, очень просто добавить карту к вашему приложению. На самом деле, Вы можете поместить карту на форму Visual Basic без написания единой строчки кода.

1. Выберите объект Map из [Visual Basic toolbox](#) . (Если компонент Map не появляется в [toolbox](#), смотрите раздел *Добавление компоненты Map* в этой главе.



2. Создайте рамку в Вашей форме, представляющую область, в которую Вы хотите поместить карту. MapX отобразит предварительное изображение (preview) Вашей карты.
3. Нажмите правую кнопку мышки на [Компоненте Map](#), и выберите [Свойства](#) из быстрого меню. Откроется диалог [Properties](#).

4. Укажите в окошке **Инструмент**, которое находится в самом низу диалога **Главная**. Установите текущий инструмент на "1003 - Zoom In" и нажмите ОК.
5. Запустите Вашу программу, что бы увидеть карту. Учтите, что всякий раз, когда курсор в области карты, он действует как увеличивающая лупа.
6. Щелкните на карте. MapX увеличит карту, центрируя ее по тому месту где был щелчок. Вы можете щелкать несколько раз, увеличивая карту далее. Вы можете также выделить мышью прямоугольник, чтобы точно определить область, подлежащую увеличению.

После многократного увеличения, Вы увидите, что многие подробности карты становятся видимыми. Это происходит потому, что некоторые слои карты попадают в интервал масштабного эффекта (их объекты показываются только в определенном интервале масштаба).

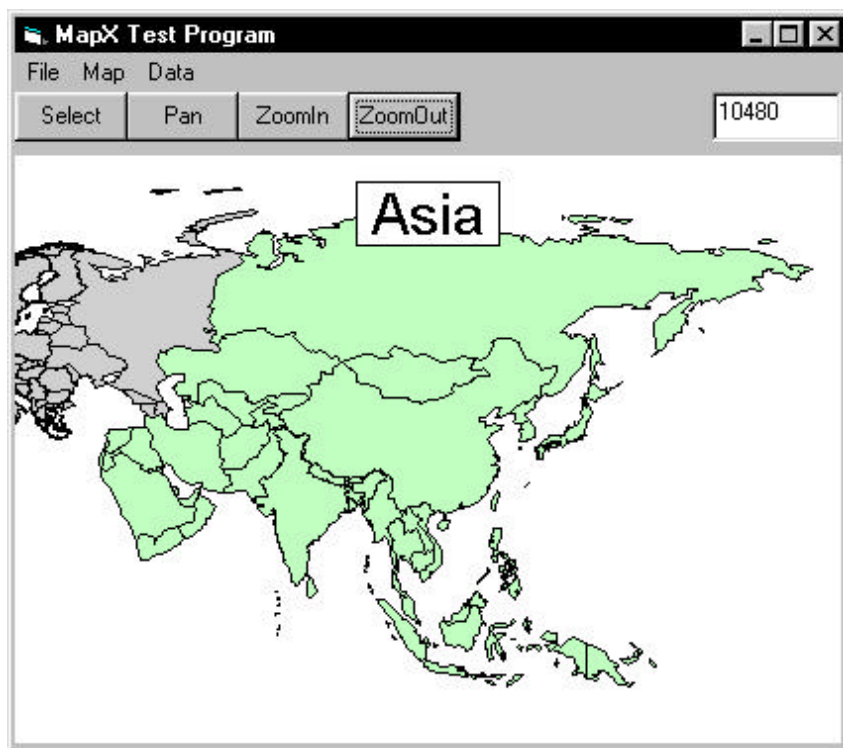
Что бы уменьшить карту, нажмите клавишу CTRL и щелкните мышкой на карте еще раз. Обратите внимание, что пока Вы держите нажатым клавишу CTRL, курсор приобретает значок уменьшающей лупы.

Один из путей изучить MapX – изучить простое приложение. Посмотрите на примеры приложений в папке:

[\<Путь к MapX>\MapInfo MapX 4.0\Samples40](#)

Учебный проект ASIA.VBP (Проект на Visual Basic 4)

Этот проект отображает простую форму с картой. Кнопки позволяют выбрать пользователю различные инструменты – **Выбор**, **Сдвиг**, **Увеличить**, **Уменьшить**. **TextBox** в правом верхнем углу позволяет пользователям видеть или устанавливать протяженность (охват) карты.



Выделение части изображения

Главная форма (Main form) содержит объект Map (Map1). Этот объект помещается в форму с использованием компоненты Map.

В [Visual Basic toolbox](#) Вы не увидите этот компонент, Вам надо добавить компонент к [toolbox](#).

Это приложение показывает данные по Азии, потому что программист выбрал геонабор "Asia". Вы выбираете геонабор (geoset) во время отладки программы, щелкнув правой кнопкой мышки на объекте Map и выбрав [Свойства \(Properties\)](#). Диалог [Свойства \(Properties\)](#) так же позволяет Вам устанавливать многие другие свойства карт.

Когда пользователь нажимает одну из кнопок в левой верхней части, приложение использует свойство Map.CurrentTool для изменения инструмента который будет использоваться. например:

```

Private Sub Command4_Click()
    map1.CurrentTool = miSelectTool
End Sub

```

Текстовое окошко в верхнем правом углу позволяет пользователю вводить нужный линейный размер карты (протяженность карты), в милях. Приложение использует свойство Map.Zoom чтобы изменить охват (масштаб) карты:

```

Private Sub Text1_LostFocus()
    map1.Zoom = Text1.Text
End Sub

```

Когда пользователь изменяет масштаб карты (используя кнопки **Увеличить** или **Уменьшить**), протяженность карты изменяется. Приложение автоматически обновляет текст в правом верхнем углу используя событие **MapViewChanged**.

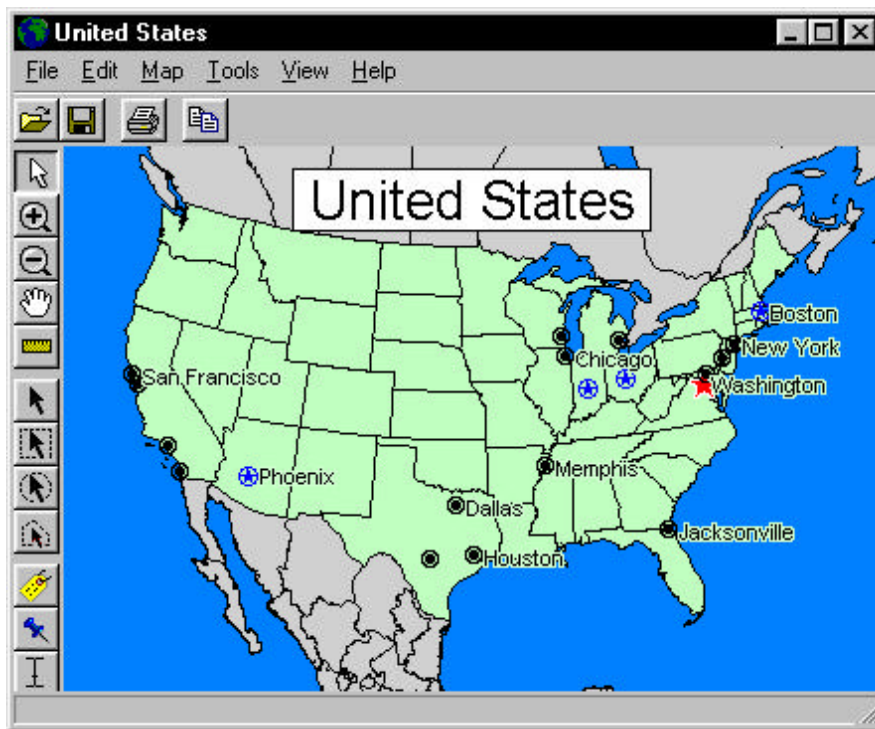
```

Private Sub map1_MapViewChanged()
    Text1.Text = map1.Zoom
End Sub

```

Учебный пример VBsample.VBP (Проект на Visual Basic 5)

Этот проект представляет другой пример использования картографического компонента. Однако он будет запускаться только при использовании MapX с версиями Visual Basic 5 или 6. Эта карта США которую MapX использует по умолчанию, но приложение организовано так, что может манипулировать любым geoset. Вы можете выбрать другой geoset на этапе проектирования программы, нажатием правой кнопки мыши на картографическом объекте и выбрав **properties**.



Этот проект использует три вида связывания данных:

- Опорные точки (Point Reference)
- Координаты XY
- Обычное (Normal)

Если Вы выбираете **Опорные точки (Point Reference)** или XY, то создастся слой. Более подробную информацию о слоях карты Вы можете найти в главе: "*Картографирование в слоях.*"

Инструменты

Линейка инструментов расположена у левого края формы. Проведите курсором над кнопками, чтобы определить их функциональность и выберите требуемый инструмент. Вы также можете получить доступ к инструментам используя меню **Tools**.



Инструмент стрелка
Увеличить
Уменьшить
Переместить
Линейка
Выбрать
Выбор-в-прямоугольнике
Выбор-в-круге
Выбор-в-полигоне
Подпись
Аннотация
Добавить текст аннотации

Вставка данных и тематических слоев

Вы можете поместить данные на эту карту используя команду [Insert Data feature](#) в меню [Map](#). После того как Вы выберете это меню, появится диалог [Add Dataset](#). Выберите данные которые Вы хотите добавить.

Что бы создать или изменить тематический слой, выберите [Create](#) или [Modify Theme](#) в меню [Map](#) и появятся соответственные диалоги. Более подробная информация о построении тематических карт содержится в главе: " *Тематическая картография и анализ* ".

Поиск объекта (Find Feature)

Команда **Find Feature** находится в меню **Map**. При выборе появляется диалог **Find**. Пользователь будет определять какой слой, объект или значение объекта будет использоваться для поиска. Возможно дополнительное уточнение параметров поиска. Более подробная информация о поиске объекта (объект Find), смотрите раздел: "*Поиск объектов на карте (Finding Features on a Map)*".

Код, используемый в этом приложении следующий:

```
For Each ftr In lyr.Selection
    ' The children of the layer are the individual
    ' features
    Set ftrNode = QueryTree.Nodes.Add(lyrNode,
tvwChild,lyr.Name _
    & ftr.Name & Str$(ftr.FeatureID), ftr.Name)
    For Each fld In ds.Fields
        ' Each feature has data attached to it; add
        ' this data as a child of the feature
        lyr.KeyField = fld.Name
        QueryTree.Nodes.Add ftrNode, tvwChild, ,
lyr.KeyField _
        & ": " & ftr.KeyValue
    Next
Next
Next
```


Выбор на заданном расстоянии

Что бы сделать выбор на заданном расстоянии, пользователь должен использовать один из инструментов выбора, для задания той области, в которой надо провести поиск. Затем он должен выбрать команду **Выбор на заданном расстоянии (Select Within Distance)** из меню Карта. Откроется диалог **Выбор на заданном расстоянии (Select Within Distance)**. Пользователь должен указать слой, расстояние, единицы измерения и объекты, которые надо выбрать. Более подробная информация о Выборе на заданном расстоянии находится в главе 7: *"Объекты и выборки"*.

Для этого используется следующий код:

```
' Здесь только один слой, для которого создается
выборка
Set ToBeSelectedLayer = _
fMainForm.Map1.Layers(ToBeSelectedLayerCombo.Text)
For Each ftr In SourceFeatures
' For each feature, select everything within the
distance
    Set FeaturesToSelect = _
    ToBeSelectedLayer.SearchWithinDistance(ftr, _
    SearchDistance, SelectionUnit, _
    miSearchTypePartiallyWithin)_
    ToBeSelectedLayer.Selection.Add
FeaturesToSelect
Next
```

Инструментарий выбора

Используйте этот инструментарий для выбора по определению площади, в пределах которой осуществиться выбор. Инструментарий выбора осуществляет доступ к данным, связанным с картой. Можно пройти к инструментарию через меню **Карта**.

Работа с Visual C++

Один из путей изучения MapX это разбор примеров приложений.

Просмотрите примеры приложений, находящиеся в папке:

<Путь к MapX>\Samples40

Когда будете изучать пример приложения C++ [mapxsamp.cpp](#), обратитесь к главе: "*Работа с Visual C++*", это поможет Вам понять, как использовать MapX с C++.

Внимание: Эти разделы предполагают, что Вы используете модель Microsoft's Document/View (стандартное приложение MFC AppWizard.) Пример приложения может быть встроен в [Developer Studio](#) с [mapxsamp.mdp](#).

Работа с PowerBuilder

Создание простой карты

Следующие шаги обеспечат создание нового проекта и поместят компоненту Map в приложение.

1. Для создания нового приложения держите Вашу карту открытой в **Application Painter** и выберите команду **New** из меню **File**. Укажите файл для **Application Library** и имя для нового приложения. Выберите **Yes** при запросе “Создать шаблон приложения?” (Application Template).
2. Откройте главное окно документа (для шаблона **Application Template**, оно называется **w_genapp_sheet**). Выберите его в **Application Painter** и затем нажмите клавишу **Назад (return key)**, что бы открыть **Window Painter**.
3. Выберите **OLE** из меню **Элементы управления (Controls)**.
4. Выберите закладку **Insert Controls** из диалога **Insert Object**, и выберите **MapInfo MapX V4** из открывающегося списка. Нажмите **ОК** для возврата к **Window Painter**.
5. Щелкните кнопкой мышки в окне документа что бы вставить новый объект Карта.

Работа с Delphi


Добавление компоненты Map

Следующие шаги устанавливают MapX в **Delphi package**. Это необходимо сделать только один раз.

1. Откройте Delphi с новым, пустым проектом.
2. Выберите **Import ActiveX Control** из меню **Components**.
3. Выберите **MapInfo MapX V4** из списка, и нажмите **Install**. В диалоге **Install**, установите его в по умолчанию в пакете программ **Borland User's Components**. Нажмите **Yes** чтобы перекомпилировать пакет программ (package), затем закройте и сохраните окно **Package**. Пиктограмма MapX должна появиться в **Controls palette**, в разделе **ActiveX**.
4. Закройте программу Delphi. В меню **Windows** найдите файл **<Path to MapX>\Samples40\Delphi\DelphiWrappers\DelphiFix.exe**. Запустите этот модуль, что бы исправлять некоторые ошибки в исходных файлах MapX генерируемых Delphi.

Создание простой карты

Следующие шаги могут использоваться при создании нового проекта, который включает в себя MapX.

1. Выберите иконку **MapX** из панели **Tools**, в разделе **ActiveX** .
2. Нарисуйте рамку окошка в форме, в этом месте появится компонент Map. Delphi может показать предварительный вид компоненты Map в такой форме.

3. Выберите команду **Run** из меню **Run**, что бы запустить новое приложение.

Работа с Lotus Notes

Введение

Что бы встроить компонент **MapX** в **Lotus Notes**, нужно иметь **Notes client** версии 4.5 или позднее.

Клиентские приложения Notes не могут вызывать **Datasets.Add** с типом **miDataSetUnbound**. Такое ограничение вызвано дефектом в Notes 4.5 (касающимся конвертации для логических величин) и который препятствует использованию метода **MapX RequestData** для настроек несвязанных данных через **LotusScript**. Lotus планирует исправить этот недостаток к следующей своей версии.

Установка и настройка драйверов для Lotus Notes

Если Вы хотите установить поддержку для баз данных Lotus Notes, убедитесь, что соответствующий флажок был установлен Вами при установке MapX. (Этот подраздел для Lotus Notes является подкомпонентой (sub-component), в разделе *Data Drivers Install*.)

MapX получает доступ к базам Lotus Notes при использовании поддержки файлов **MNotesDataset.dll** и **MMapXColumnInfo.dll**, которые хранятся в директории **MapX Program**. Если Вы поставили флажок **Lotus Notes Data Driver** во время установки, то установщик MapX попытается зарегистрировать эти библиотеки dll.

Внимание: Если файл `nnotes.dll` (являющийся частью Lotus Notes) находится в другой директории, то установщик MapX не сможет зарегистрировать эти библиотеки dll, и во время установки появится сообщение об ошибке.

Что бы исправить эту ошибку, измените Ваш `SYSTEM PATH`, включив в него директорию с Вашим Lotus Notes. Затем зарегистрируйте библиотеки dll вручную (запустив `regsvr32.exe`) или перезапустив установщик MapX с поставленным флажком `Lotus Notes data driver`.

Внимание: Для доступа к базе данных на Notes server, должна быть хотя бы одна база данных из сервера, установленная на Вашем компьютере с Notes (desktop.dsk).

Доступ к данным через Notes View

Доступ к `Notes View` обеспечивается использованием объекта `NotesViewInfo`. Этот объект имеет три свойства: `Server`, `Database` и `View`.

- Свойство `Server` указывает имя `Notes server`. Если файл доступен локально, свойство может быть установлено в виде нулевого стринга.
- Свойство `Database` используется для указания пути к файлу базы данных. На `Server`, этот путь будет относительно директории данных Notes; в локальном варианте это должен быть полный путь, начинающийся с имени дисковогода.

Внимание: Поскольку спецификация пути UNC не поддерживается `Notes API`, должен использоваться `drive mappings` для доступа к базе данных в сети.

- Свойство `View` это имя `Notes View` к которому происходит обращение.

Notes View обычно используется с объектом `MapX Fields` для определения, какие колонки View будут использоваться. Объект `MapX Fields` используется обычным способом, хотя важно указать заголовки колонок.

Пример: Использование Notes View из Visual Basic

Следующий пример Visual Basic показывает код, используемый для генерирования простой тематической карты с использованием колонок "state" и "value" из Notes View названных "State Info" в базе данных, называемой "states.nsf". Пример предполагает, что база данных существует на сервере с именем NS1 в поддиректории с примерами из директории данных Notes data.

```
Dim ds as DataSet
Dim flds As New MapXLib.Fields
Dim NDO As New NotesViewInfo

flds.Add "state", "BindField", 4, 0
flds.Add "value", "ValueField", 1, 1

NDO.Server = "NS1"
NDO.Database = "samples\states.nsf"
NDO.View = "State Info"

Set ds = Map1.Datasets.Add(miDataSetNotesView,
NDO, _
    "NotesThematic", "BindField", , , flds)

Map1.Datasets("NotesThematic").Themes.Add 0,
"ValueField", _
    "NotesThematic"
```

Пример: Использование Notes View из LotusScript

В предыдущем примере Visual Basic, типы данных `MapXLib.Fields` и `NotesViewInfo`, а так же константа типа данных `DatasetType` `miDataSetNotesView` были получены программой Visual Basic через библиотеки `MapX.dll` и библиотеку типов (type library). В такой среде, как LotusScript где библиотека типов не возможна, обработка таких объектов происходит следующим образом:

```
Dim uiws as New NotesUIWorkspace
Dim uidoc as NotesUIDocument
Dim ds as Variant
Dim flds As Variant
Dim NDO As Variant
Dim Map1 as Variant

'if the Map is embedded on a form, get the handle
as follows...

set uidoc = uiws.currentDocument
set Map1 = uidoc.getobject("Map1")

'в противном случае создайте новый объект карту...
set Map1 = createobject("MapX.Map.3")

set flds = createobject("MapX.Fields.3")
set NDO = createobject("MapX.NotesViewInfo.3")

flds.Add "state", "BindField", 4, 0
flds.Add "value", "ValueField", 1, 1

NDO.Server = "NS1"
NDO.Database = "samples\states.nsf"
```



```

NDO.View = "State Info"

Set ds = Map1.Datasets.Add(7, NDO,
"NotesThematic", _
"BindField", , , flds)

Map1.Datasets("NotesThematic").Themes.Add 0,
"ValueField", _
"NotesThematic"

```

Доступ к Notes через Notes Function Query

Доступ к данным Notes через функцию запроса сопровождается использованием объекта NotesQueryInfo. Объект NotesQueryInfo имеет свойства Server и Database, которые ведут себя точно так же как и для объекта NotesViewInfo. В добавок, объект NotesQueryInfo имеет третье свойство, называемое Query, которое представляет текст для предполагаемого запроса. Синтаксис этого запроса является стандартным, как и для формул Notes.

Функции запросов Notes обычно используются в сопряжении с объектами MapX Fields для определения того, какие поля в записях базы данных используются. В отличие от заголовков колонок Notes View, ссылки на имена полей записи не чувствительны к разным ситуациям (not case-sensitive).

Пример: Использование Notes Query из Visual Basic

Обращаясь к предыдущему примеру Visual Basic, предположим, что записи базы данных имеют поля для штатов (state) и значений, и анализ, проводимый Вами, требует создания тематики на основе записей в базе данных, где поле штат это один из NY, TX, CA или FL. Код будет выглядеть так:

```

Dim ds as DataSet
Dim flds As New MapXLib.Fields

```

```

Dim NDO As New NotesQueryInfo

flds.Add "state", "BindField", 4, 0
flds.Add "value", "ValueField", 1, 1

NDO.Server = "NS1"
NDO.Database = "samples\states.nsf"
NDO.Query =
"@IsMember(state;" & "NY" & ":" & "TX" & ":" & "CA" & ":" & "FL" & ")"

Set ds = Map1.Datasets.Add(miDataSetNotesQuery,
NDO, "NotesThematic", "BindField", , , flds)

Map1.Datasets("NotesThematic").Themes.Add 0,
"ValueField", "NotesThematic"

```

Как и в первом примере, это можно легко адаптировать к LotusScript или другим средам. Когда библиотека типов не доступна, miDataSetNotesQuery может быть перемещен по его числовому эквиваленту: 8.

Документация MapX

В дополнение к [Руководству разработчика](#), документация MapX включает [Справочник MapX](#) и файл [Справки](#).

Справочник MapX

Справочник MapX это полный путеводитель по объектам, свойствам, методам, событиям, кодам ошибок и выражениям для MapX.

Организованный в алфавитном порядке, большинство разделов включают описание назначения, перекрестные ссылки на другие разделы, и во многих случаях, примеры кода написанные на Microsoft Visual Basic и C++.

Файл справки

Всеобъемлющая система электронной справки MapX обеспечивает информацией, необходимой для обучения и грамотного использования MapX. Это все содержится и в *Справочнике* и в *Руководстве разработчика*. Получить доступ к этой информации можно следующим образом:

- Используя оглавление *Электронной справки*, выберите темы из книг. Щелкните на иконке книги и откроется ее содержание, выберите нужный Вам раздел.
- Используйте раздел **Find** для поиска по указанному слову. Выполните: **Start menu > Programs > MapInfo MapX > MapX Online Help**. Затем, выберите индекс и перейдите к закладке **Поиск**. Введите слово, которое надо найти и нажмите **Rebuild**. MapX покажет список слов, близких по написанию к искомому. Щелкните левой кнопкой мышки на нужном слове и откроется список разделов, содержащих нужное слово. Дважды щелкните на разделе, который Вам нужен или нажмите кнопку **Display** что бы показать раздел. Вы можете настроить Ваш поиск, нажав кнопку **Options**, после чего уточнить запрос поиска.
- Используйте раздел **Index** для быстрого поиска нужной темы. Напечатайте первые несколько букв того слова, которое Вы ищете. Откроется раздел, который наиболее подходит к введенным буквам. Щелкните на том индексе, который Вы хотите раскрыть.
- Кроме этого: Щелкните на тексте, подчеркнутым зеленой линией в любом окне Справки, и откроется информация, касающаяся ключевых слов и фраз.

Электронная справка сделана так, что бы показать информацию в небольшом окне которое не закроет целиком Ваше окно Карты, Списка и др.. Конечно, Вы всегда можете поменять размер окна Справки, на более удобный. Выполните команду **Help > Always On Top** что бы расположить окно **Справки** на экране так, что бы мог продолжаться сеанс работы в MapX. Или используйте клавиши **Alt-Tab** для переключения между экраном со **Справкой** и экраном с сеансом MapX.

Особенности использования файла справки

Использование примеров кода

Вы можете легко скопировать примеры программного кода из окна **Справки**. Что бы скопировать текст, выделите его, и нажмите **CTRL-C**; далее, чтобы вставить текст в программной среде, нажмите **CTRL-V**. Так же можно использовать технику **Drag and Drop** для переноса выделенного текста из **Справки** в другие приложения.

Загрузка последних версий Электронной справки

Корпорация MapInfo регулярно обновляет справочную систему MapX. Чтобы загрузить последнюю версию файла справки, посетите web site MapX по адресу: <http://www.mapx.com>.

На сайте MapX также представлены примеры приложений, дискуссии, где обсуждаются актуальные вопросы и можно взаимодействовать с другими пользователями MapX.

Что читать дальше

В этой главе был дан краткий обзор возможностей MapX. Другие разделы этой документации описывают возможности MapX значительно подробнее.

Перед тем, как Вы прочтете документацию дальше, обратите внимание на объектную модель [MapX OLE Automation](#), показанную в виде диаграммы. Эта диаграмма включена в поставку MapX и весьма полезна в процессе изучения MapX.

3

Новое в MapX

- ◆ Новое в MapX 4.0
- ◆ Улучшения и добавления в MapX 3.5.1
- ◆ Улучшения и добавления в MapX 3.5
- ◆ Улучшения и добавления в MapX 3.0
- ◆ Обновление приложений MapX 2.0 до MapX версии 3

В этой главе обобщены изменения, улучшения и дополнения, привнесенные в версиях MapX 4.0, 3.5.1, 3.5, и 3.0. Если Вы сделали обновление с MapX версии 2, обратите внимание на функциональные возможности добавленные в версии 3.0., и на то как обновить Ваши существующие приложения написанные с использованием MapX версии 2.

Новое в MapX 4.0

Новые возможности

- **Отображение поверхности и прозрачные растры:** MapX открывает файлы поверхности и 24-bit прозрачные растры.
- **Автоматическая регистрация растровых изображений:** MapX может непосредственно открывать растровые файлы (не .tab файлы) и автоматически регистрировать их используя информацию сохраненную в растровом файле.
- **Поддержка ADO и RDO:** MapX поддерживает ADO и RDO как источник данных для связывания.
- **Поддержка True color:** MapX по умолчанию использует 24-bit цвет.
- **Поддержка DB2:** MapX поддерживает открытие удаленных слоев из базы данных DB2.
- **Быстрый доступ к слою:** Улучшена производительность повторяющихся операций со слоем и dataset.
- **Поддержка векторных символов:** MapX теперь может отображать векторные символы (совместимые с MapInfo 3.0).
- **Создание объектов и инструменты для редактирования:** Инструмент select теперь позволяет выполнять операции перемещения, изменения размера и удаления объектов в редактируемом слое. Добавлены четыре новых инструмента создания объектов:
 - (1) `miAddPointTool`
 - (2) `miAddLineTool`
 - (3) `miAddPolylineTool`
 - (4) `miAddRegionTool`.

- **Поддержка базы данных Oracle 8i:** MapX поддерживает данные Oracle 8i.

Улучшенные возможности

- **Возможность определить курсор для инструмента:** Теперь в MapX программист может использовать **CreateCustomTool** для связывания встроенных возможностей редактирования и работы курсора стандартных инструментов. Дополнительно, программист может также указать, какой курсор использовать, когда нажаты и удерживаются клавиши **Shift** или **control** в процессе использования инструмента. Если **ControlCursor** или **ShiftCursor** не заданы будет использоваться курсор принятый по умолчанию **DefaultCursor** (либо стандартный либо пользовательский).
- **Нет необходимости в Geodictionary:** MapX теперь может работать без Geodictionary, за исключением случаев когда требуется автоматическое сравнение. Теперь в реестре могут быть следующие записи для Geodictionary:

- (1) Может содержать полный путь и имя файла Geodictionary. Эффект в том, что директория с данными устанавливается по пути указанному в этом ключе. Геословарь, хранящийся в памяти, инициализируется из файла Geodictionary и слои, содержащиеся в Geodictionary могут автоматически использоваться из памяти. Слои, добавленные к карте, являются добавленными к геословарю, размещенному в памяти.
- (2) Может содержать только путь. Директория с данными устанавливается по пути, определенному этим ключом. Геословарь, размещающийся в памяти, инициализируется пустым. Слои, добавленные к карте, добавляются и в геословарь, и из него производится в дальнейшем поиск.
- (3) Может не существовать вообще. Геословарь, размещающийся в памяти, инициализируется пустым. Слои, добавленные к

карте, добавляются и в геословарь, и из него производится в дальнейшем поиск.

- **Быстрый доступ к узлам объекта:** MapX размещает данные об узлах таким образом, что пользователь может запросить информацию о всех узлах за один раз, после чего информация возвращается в массив **SafeArray** (Смотрите **Feature.Nodes**).
- **Обработка пути:** MapX успешно обрабатывает файл пути.

Изменения в свойствах, методах и событиях

- **Событие AddFeatureToolUsed:** Это событие вызывается когда пользователь использует один из стандартных инструментов создания объектов (**miAddPointTool**, **miAddLineTool**, **miAddPolylineTool**, или **miAddRegionTool**). Поведение события подобно событию **PolyToolUsed**.
- **Feature.CenterX и Feature.CenterY:** Теперь возвращает истинный центр текстового объекта.
- **Layer.Add:** Теперь первый параметр имеет тип **VARIANT**, и может быть либо объектом **LayerInfo** либо строкой. Смотрите объект **LayerInfo**.

Примечание: **AddServerLayer** более не является предпочтительным способом открытия удаленного слоя через **Layer.Add**. Вызов объекта **LayerInfo** более эффективен.

- **Map.InfotipPopupDelay:** Позволяет программисту определить время задержки курсора мыши над объектом карты до появления всплывающей подсказки (если включен режим показа).
- **Map.InfoTipSupport:** Позволяет программисту включить или выключить режим отображения всплывающих подсказок (**InfoTips**) при задержке курсора мыши над объектом карты.
- **Свойство Map.MouseIcon:** Позволяет программисту определить собственный курсор, используемый как курсор, принятый в MapX по умолчанию. (Смотрите свойство **Map.MousePointer**)

- **Событие MapDraw:** Это событие будет вызываться один раз перед отрисовкой карты, и один раз после того как отрисовка завершена. Имеет один параметр, который показывает, началось или закончилось рисование.
- **Событие ResolveDataBindingEx:** Это событие подобно событию **ResolveDataBind**, но передает коллекцию **ResolveObjects** вместо строкового массива.

Дополнительные объекты

- **Объект FindResult:** будет возвращать информацию о **FindRC** в форме свойств, что делает более легким доступ к результатам поиска.

Примечание: Это переименованный объект **FindResults**.

- **Объект LayerInfo:** объект сохраняет параметры передаваемые методу **Layers.Add** и определяет тип из **LayerInfoTypeConstants**.

Примечание: **AddServerLayer** более не является предпочтительным способом открытия удаленного слоя через **Layer.Add**. Более эффективным является вызов объекта **LayerInfo**.

- **Объект RowValue и коллекция RowValues:** Объект **RowValue** представляет одно значения для поля в dataset. Коллекция **RowValues** это группа объектов **RowValue**. Эти объекты используются с **Layer.InsertFeature**, **Layer.UpdateFeature**, **Feature.Update** и **Dataset.RowValues** для чтения/обновления полей атрибутов в таблицах MapInfo.

Дополнительные свойства

- **BindLayer.ReferenceLayerField:** Указывает какое поле (числом начинающимся с 1) в таблице mapinfo будет использоваться для связывания данных. Если не указано, MapX просмотрит колонки данных и выберет колонку с наибольшим уровнем соответствия. Если соответствия не обнаружено, будет выбрана первая колонка.
- **Dataset.RowValues:**

- **Feature.FeatureKey:** Возвращает ключ объекта. Каждый объект в слое содержит уникальный для слоя ключ. Это строковое значение только для чтения. Это замена свойства **Feature.FeatureID** (которое работает как и прежде, но рекомендуется чтобы Вы использовали новое свойство **FeatureKey**).
- **Свойство Feature.Nodes:** Это свойство только для чтения, позволяет получить информацию об узлах таким образом что пользователь может запросить информацию об всех узлах за один раз и возвратит эту информацию в единый непрерывный блок памяти.
- **Find.CloseMatchesMax:** Возвращает максимальное число близких соответствий загруженных в **FindResult** из **SearchEx**.
- **Свойство LabelProperties.PartialSegments:** Установка значения true позволит MapX подписывать полилинии, даже если только небольшая часть полилинии видна в данный момент. Таким образом, это не работает это не работает для других объектов.
- **Свойство Layer.DataSets:** Это свойство только для чтения является коллекцией **DataSets** ассоциированной со слоем. Эта коллекция subset полной коллекции **DataSets (Map.DataSets)**. Объекты в **Layer.DataSets** те же что и объекты в **Map.DataSets**.
- **Layer.Editable:** Это свойство позволяет редактировать выборку в слое (переместить, изменить размер или удалить).
- **Свойство LayerInfo.Type:** Это свойство для чтения/записи чье значение является одной из констант **LayerTypeConstants**.
- **Map.MatchThreshhold:** Это минимальная пороговая величина, в процентах, требуемая для подбора слоев из источника данных. Используется для контроля автоматического сравнения (auto-matching) в процессе **Datasets.Add**.
- **Map.PanAnnnimationLayer:** Это свойство контролирует является ли слой анимации прорисованным в резервной памяти, когда используется инструмент ладошка. Значение по умолчанию **FALSE**.
- **RowValue.Dataset:** Это свойство используется для определения того, какое значение набора данных используется.

- **RowValue.Field:** Это свойство используется для определения того, какое значение поля набора данных используется.
- **RowValue.ReadOnly:** Это свойство определяет, может ли быть установлено свойство для объекта.
- **RowValue.Value:** Это свойство используется для присваивания значения полю.
- **RowValues.Count:** Это свойство только для чтения определяет число строк в коллекции **RowValues**.
- **RowValues.ReadOnly:** Это свойство только для чтения определяет может ли устанавливаться свойство коллекции.
- **Style.MaxVectorSymbolCharacter:** Это свойство только для чтения возвращает максимальный корректный код для векторного символа.
- **Style.MinVectorSymbolCharacter:** Это свойство только для чтения возвращает минимальный корректный код для векторного символа.
- **ThemeProperties.ApplyAttribute:** Контролирует, какой стиль оформления диапазонов применяется к тематическому слою. Может использоваться для изменений некоторых настроек, например, цвета или параметров диапазонов, не изменяя другие настройки.
- **ThemeProperties.BarFrameStyle:** Стиль фрейма столбчатой диаграммы.
- **ThemeProperties.BarGraduatedStack:** Определяет должны ли быть градуированны столбчатые диаграммы (стопкой).
- **ThemeProperties.BarIndependentScale:** Это свойство заменяет свойство **ThemeProperties.Independent**.
- **ThemeProperties.BarStacked:** Определяет показывать ли столбчатые диаграммы стопкой.
- **ThemeProperties.BarWidth:** Это свойство заменяет **ThemeProperties.Width**
- **ThemeProperties.BorderStyle:** Стиль границы круговой или столбчатой диаграммы.

- **ThemeProperties.ColorMethod:** Определяет метод, используемый для интерполяции между верхним и нижним цветами для получения промежуточных цветов.
- **ThemeProperties.DotColor:** Цвет точек в тематической карте плотности точек.
- **ThemeProperties.GraduateSizeBy:** Контролирует метод градуировки используемый для круговых и столбчатых картограмм или символов.
- **ThemeProperties.InflexionColor:** Имеет значение OLE_COLOR. Определяет цвет переломного значения.
- **ThemeProperties.InflexionRange:** Контролирует, какой диапазон берет цвет, указанный в **InflexionColor**. Диапазоны выше и ниже **InflexionRange** будут иметь смешанные цвета.
- **ThemeProperties.InflexRanges:** В картах диапазонов, точка перелома может использоваться для разделения диапазонов на две секции. Когда **InflexRanges** равно true, цвета в карте диапазонов будут оформляться от верхнего цвета к точке перелома **InflexionColor**, затем от **InflexionColor** к нижнему цвету диапазонов.
- **ThemeProperties.NegativeSymbolStyle:** Определяет стиль символов для отрицательных значений в тематической карте построенной методом градуированных символов.
- **ThemeProperties.PieClockwise:** Указывает рисовать сектора круговой диаграммы по часовой стрелке или против.
- **ThemeProperties.PieGraduated:** Это свойство заменяет свойство ThemeProperties.Graduated.
- **ThemeProperties.PieHalfPies:** Определяет как рисовать круговые диаграммы целыми кругами или полукругами.
- **ThemeProperties.PieStartAngle:** Начальный угол, от которого рисуется первый сектор круговой диаграммы.
- **ThemeProperties.PositiveSymbolStyle:** Это свойство заменяет свойство ThemeProperties.SymbolStyle.
- **ThemeProperties.RoundBy:** Указывает интервал округления диапазонов.

- **ThemeProperties.RoundRanges:** Контролирует, являются ли границы диапазонов округленными.
- **ThemeProperties.ShowNegativeValues:** Это свойство определяет отображать или нет отрицательные значения для тематической карты градуированных символов.

Дополнительные методы

- **Метод Coordsys.Clone:** Возвращает копию объекта **CoordSys**.
- **Метод DataSet.AddField:** Позволяет полю ('колонке') быть добавленной к набору данных, который является выражением, содержащим функции, операторы и поля набора данных (только из текущего dataset).
- **Метод Features.Clone:** Клонировывает коллекцию features.
- **Метод Find.SearchEx:** Расширяет функциональность метода search. Позволяет уточнять поиск для более полного совпадения, возвращаемого в коллекцию.
- **Метод Layer.BeginAccess:** Этот метод открывает таблицу и блокирует ее для доступа на чтение или запись. Он улучшает обработку для повторяющихся операций на слоях и наборах данных. Вы должны вызвать **EndAccess** один раз для каждого вызова метода **BeginAccess**.
- **Метод Layer.EndAccess:** Этот метод разблокирует картографические таблицы. Вы должны вызвать **EndAccess** один раз для каждого вызова метода **BeginAccess**.
- **Layer.Refresh:** Этот метод перемещает кэш из слоя. Это полезно для серверных слоев, имеющих включенное кэширование.
- **Метод Layer.Search:** Этот метод представляет возможность делать SQL запросы. Выражение (часть предложения "where") проверяет каждую строку в таблице слоя и возвращает коллекцию **Features**.
- **Метод LayerInfo.AddParameter:** Устанавливает параметры требуемые для данного вызова метода **Layers.Add**, и определяется типом передаваемого объекта **LayerInfo**.

- **RowValues.Add:** Это свойство добавляет объект **RowValue** к указанной коллекции **RowValues**.
- **RowValues.Clone:** Этот метод клонирует значения указанной строки из dataset и возвращает новый объект **RowValues**. Клонирование не является “только для чтения” или “чтение/запись”.
- **RowValues.Item:** Возвращает указанный объект **RowValue** из коллекции **RowValues**.
- **RowValues.Remove:** Этот метод удаляет указанный объект **RowValue** из коллекции.
- **RowValues.RemoveAll:** Этот метод удаляет все объекты **RowValue** из коллекции **RowValues**.
- Метод **Style.SymbolVectorColor:** Этот метод определяет цвет векторного символа.
- Метод **Style.SymbolVectorSize:** Этот метод определяет размер векторного символа в точках.

Улучшения и добавления в MapX 3.5.1

- Возможность записи, вставки, обновления прямо в слоях **SpatialWare**: Если Вы добавили слой **SpatialWare** используя метод **AddServerLayer**, любые редакции и изменения в этом слое будут фиксироваться в таблице **SpatialWare**. Смотрите **Feature.Update**, **Layer.UpdateFeature**, **Layer.AddFeature** и **Accessing Remote Spatial Data**.
- Экспорт образцов оформления: Объект **Style** имеет четыре новых метода, позволяющие Вам экспортировать символы, полигоны, линии и образцы стиля текста. Это методы **ExportSymbolSample**, **ExportRegionSample**, **ExportLineSample** и **ExportTextSample**.
- Поворот текстовых объектов: Новое свойство **TextFontRotation** объекта **Style** позволяет Вам контролировать угол поворота в градусах (по часовой стрелке) объекта **Text**. Ранее применялось только к объектам **Feature**.
- Диапазон **AllOthers**: Новое свойство **AllOthersCategory** коллекции **RangeCategories** и новое свойство **AllOthersCategory** коллекции **IndividualValueCategories** позволяет Вам определять категорию для всех диапазонов в тематических картах, отличных от карт диапазонов или индивидуальных значений. Текст, который появляется в легенде для объекта категории диапазонов **AllOthersCategory** могут устанавливаться со свойством **LegendTexts.AllOthersText**.
- Получение идентификатора объекта (**Feature Id**) по его имени: Новое свойство объекта **Layer FeatureIDFromFeatureName** возвращает ID объекта, указанного по имени.
- Выбор объекта по его идентификатору (**FeatureID**): Новый метод **SelectByID** коллекции **Selection** позволяет выбирать объект по его **FeatureID**.
- Добавление **Feature** по **FeatureID**: Новый метод **AddByID** коллекции **Features** позволяет Вам добавить объект **Feature** с указанным **FeatureID** к коллекции **Features**.

- Удаление объекта **Feature** по его **FeatureID**: Новый метод **RemoveByID** коллекции **Features** позволяет удалить объект **Feature** с указанным **FeatureID** из коллекции **Features**.
- Управление задержкой курсора: Новое свойство **WaitCursorEnabled** объекта **Map** позволяет включать и выключать **Wait Cursor**.
- Добавлены новые параметры для **Layers.AddServerLayer**: Принимает значение констант **LayerCacheConstants**. Константа **MiLayerCacheOn**, принятая по умолчанию, позволит сохранять атрибуты и объекты, считанные в память; если Вы будете увеличивать карту, то отпадает необходимость считывать объекты из базы данных. Поскольку MapX будет извлекать записи из памяти, возможно, Вы не увидите последних изменений в данных. Если **MiLayerCacheOff**, все данные будут извлекаться из базы данных; при этом вы всегда будете видеть все последние изменения, но данные будут извлекаться менее эффективно.
- Добавлен новый тип **Dataset**, сохраняющий массив **dataset**: Это **COM dataset** (нуждающийся в регистрации посредством regsvr32.exe перед использованием) который принадлежит только **IMMapXStaticDataset** (он не делает динамическое связывание). Добавлена также новая константа **DatasetTypeConstant**, **miDataSetSafeArray = 9**. При использовании этого dataset, ссылка на данные (параметр **SourceData** метода **Datasets.Add**) должна быть **VARIANT** которая содержит сохраняемый массив (safearray) заполненный данными, которые будут связываться.

Улучшения и добавления в MapX 3.5

Следующий обзор суммирует улучшения, сделанные в версии MapX 3.5.

Улучшения в производительности

- Улучшена скорость связывания для типов XY и PointRef.
- Увеличена скорость работы с Lotus Notes datasets.
- Увеличена скорость рисования карты.
- Поддержка импорта большего числа графических форматов, включая GIF, JPEG, и PNG.
- Поддержка Delphi Native Dataset для Delphi 3.0.

Улучшенные функциональные возможности

- Более быстрое создание тематических слоев для серверных и сборочных слоев: Когда создается тематика с помощью **Themes.Add**, рассчитываются диапазоны для слоев с большим числом строк, таких как сборочные или серверные слои. Новый параметр **ComputeTheme** метода **Add** позволяет Вам создавать невычисляемые тематические слои любого типа. Невычисляемые тематические слои дают Вам возможность создавать такие слои без автоматического вычисления диапазонов. Вы можете затем создавать диапазоны самостоятельно. Это быстрее для сборочных и серверных слоев. Смотрите **Themes.Add**, **DataMin**, **DataMax**, **ComputeTheme** и **Value**. Эта функциональность так же работает вместе с **IndividualValueCategories**, **MultivarCategories** и **RangeCategories**.

- **Поддержка NADCON:** Начиная с версии MapX 3.5, используется алгоритм NADCON для конвертации координат между NAD 27 и NAD 83 если эти координаты лежат в пределах территорий покрываемых NADCON (США, Пуэрто-Рико, и Виргинские острова). Если координаты лежат за пределами этих территорий или если они используют datums отличные от NAD 27 или NAD 83, MapX будет использовать для преобразования методы Молоденского или Bursa-Wolfe. Смотрите раздел *“Конвертация регионов (Datum)”*.
- **Поддержка Intellimouse:** Intellimouse имеет колесо как третью (среднюю) кнопку на мышках Microsoft. “Мышиное колесо” будет масштабировать карту, когда колесо вращается, прокручивать карту вверх и вниз, когда колеса вращается при удерживаемой клавише **control** и **”AutoScroll”** когда нажата средняя кнопка и движение происходит от того места, где был щелчок мышки. Свойство **MouseWheelSupport**, определяет уровень поддержки **Intellimouse**: нет поддержки, только масштаб/прокрутка (без автопрокрутки), или полная. Событие **MouseWheel** позволяет управлять событием.
- **Инкрементная прорисовка карты:** Ранее карта прорисовывалась как экранный bitmap; после завершения прорисовки, часть переводится на экран. Свойство **Map.RedrawInterval** определяет как часто переводится карта на экран, как если бы она была нарисована в памяти.
- **Сравнение числовых колонок:** Это свойство контролирует сравнение числовых полей. Это свойство появляется на странице свойства Data (во время отладки). Если для **MatchNumericFields** установлено значение true, то числовые поля рассматриваются как кандадаты в ключи (среди полей буквенно-цифровых полей) когда MapX производит автоматическое сравнение. Если установлено false, то числовые колонки не рассматриваются при автоматическом сравнении. Смотрите раздел *“Связывание данных”*.

- **Стиль пересекающихся линий:** Способность создать перекрещивание на карте, которое выглядит как настоящее перекрещивание и задать **Ширину линии** в измеряемых единицах, а не только в пикселах. Это делает более легким оформление карты. Смотрите свойство **Style.LineInterleaved**.
- **Стили линий:** Новый файл `mapx.rep` содержит большое количество новых стилей линий.
- **Максимальное число узлов:** Максимальное число узлов для регионов и полилиний увеличено до 1,048,572 для одного региона или полилинии. Если объект с более чем 32K узлов сохранен и таблица читается в версии MapX более ранней чем 3.5, объект(ы) будет невидим. Объекты в таблице число узлов, которых не превышает 32K, будут видимы.
- **Поддержка пространственных предикатов и функций в SQL** запросах передаваемых в SpatialWare:

раньше Вы были ограничены непространственными запросами в SQL:

```
SELECT SW_GEOMETRY, SW_MEMBER, CUSTOMER_NAME FROM
CUSTOMER
WHERE CUST_ID > 100000;
```

Теперь Вы можете включать пространственные операторы:

```
SELECT SW_GEOMETRY, CUST_ID, ST_AREA (SW_GEOMETRY)
FROM CUSTOMER, ZIPCODE
WHERE ST_OVERLAPS (CUSTOMER.SW_GEOMETRY,
ZIPCODE.SW_GEOMETRY)
AND ZIPCODE = '12345';
```

Изменения в свойствах, методах и событиях

- Новый параметр **ComputeTheme** метода **Theme.Add** позволяет создавать невычисляемые тематические слои. Невычисляемые тематические слои дают Вам возможность создавать тематические слои без автоматического вычисления диапазонов. Вы можете затем создать диапазоны самостоятельно. Это быстрый способ для сборочных и серверных слоев. Смотрите также **DataMin**, **DataMax** и **ComputeTheme**.

- Есть новые **ColorConstants** для системных цветов, таких как **miColorWindowBackground**.
- Опция **DLG** свойства **ConnectString** контролирует отображение диалога:
- Теперь Вы можете передавать объект **Rectangle** в метод **CreateRegion**, который создает регион, используя четыре угла прямоугольника.

Дополнительные свойства, методы и события

- Метод **ActiveAnnotation** возвращает выбранные аннотации (или NULL если нет выбранных аннотаций).
- Свойство **BackColor** контролирует, какой цвет подложки карты является 'стертым' перед рисовкой карты.
- Свойство **Border** определяет рисовать ли рамку у заголовка.
- Свойства **ComputeTheme**, **DataMin** и **DataMax** контролируют, вычислялись ли тематические слои и минимальное и максимальное значения для вычисляемых тематических слоев.
- Свойство **DrawLabelsAfter** позволяет лучше управлять тем, в каком порядке отрисованы подписи.
- Свойство **LineInterleaved** используется для создания типа пересекающихся линий, если такой стиль линии поддерживает пересечение со свойством **LineSupportsInterleave**.
- Свойство **LineSupportsInterleave** открыто только для чтения и является логическим, определяющим, поддерживает ли стиль линии пересечение или нет.
- Свойство **LineWidthUnit** устанавливает единицы исчисления ширины линии. Ширина может определяться в пикселах (по умолчанию) или в десятичных долях пунктов, указанием значения **StyleUnitConstants**.
- Событие **MapInitialized** вызывается немедленно после завершения инициализации карты. Оно нужно для использования контейнера приложения в наилучшее время, что бы сделать необходимые run-time конфигурации.

- Свойства **MapScreenWidth** и **MapScreenHeight** облегчают позиционирование легенды.
- Свойство **MouseWheelSupport** контролирует уровень поддержки **Intellimouse**: нет поддержки, масштаб/прокрутка (без автопрокрутки), или полная поддержка.
- Свойства **PaperHeight** и **PaperWidth** определяют высоту и ширину легенды в единицах **Map.PaperUnit**.
- Метод **PrintLegend** печатает легенду в указанном прямоугольнике с определенным содержанием.
- Свойство **RegionBorderWidthUnit** устанавливает единицы исчисления ширины границы для регионов. Ширина определяется или в пикселах (по умолчанию) или в десятых долях пунктов, указанием значения **StyleUnitConstants**.
- Метод **Datasets.RemoveAll** удаляет все объекты **Dataset** из коллекции.
- Метод **Fields.RemoveAll** удаляет все объекты **Field** из коллекции.
- Метод **Layers.RemoveAll** удаляет все объекты **Layer** из коллекции.
- Метод **Parts.RemoveAll** удаляет все объекты **Parts** из коллекции.
- Метод **Points.RemoveAll** удаляет все объекты **Point** из коллекции.
- Метод **Themes.RemoveAll** удаляет все объекты **Theme** из коллекции.
- Свойство **SearchPath** позволяет пользователям установить **Search Path** динамически. Это свойство доступно только в режиме исполнения.
- Свойство **ShowCount** контролирует показываются ли в легенде численные значения интервалов диапазонов.
- Свойство **Value** получает или устанавливает значение, использованное в слоях индивидуальных значений.
- Свойство **Visible** определяет будет ли видим диапазон в легенде.

Примечание: Следующий раздел суммирует улучшения, сделанные

в версии MapX 3.0. Если Вы обновляли прямо из MapX 1, Вы так же можете захотеть прочитать об особенностях, добавленных в версии 2.

Улучшения и дополнения в MapX 3.0

Следующий обзор суммирует улучшения, появившиеся в версии MapX 3.0.

- MapX теперь обеспечивает поддержку систем координат и проекций карт, включая поддержку пользовательских datums и аффинные преобразования. Подробнее смотрите обсуждение Систем координат.
- Введен новый тип слоя, Сборочный слой, позволяющий пользователю получать более подробную карту при щелчке мышкой на объекте карты.
- Доступ к пространственному серверу данных (Spatial Server Access, SSA) новая мощная возможность позволяющая разработчикам устанавливать соединение с данными хранящимися на пространственном сервере, таком как MapInfo SpatialWare в связке с базами данных Oracle и Informix. Пространственные серверы позволяют хранить большие массивы информации на сервере под централизованным управлением и защитой. Пространственные серверы типа SpatialWare имеет развитый инструментарий обработки запросов улучшенные возможности для организации пространственных данных. Подробнее смотрите *“Доступ к удаленным пространственным данным”*.

- Новым является и Динамическое Связывание Данных, улучшающее обработку при использовании ODBC для живого доступа к данным в сервере больших баз данных. Обычно, когда Вы связываете данные, MapX делает статическую копию данных, когда база данных открыта. Новый, добавочный аргумент метода **Datasets.Add** позволяет Вам указать, что связывание данных должно быть динамическим, в этом случае MapX забирает данные в “живом” режиме, нужно только копирование данных. Динамическое связывание данных так же поддерживается, если Вы разрабатываете собственную поддержку наборов данных.
- Новые возможности операций над объектами, позволяют объединять объекты, строить буферные зоны, разрезать, стирать точки, линии и полигоны. Более подробно, смотрите объект **FeatureFactory**.
- Список стандартных инструментов теперь включает инструмент выбор-в-полигоне. Этот инструмент позволяет пользователю рисовать полигон, и затем делать в нем выборку. Таким образом, метод **CreateCustomTool** теперь дает дополнительные, необязательные параметры, позволяющие Вам указывать различные курсоры, что должно использоваться, если пользователь нажал **SHIFT** или **CTRL** пока используется инструмент.
- Теперь Вы можете отображать точечные объекты используя либо символьные шрифты TrueType (как в ранних версиях MapX) либо растровые картинки (bitmap symbols). Символы шрифтов TrueType теперь поддерживают новое свойство **Style.SymbolFontRotation**, которые позволяют Вам вращать символы. Также объект Style теперь поддерживает новое свойство **LineStyleCount** и новое свойство **RegionTransparent** которое позволяет Вам создавать прозрачную заливку.
- MapX теперь включает новую утилиту, **GeosetManager.exe**, которая позволяет легко создать Ваш собственный geosets. Вы можете запустить это приложение из меню **Пуск**.

Примечание: В комплект MapX 3.0 включен исходный код проекта на Visual Basic 5 - **GeosetManager**, который имеет

функциональность подобную **GeosetManager.exe**. Для запуска утилиты **GeosetManager.exe**, установка этого примера VB проекта не требуется.

- Объект **BindLayer** имеет несколько новых свойств: свойство **CoordSys** определяет координатную систему когда Вы создаете слой методами **xy** или **pointref**. Свойство **Filespec** позволяет создавать постоянный слой вместо временного. Свойство **KeyLength** указывает длину колонки в символах в результирующем слое.
- Новое свойство **Bounds** возвращает прямоугольник представляющий минимальный описанный прямоугольник. Несколько объектов — объект **Map**, коллекция **Layers**, объект **Layer**, коллекция **Features**, и объект **Feature** — поддерживают свойство **Bounds**. Свойство **Bounds** полезно когда надо определить географическую протяженность одного или более объектов карты (т.е. установить масштаб достаточный для того чтобы видеть на карте все объекты). Примеры смотрите в разделе “Показать слой полностью”.
- Объект **Map** поддерживает новый метод **SaveMapAsGeoset**, который создает новый **Geoset** основанный на текущем состоянии карты (настройки слоев, масштаб, и т.п.).
- Объект **Theme** поддерживает новое свойство **Fields**, которое возвращает коллекцию **Fields**.
- Диапазоны для тематической карты теперь могут быть рассчитаны с использованием двух новых методов: **Natural Break** и **Standard Deviation**. Смотрите **ThemeProperties.DistMethod**.
- Два новых свойства — **ThemeProperties.AllowEmptyRanges** и **Legend.ShowEmptyRanges** — дают больший контроль над “пустыми” диапазонами (диапазоны не содержащие объектов).
- Два новых свойства — **Map.SelectionStyle** и **Map.ExportSelection** — позволяют контролировать отображение и экспортирование выделенных стилей.

- Даже если Вы не можете получить доступ к источнику данных через стандартное связывание данных MapX, Вы можете создать собственную поддержку набора данных dataset. MapX 3 обеспечивает расширенную архитектуру run-time, позволяющую вставлять в собственные типы dataset через основанные на COM интерфейс dataset. Смотрите **Custom Dataset Support**.
- Улучшения в объекте **Layer**: Новый метод **GetFeatureByID** извлекает объект по его ID. Метод **SearchWithinFeature** может теперь искать внутри автономного (stand-alone) объекта. Метод **SearchWithinDistance** теперь позволяет Вам указывать или объект **Point** или объект **Feature** как стартовое место для поиска; ранние версии не позволяли указывать объект **Feature**.
- Раздел **Layers** страницы свойств теперь включает дополнительные кнопки—**Добавить**, **Удалить**, **Вверх** и **Вниз**—которые позволяют добавлять, удалять и изменять порядок слоев в карте.
- Свойства объекта **Feature** - **Bounds**, **Length**, **Perimeter**, и **Area** теперь работают и с автономными (stand-alone) объектами.
- Если таблица MapInfo содержит дуги, круги или скругленные прямоугольники и приложение MapX изменяет эти объекты, MapX автоматически конвертирует объект в линии или полигон при обновлении объекта. Смотрите “*Автоматическая конвертация пространственных объектов*”. Пользователи могут нажимать клавишу ESC для прерывания рисования карты.

Обновление приложений MapX 2.0 до MapX версии 3

Некоторые изменения в MapX 3.0 могут накладываться с приложениями, написанными в версии 2. Если ваше приложение для MapX версии 2 работает не корректно с 3 версией, попробуйте следующее:

- В версии 2, стандартный инструмент подписывания (**miLabelTool**) имел значение 1010. В версии 3, инструмент подписывание имеет значение 1011; а новый инструмент **miPolygonSelectTool** имеет значение 1010. Если Вы использовали значение (1010) вместо его определения, Вам надо обновить такой код на новое значение.
- Событие **PolyToolUsed** теперь имеет дополнительные параметры (добавлены поддержка для модифицированных ключей, и позволяет отменять действие инструмента).
- Событие **DrawUserLayer** теперь имеет дополнительные параметры (добавлены к проблемам адресов, когда происходит рисование в метафайл).
- Метод **SearchWithinDistance** теперь имеет дополнительный параметр **Units**, который используется для указания единиц карты и их соответствия единицам радиуса поиска.
- Автономные (Stand-alone) объекты должны быть "присоединены к карте," так что система координат ассоциируется с объектом. Вы можете присоединить объекты к карте используя новый метод **Feature.Attach**. Нет необходимости использовать метод **Attach** для объектов созданных с использованием нового объекта **FeatureFactory**; метод **FeatureFactory** создает объекты которые уже присоединены к карте.
- Свойство **Map.MBR** переименовано в **Map.Bounds**, а свойство **Feature.MBR** переименовано в **Feature.Bounds**.

- Если Вы измените объект в слое (не автономный объект), и не произведете Update для сохранения изменений, свойства объекта **Feature - Bounds, Length, Perimeter** и **Area** возвращают значения основанные на “модифицированной” версии объекта (существующего в памяти). В более ранних версиях MapX, эти свойства возвращали значения соответствующие неизмененному объекту.
- Поведение свойств **Title.Position** и **Graphic.Position** отличается в версии 3; как результат заголовки и текстовые аннотации могут быть позиционированы отлично от того как это было в версии 2. Например, предположим, Вы присвоили значение **miPositionTL** ("Top Left") свойству **Graphic.Position**. MapX 3 позиционирует текстовую аннотацию, так что текст располагается выше и слева места прикрепления; MapX версии 2 использует противоположную ориентацию (помещая место прикрепления в верхнем левом углу текста). Так же, изменение свойства **Position** для заголовка (или аннотации) будет держать свойства объекта (X, Y) постоянными и изменять позицию объекта используя новое значение позиции. (В версии 2, MapX оставляет объект на месте и изменяет (X,Y) что приводит к новой позиции.)
- Библиотека ODBC используемая в версии 2 (**mideodbc.dll**) заменена новой библиотекой: **MODBCDataset.dll**. Для доступа к ODBC требуется также новая библиотека: **MMapXColumnInfo.dll**.
- Библиотека Lotus Notes используемая в версии 2 (**midenote.dll**) была заменена новой библиотекой: **MNotesDataset.dll**. Для доступа к Lotus Notes также требуется новая библиотека: **MMapXColumnInfo.dll**.
- MapX версии 3 включает обновленный набор библиотек (DLL) отвечающих за экспорт изображений в растровые форматы. Эти DLL имеют небольшое отличие в именах соответствующее новой версии (т.е. библиотека для экспорта в формат GIF была переименована из **lfgif60n.dll** в **lfgif70n.dll**).

- Если Вы создавали программу установки для Вашего приложения написанного с использованием MapX версии 2, Вам может потребоваться модифицировать часть скрипта программы установки, которая создает записи в реестре. Вместо регистрационных ключей, кончающихся на "2.0", MapX 3 использует регистрационные ключи оканчивающиеся на "3.0". Смотрите "*Распространение Ваших MapX Приложений*".
- Если Вы писали приложение на Delphi, PowerBuilder или LotusScript с использованием MapX версии 2, и Ваше предложение создавало автономные (stand-alone), создаваемые (creatable) объекты, Вам может потребоваться изменить синтаксис, для нормальной работы приложения с MapX версии 3. Например, Вам может понадобится изменить "MapX.Style.2" чтобы прочитать "MapX.Style.3". Смотрите *раздел Создаваемые объекты (Creatable Objects)*.
- Объект **ThemeCategories** теперь пополнен коллекцией трех новых объектов: **RangeCategories**, **IndividualValueCategories** и **MultivarCategories**. Например, в версии 2, Вы имели следующий код:

```
DIM categories AS ThemeCategories
SET categories =
Map1.Datasets(1).Themes(1).ThemeProperties.RangeCategories
```

В MapX версии 3, Вам потребуется изменить код, изменив имя объекта, следующим образом:

```
DIM categories AS RangeCategories
SET categories =
Map1.Datasets(1).Themes(1).ThemeProperties.RangeCategories
```

4

Основы картографии

- ♦ Основы картографии
- ♦ Организация Ваших данных и карт: Обзор таблиц
- ♦ Что такое GeoSets?
- ♦ Объекты карты
- ♦ Помещаем Ваши данные на карту
- ♦ Мощь MapX

Теперь, когда Вы установили MapX и ознакомились с широким набором новых функциональных возможностей в главе Что нового, Вам, наверное, не терпится приступить к созданию картографического приложения. Но сначала потратьте несколько минут на то, чтобы прочитать эту главу, особенно если Вы новичок в MapX. Эта глава даст Вам хорошее понимание основ успешного построения картографических приложений с использованием MapX.

Организация Ваших данных и карт: Обзор таблиц

Для использования MapX, требуются файлы содержащие Ваши записи и карты в формате MapInfo. MapX организует всю эту информацию в виде таблиц MapInfo; каждая таблица MapInfo это группа файлов которая представляет слой карты.

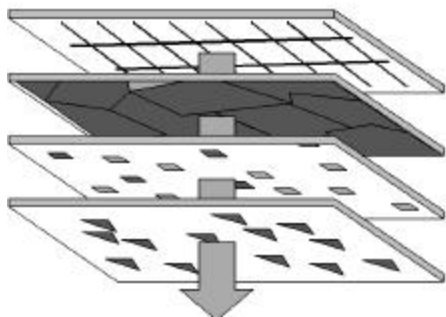
Файлы образующие таблицу

Все таблицы MapInfo имеют следующие файлы:

- <имяФайла >.tab: Это файл описывающий структуру таблицы MapInfo. Это небольшой текстовый файл описывающий формат файла содержащего данные.
- < имяФайла >.dat(.mdb, .aid, или .dbf): Эти файлы содержат табличные данные.
- < имяФайла >.mar: Этот файл описывает графические объекты (может не существовать если таблица не содержит графических объектов).
- < имяФайла >.id: Это файл ссылок, соединяющий табличные данные с графическими объектами (может не существовать если таблица не содержит графических объектов).
- < имяФайла >.ind: Это индексный файл. Индексный файл позволяет вам осуществлять поиск объектов карты с использованием объекта Find (поиск).

Таблицы MapInfo и слои MapX

Каждая картографическая таблица MapInfo может быть отображена как слой карты. Например, Вы можете отобразить таблицу клиентов, таблицу улиц и таблицу границ кварталов.



Представьте эти слои как прозрачные пленки, где каждый слой содержит различную картографическую информацию. Эти слои собранные стопкой один над другим и позволяют вам видеть все содержание карты одновременно.

Что такое GeoSets?

Рабочий набор или, по-английски, **geoset**, хранит набор слоев карты и их настройки, в легко доступном для Вас виде. Geosets это описание набора файлов карт в стандартном формате MapInfo (.tab) на один и тот же регион, отсюда и название - geoset. Geosets помогает Вам избежать операций требующих больших затрат времени таких как открытие и настройка отображения каждого отдельного слоя, каждый раз когда Вы хотите работать с ними как с единой картой. Расширение для файла geoset - **.gst**. **.gst** это текстовый файл, который содержит несколько ключей метаданных, указывающих компоненту MapX какие таблицы отображать и как отображать их.

Когда geoset открывается, он автоматически открывает все файлы, которые в нем содержатся и отображает их по умолчанию. Разработчик может изменить настройки по умолчанию, чтобы они отвечали конкретным требованиям. Настройки geoset's включают проекцию, охват карты, авто подписывание объектов, масштабный эффект для слоев и видимость слоя. MapX позволяет также открывать любой отдельный слой карты (.tab), указанный разработчиком. Geosets сделан для удобства и не обязателен для функционирования MapX. MapX не будет открывать рабочие наборы MapInfo (.wor файлы).

Geosets в комплекте MapX

Geosets	Слои в Geoset
Argentina	Argentina Major Cities Argentina Cities Argentina Country Boundary South America Country Boundaries World Ocean (Lat / Long)

Asia	Asia Asia Major Cities Asia Capitals World Ocean (for Asia) European Country Boundaries
Australia	Australia State Boundaries Australia State Capital Cities Australia Major Cities Australia Cities Australia Highways World Ocean (Lat / Long)
Brazil	Brazil Major Cities Brazil Cities Brazil Country Boundary South America Country Boundaries World Ocean (Lat / Long)
Canada	Canadian Province Boundaries Canadian Province Capital Cities Canada Major Cities Canada Cities Canada Highways World Ocean (Lat / Long) US State Boundaries
China	China Major Cities China Cities China Country Bdy China Highways World Ocean (for Asia) Asia Major Cities Asia Country Boundaries
Dallas, TX	Dallas Major Highways A Dallas Major Highways B Dallas Streets A Dallas Streets B Dallas Streets C

	Dallas City Boundary A Dallas City Boundary B Dallas Water Rivers Dallas Water Bodies Dallas, TX Raster Dallas Locations Crime Demo Map Crime County Map
DC	DC Landmarks DC Roads DC State Roads DC Interstate Roads DC Highways DC Zips Dc Area Landmarks Dc Water Layer Dc City Boundaries
Europe	European Country Boundaries European NUTS 1 Level Boundaries European NUTS 2 Level Boundaries European Capitals European Highways European Major Cities European Cities World Ocean (Lat / Long) Asia
France	France NUTS 2 Level Administrative Boundaries France Major Cities France Cities France Highway Map World Ocean (Lat / Long) European Country Boundaries
Germany	Germany NUTS 2 Level Bdys Germany Major Cities Germany Cities

	Germany Highways World Ocean (Lat / Long) European Country Boundaries
India	India State Bdys India District Bdys India Minor Cities India Major Cities India Capital Cities Asia World Ocean (for Asia)
Israel	Israel Major Cities Israel Cities Israel Country Boundary World Ocean (Lat / Long) Africa Country Boundary Asia
Italy	Italy Major Cities Italy Cities Italy Highways Italy NUTS 2 Level Bdys World Ocean (Lat / Long) European Country Boundaries
Japan	Japan Country background Japan Cities Japan Major Cities Japan Cased Roads Japan Highways Japan Rivers and Lakes World Ocean (Lat / Long)
Mexico	Mexico State Boundaries Mexico Cities Mexico State Capital Cities Mexico Highways Mexico Major Cities World Ocean (Lat / Long)

	US State Boundaries
Mid-Atlantic	Mid-Atlantic Capitals Mid-Atlantic Counties Mid-Atlantic Highway Mid-Atlantic Major Cities Mid-Atlantic Cities Mid-Atlantic States World Ocean (Lat / Long) US State Boundaries
North American	Canadian Province Boundaries Canadian Province Capital Cities Canada Major Cities Canada Cities Canada Highways Mexico State Boundaries Mexico Cities Mexico State Capital Cities Mexico Highways Mexico Major Cities US Cities US Major Cities US Highways US State Boundaries US State Capitals World Ocean (Lat / Long)
Portugal	Portugal Major Cities Portugal Highways Portugal NUTS 2 Level Boundary World Ocean (Lat / Long) European Country Boundaries
South Korea	South Korea Major Cities South Korea Cities South Korea Country Boundary Asia World Ocean (for Asia)

Spain	Spain Highways Spain Major Cities Spain Cities Spain NUTS 2 Boundaries World Ocean (Lat / Long) European Country Boundaries
UK	United Kingdom Cities United Kingdom Standard Regions United Kingdom Class A Roads United Kingdom Motorways World Ocean (Lat / Long) European Country Boundaries
US Detail	US Cities US top 20 Cities US Major Cities US Highways US State Boundaries US State Capitals World Ocean (Lat / Long) Mexico State Boundaries Canadian Province Boundaries US County Boundaries US Zipcode Boundaries Landmark Map
US	US Cities US top 20 Cities US Major Cities US Highways US State Boundaries US State Capitals World Ocean (Lat / Long) Mexico State Boundaries Canadian Province Boundaries US County Boundaries
World Detail	World Countries

	World Capitals World Top 25 Cities World Ocean (Robinson) World Graticule World Major Cities World Minor Cities
World	World Countries World Capitals World Top 25 Cities World Ocean (Robinson) World Graticule
Другие файлы	US 5-Digit Zipcode Points US 5-Digit Zipcode Points (compressed point file) Dallas Major Highways from MapXsite Streets 1.0 Dallas Streets from MapXsite Streets 1.0 Dallas City Boundary from MapXsite Streets 1.0 Dallas Water Rivers from MapXsite Streets 1.0 Dallas Water Bodies from MapXsite Streets 1.0

MapStats.mdb

Это не geoset, а база данных в формате access содержащая демографическую информацию, которая может быть привязана ко многим geosets входящим в комплект MapX.

MapStats.mdb	<ul style="list-style-type: none"> • Demographics for Australia • Demographics for Asia • Demographics for the World • Demographics for DC • Demographics for Mid-Atlantic States • Demographics for US • US Customer Database • US County Age Demographics
--------------	---

	<ul style="list-style-type: none"> • US County Age Demographics by Gender • US County Household demographics by Age, Income • US County Household demographics • US County Household Income • US County Housing Values • US County Population Demographics
--	--

Объекты карты

Ранее мы упоминали, что карты в MapX организованы как наборы слоев содержащих картографические объекты. Объекты карты доступны в MapX через объект Feature. Существует четыре основных типа объектов:

- **Полигоны:** замкнутые объекты покрывающие некоторую область, включая, эллипсы и прямоугольники. Пример, границы районов, областей, и т.п.
- **Точечные объекты:** представляют единое местоположение. Пример, размещение покупателей, ресторанов и т.п.
- **Линейные объекты:** протяженные объекты имеющие некоторую длину, включая линии, полилинии и дуги. Например, улицы, реки, линии электропередачи и т.п.
- **Текстовые объекты:** текст описывающий карту или объекты карты, например, заголовки и подписи.

Вы можете иметь отдельные слои для каждого типа объектов (более употребительно), или можете комбинировать типы объектов в одном слое. MapX позволяет вам создавать, редактировать, настраивать стили и отображать эти объекты, чтобы сделать карты отвечающие Вашим потребностям.

Помещаем Ваши данные на карту

Datasets позволяет Вам привязать данные к карте. Например, Вы имеете базу данных Microsoft Access с данными об объеме продаж по областям, вы можете связать свои данные с картой, и отобразив объемы продаж на карте увидеть закономерности и тенденции, которые невозможно отследить просто в табличных данных.

В настоящее время существует много различных типов баз данных, поэтому MapX позволяет Вам осуществлять соединения с несколькими различными типами **DataSources**. Первый аргумент метода **Datasets.Add** позволяет Вам указать значение **DatasetTypeConstants**, которое определяет тип данных, с которыми вы будете осуществлять связывание. Вы можете присоединяться к следующим типам данных:

Источник данных	Описание
ADO	Этот тип связывания использует MS Active data objects (ADO) recordset.
DAO	Объект DAO Recordset. Вы можете использовать его из Visual Basic data control, формы Access, или создавая его в Visual Basic, Access, или C++.
Delphi	Этот тип использует Borland BDE как источник данных.
Global Handle	Этот тип позволяет осуществлять связывание данных передаваемых блоками разделенными табуляцией.
Layer	Этот тип связывания данных позволяет создать dataset на основе полей из таблицы MapInfo.
Notes View/ NotesQuery	Тип связывания для Lotus Notes.
ODBC	MapX может использовать ODBC для извлечения данных из любого источника данных ODBC.
OLE Data	Для средств разработки типа PowerBuilder.
Oracle Express Objects	Позволяет осуществлять доступ к datacube как к dataset.

RDO	Использует объекты MS Remote Data Objects и RDO Resultset.
SafeArray	COM dataset позволяет осуществить статическое связывание данных из safearray.
Unbound	Если Вас не устраивает ни один из имеющихся типов связывания, MapX имеет ‘потайную дверь’. Используя этот тип Вы можете установить цикл в котором MapX будет запрашивать данные по одной ячейке за раз.

Инструменты для работы

Большинство картографических приложений содержит ряд инструментальных панелей (инструментов) для решения наиболее общих задач создания графических объектов (таких как рисование линий на карте) и задач навигации (таких как изменение масштаба карты). MapX уже содержит несколько таких предопределенных инструментов общего картографического назначения, плюс Вы можете создать Ваши собственные инструменты.

Стандартные инструменты

Используя MapX, Вы можете легко включить в ваше приложение ряд инструментов, не изобретая колесо заново. MapX представляет встроенную поддержку нескольких наиболее употребляемых картографических инструментов включая:

- **Инструменты навигации** (Увеличить, Уменьшить, Сдвиг, Центрирование) которые позволяют пользователю изменить масштаб и центровку карты.
- **Инструменты подписывания**, позволяющие пользователю подписать объект щелчком мыши на нем.
- **Набор инструментов для** осуществления различными путями выбора объектов карты.
- **Аннотации** (символы и текст).

Создание собственных инструментов

Если Вам необходим инструмент, которого нет в MapX , Вы можете просто создать свой инструмент, используя метод **CreateCustomTool**.

Мощь MapX

Теперь, когда Вы ознакомились с таблицами, слоями, geosets, объектами карты, datasets и инструментами, Вы готовы привести в действие все возможности MapX. Используя MapX, Вы можете найти слой в карте и найти местонахождения определенного объекта на этом слое. Например, Вы можете использовать это для поиска ближайшего поставщика услуг в Вашем городе. Или Вы можете подсчитать удаленность клиентов от больницы, а затем подсчитать, сколько пациентов живет в заданном радиусе от больницы. Или вы можете закрасить границы (районов, городов, областей, стран) в соответствии с общим числом покупателей на каждой территории или в соответствии с числом покупателей совершавших покупки в последний год. В MapX это называется тематическим картографированием.

Когда Вы начнете общаться с MapX на “ты”, Вы увидите, что сфера его применения ограничена только воображением.

5

Основы MapX

- ◆ Объект Map
- ◆ Страница свойств
- ◆ Слои
- ◆ GeoSets
- ◆ DataSets
- ◆ Аннотации
- ◆ Создание объектов

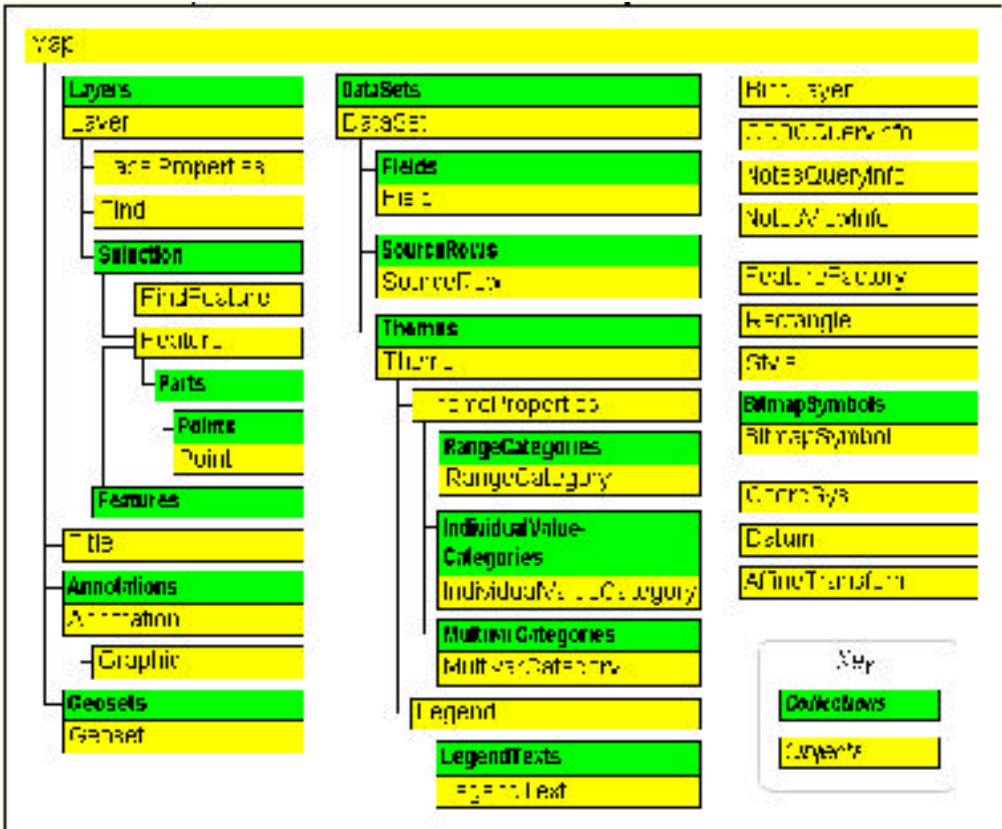
В главе “*Начинаем работу с MapX*” мы создали простое картографическое приложение, используя компонент MapInfo MapX, и ознакомились с некоторыми картографическими основами MapX. В этой главе рассматриваются основные составляющие MapX для создания и манипулирования картами в приложении. Некоторые разделы будут обсуждаться более детально в последующих главах.

Объект Map

Посмотрев на объектную иерархию MapX на нижеследующей диаграмме, Вы увидите, что на самом верху находится объект Map. Каждый объект, свойство или метод MapX, происходит от объекта Map. Каждое свойство и метод, показанные ниже объекта map делают вклад в создание всего объекта Map. Первостепенные объекты **DataSets**, **Layers**, и **Annotations** определяют каждый объект Map.

Объектная модель MapX

Примечание: Это не полная объектная модель MapX.



В нижеследующей таблице показаны некоторые свойства объекта Map, представленные численными значениями. Эти свойства могут быть изменены на этапе проектирования или во время выполнения программы с использованием нижеприведенных примеров кода:

Свойство	Описание	Пример кода
Zoom	Устанавливает число миль (единица измерения принята по умолчанию) отображаемых в окне карты.	Map1.Zoom = 500
Rotation	Поворачивает карту на определенное число градусов.	Map1.Rotation = 179
CenterX	Устанавливает координаты x и y которые могут быть долготой и широтой. Это зависит от проекции карты.	Map1.CenterX = -79.4458
CenterY	Устанавливает координаты x и y которые могут быть долготой и широтой. Это зависит от проекции карты.	Map1.CenterY = 44.9932

Используя объект [Map](#), Вы можете контролировать вид карты, манипулируя несколькими методами и свойствами карты. Некоторые свойства представлены другими объектами. Например, когда Вы видите карту MapX, Вы видите набор отдельных слоев представленных коллекцией **Layers**. Коллекция **Layers** это свойство объекта [Map](#).

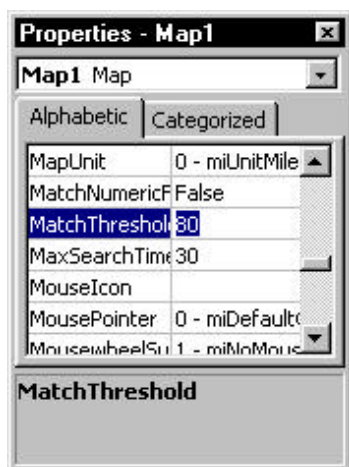
Теперь давайте посмотрим, как изменять свойства нашей карты. **Property Page** (страница свойств) позволяет нам манипулировать многими свойствами объекта [Map](#).

Страница свойств

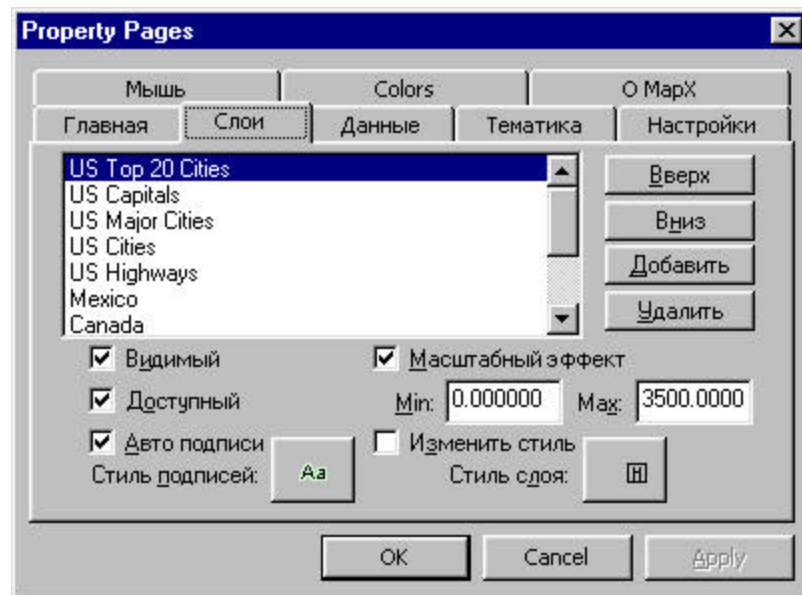
Страница свойств (Property Page) крайне полезный инструмент для изменения свойств карты в процессе создания и отладки приложения.

Чтобы открыть страницу свойств в проекте Visual Basic:

1. Нажмите **Custom** в окне **Properties**.
2. Нажмите кнопку  на строке **Custom**.



Вы также можете вызвать страницу свойств на этапе проектирования щелчком правой кнопки мыши на объекте **Map** на форме и выбрав "Properties..."



Вы также можете вывести диалог **Property Pages** во время исполнения программы. Для этого добавьте следующую строчку кода к событию `Map.Mouse_Up`.

```
If Button = 2 Then Map1.PropertyPage
```

Сначала будет проверено, была ли нажата правая кнопка мыши и затем отображен диалог **Property Pages**. Используйте правую кнопку, поскольку это не будет мешать использованию инструментов (левая кнопка).

Примечание: Уместнее использовать этот код только в процессе разработки и тестирования Вашего приложения, так как это может дать Вашим пользователям избыточный контроль над картой. Однако, если Вы хотите чтобы Ваш продукт отображал диалог для конечных пользователей, Вы можете использовать метод **Layers.LayersDlg** вместо метода **PropertyPage**, поскольку диалог **LayersDlg** имеет более дружелюбный интерфейс. Этот метод будет обсуждаться в следующей главе.

Слои

Давайте представим что мы имеем базу данных точек, представленную на карте черными звездочками. Сама по себе эта карта не очень полезна, но если поместить этот слой точек поверх слоя линейных и полигональных объектов, Вы уже получите полезную карту. Каждая индивидуальная карта представляет собой слой, и MapX сохраняет карту как коллекцию слоев.



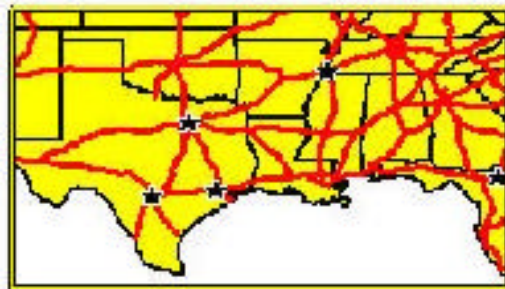
Regions



Lines



Points



Слои могут быть собраны и настроены на этапе разработки, с использованием диалога MapX **Properties**, или программно в процессе исполнения программы. В разделе выше было описано как диалог **Properties** позволяет разработчику манипулировать слоями просто изменяя их настройки. В следующей главе мы ознакомимся с примерами кода, который может быть использован для изменения свойств и методов слоев карты в любое время. Добавление новых слоев, удаление слоев, изменение видимости и стиля оформления слоя это некоторые из действий которые Вы можете осуществлять над слоями.

Коллекция **Layers**

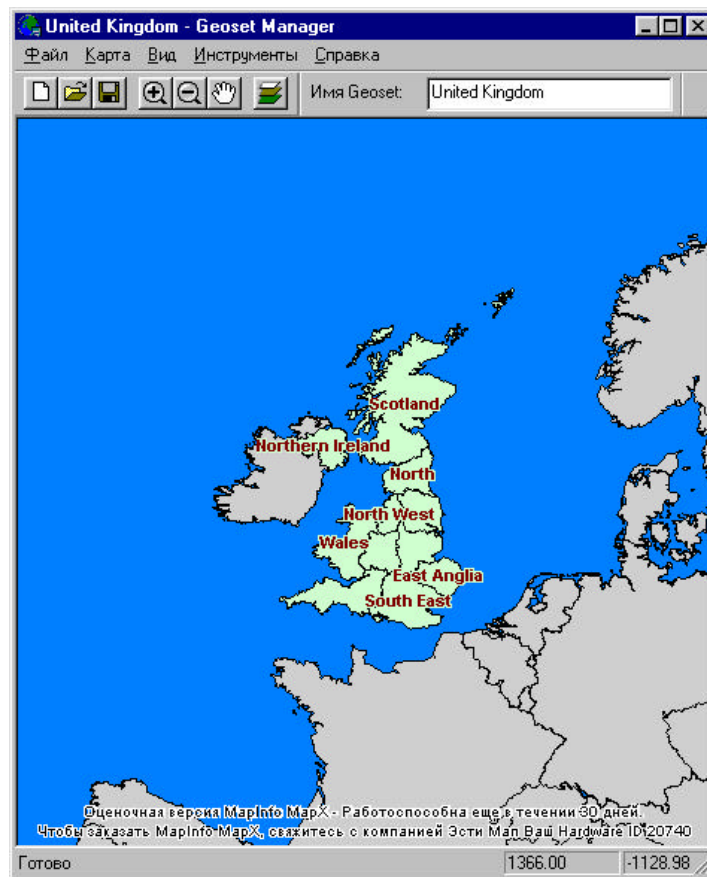
Коллекция **Layers** образуется из объектов **Layer** (0-N). Объект **Layer** образован коллекцией объектов (features), где у каждого объекта есть его собственные свойства и стили оформления. Коллекция объектов образована объектами **Feature**, которые соответствуют объектам карты, таким как точки, линии или регионы. Вы можете создать автономные **Feature** объекты, или Вы можете получить коллекцию объектов **Feature**. **Features** будут обсуждаться в главе: "*Объекты и выборки*".

GeoSets

GeoSet это набор слоев карты и их настроек. GeoSet определяет коллекцию таблиц MapInfo используемых в объекте Map и их настроек. GeoSet может быть определен на этапе разработки. Если он устанавливается во время исполнения программы, сначала будут удалены все загруженные слои и, а затем загружен новый GeoSet. По умолчанию загружается GeoSet - US.GST. Если Вы знакомы с MapInfo Professional, рабочий набор это аналог GeoSet в MapX.

Указание другого GeoSet для загрузки, добавление слоя к объекту map, или изменение вида слоя может быть осуществлено в диалоге Property Page, как обсуждалось ранее, или через программу Geoset Manager которая входит в комплект MapX и может быть найдена в MapX Program Group.

Ниже показана карта Великобритании с соответствующим Geosets.



Когда Вы удовлетворитесь видом вашей карты, Вы можете сохранить ее. При этом на диск будет записан файл **GeoSet (*.GST)**. Когда Вы откроете этот **GeoSet**, карта откроется, восстановив все настройки, в том виде, в каком Вы ее сохранили. **GeoSet Manager** позволяет модифицировать слои, управлять масштабом, подписями и другими свойствами. Более подробную информацию о **GeoSet Manager**, смотрите в приложение: " *Утилита GeoSet Manager* ".

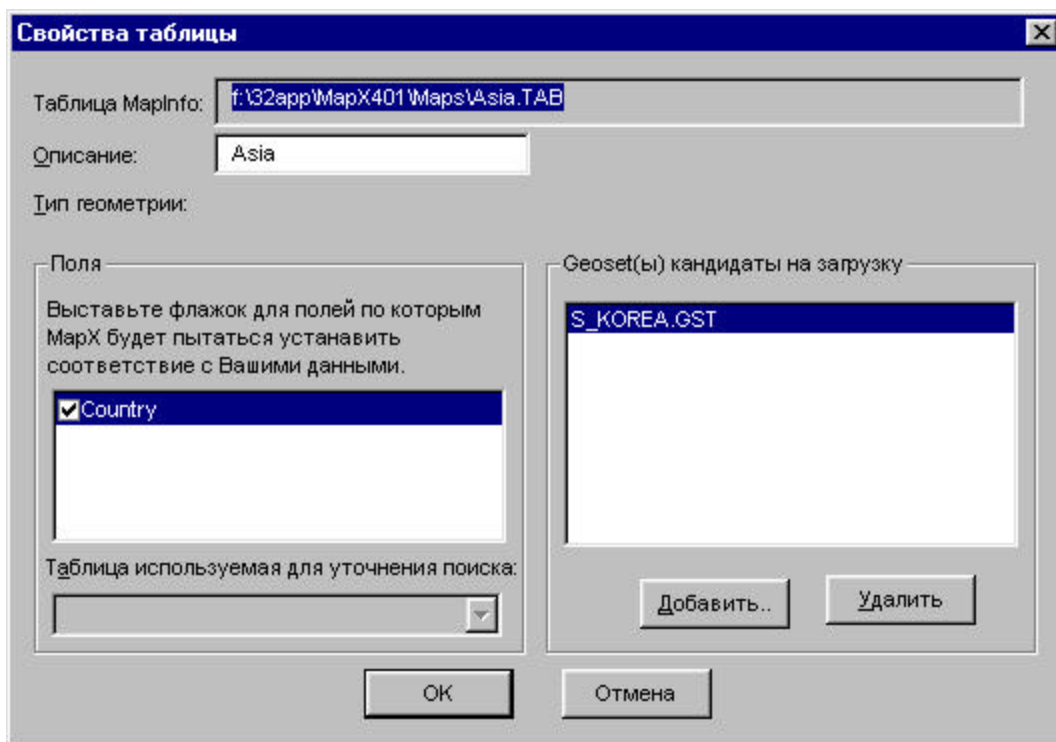
GeoDictionary

GeoDictionary используется когда осуществляется попытка найти соответствие между данными и слоем карты, для связывания данных с картой или создается тематическая карта. GeoDictionary это файл (обычно имеющий имя geodict.dct) содержащий информацию о том, какие слои карты могут использоваться для поиска, и какие поля могут использоваться для связывания данных с картой. Файлы должны быть зарегистрированы в [GeoDictionary](#) если Вы хотите использовать функции автосовмещение/автосвязывание (automatching/autobinding). Например, если у вас есть данные с объемами продаж по областям, [GeoDictionary](#) может определить какие данные будут извлечены из слоя карты “Russia”. Вы можете задать на программном уровне или непосредственно, колонки из карты, по которым проводится сравнение в таблице/файле данных, или Вы можете позволить MapX обратиться к [GeoDictionary](#) для поиска совпадений.

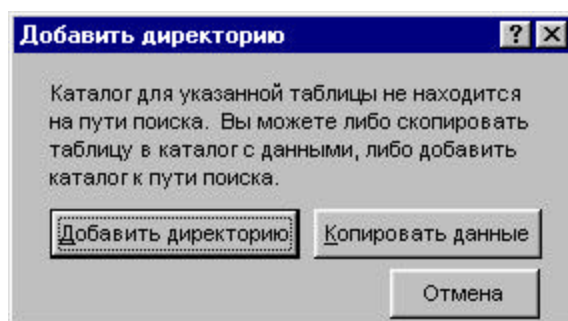
Редактирование GeoDictionary

Регистрация таблицы MapInfo в [GeoDictionary](#) требуется, если Вы хотите использовать возможность автосвязывания, автоматической процедуры привязки данных к карте. Привязка данных будет обсуждаться в главе: “*Размещение данных на карте*”. Программа MapX [GeoDictionary](#) в комплект MapX и позволяет Вам регистрировать новые таблицы MapInfo в [GeoDictionary](#). В дополнение к этому, в [MapX GeoSet Manager](#) имеются разделы меню **Tools**.

В MapX [GeoDictionary](#), нажмите кнопку **Register**, и выберите таблицу, которую Вы хотите зарегистрировать, установите свойства таблицы и введите описание таблицы. Если Вы хотите автоматически загрузить таблицу в GeoSet, нажмите кнопку **Add** и выберите существующий GeoSet к которому Вы хотите добавить таблицу.



Если путь для новой таблицы, которую Вы зарегистрировали, не виден в списке путей для поиска в MapX, программа Map Manager запросит, хотите ли Вы скопировать данные в директорию с данными MapX или просто добавить путь к маршрутам поиска путей MapX.



Когда Вы распространяете свое приложение и связанные с ним данные Вам необходимо помнить, что многие данные защищены законами об авторских правах. Убедитесь в том, что Вы имеете права распространять эти данные. Если Вы не уверены, свяжитесь с поставщиком этих данных.

Более подробная информация о [GeoDictionary](#), находится в Приложении: *Использование приложения GeoDictionary Manager*.

DataSets

DataSets позволяет привязать данные к вашим картам. Например, если у Вас есть информация о продажах по районам в базе данных MSAccess и карта районов, Вы можете привязать эти данные к карте районов и проанализировать продажи в районном разрезе, имея наглядное представление региональных различий, которые трудноуловимы в обычных табличных данных. Как показано выше, если для MapX требуется указание соответствия между Вашими данными и картой, соответствие определяется в процессе называемом автосовмещение/автосвязывание (automatching/autobinding). Предварительно Вы должны зарегистрировать карту в GeoDictionary для того чтобы иметь возможность использовать автосовмещение/автосвязывание.

После того как данные привязаны к карте, Вы можете видеть Вашу информацию географически. Визуальное представление данных позволяет создавать тематические карты. Тематические карты это карты на которых географические объекты представлены специальным образом, например различными цветами, в зависимости от данных которые сопоставлены объектам. Привязка данных к карте обсуждается в главе: "*Помещение данных на карту*". Тематическая картография обсуждается в главе: "*Тематическая картография и анализ*".

Аннотации

Коллекция Annotations это простой путь поместить текст и символы на карту. Аннотации “сидят на самом верху” над всеми другими слоями и не связаны с какими-либо данными. Если Вы знакомы с программным продуктом MapInfo, аннотации по целям использования подобны косметическому слою карты.

Ниже приведены свойства и методы для коллекции Annotations:

Методы коллекции Annotations

Метод	Описание	Пример кода
AddSymbol	Добавляет символ к коллекции Annotation. Используется стиль по умолчанию (как определено в Map.DefaultStyle).	Map1.Annotations.AddSymbol _ X, Y
AddText	Добавляет текст к коллекции Annotation. Четвертый параметр это начальная позиция текста относительно данных координат.	Map1.Annotations.AddText _ “Developer Services”, _ 79.44, 46.8889, _ miPositionTL
Remove	Удаляет определенную аннотацию из коллекции.	Map1.Annotations.Remove 1
RemoveAll	Удаляет все аннотации из коллекции.	Map1. Annotations.RemoveAll

Свойства коллекции Annotation

Свойство	Описание	Пример кода
Editable	Указывает будут ли редактируемы аннотации.	Map1.Annotations.Editable _ = True

Type	Определяет тип объекта Annotation.	If Map1.Annotations(2).Type_ = miTextAnnotation Then _ Print "It is text"
Graphic	Содержит объект Graphic, который имеет свойства для Annotation. Смотрите описание объекта Graphic в файле справки.	

Обратите внимание, что объект annotation не имеет свойств для изменения местоположения, стиля символа или текста. Эти графические свойства содержатся в объекте graphic. Чтобы модифицировать аннотации, модифицируйте графический объект аннотаций.

Следующий код добавляет символ в указанное местоположение:

```
'Add a symbol at location
Map1.Annotations.AddSymbol X1, Y1
```

Создание объектов

В объектной модели MapX, Вы можете создать одиночные, автономные (stand-alone) объекты, используя следующие классы объектов.

BindLayer	ODBCQueryInfo
BitmapSymbols	Parts
Feature	Point
Fields	Points
NotesQueryInfo	Rectangle
NotesViewInfo	Style

Следующие примеры показывают, как создать автономный объект, показать диалог стиля отображения, и использовать новый стиль оформления для слоя. Запомните, что в Delphi и PowerBuilder необходимо указывать номер версии MapX после имени объекта (т.е., "MapX.Style.4"). Это соглашение позволяет Вам иметь несколько версий MapX на компьютере одновременно. Если Вы разрабатываете приложение на MapX версии 4, и затем устанавливаете более позднюю версию MapX на тот же компьютер, новая версия MapX не будет взаимодействовать с приложением на MapX версии 4.

Пример на Visual Basic

```
Dim s as new MapXLib.Style  
s.PickRegion  
set Map1.Layers(1).Style = s
```

Пример на C++

```
CMapXStyle style;  
style.CreateDispatch(style.GetClsid());
```

```
// can also use
style.CreateDispatch("MapX.Style.4"),
// but above is more portable between versions of
MapX
style.PickRegion();
m_ctrlMapX.GetLayers().Item(1).SetStyle(style);
```

Пример на Delphi

```
Var
    s : variant;
begin
    s := CreateOleObject('MapX.Style.4');
    s.PickRegion;
    MapObject.Layers.Item(1).Style := s;
end
```

Пример на PowerBuilder

```
OLEObject s
long oleStatus

s = CREATE OLEObject

oleStatus = s.ConnectToNewObject("MapX.Style.4")
ole_1.Object.Layers.Item(1).Style = s
```

6

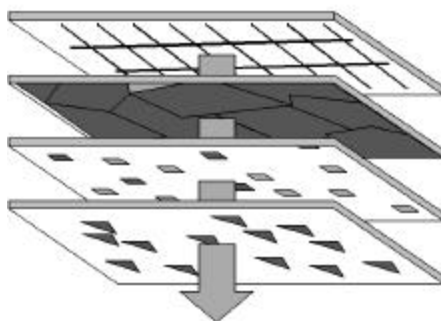
Картографирование по слоям

- ♦ Карта как набор слоев
- ♦ Коллекция слоев: строительные блоки Вашей карты
- ♦ Некоторые свойства коллекции слоев
- ♦ Некоторые методы коллекции слоев
- ♦ Объект Layer
- ♦ Порядок слоев
- ♦ Информация о слоях
- ♦ Проверка типов объектов на слое
- ♦ Масштабный эффект
- ♦ Создание подписей для слоя
- ♦ Аннотации
- ♦ Растровые изображения
- ♦ Анимационные слои
- ♦ Собственные слои пользователя

Эта глава представляет отношения между таблицами и картами и показывает как они располагаются слоями друг над другом, создавая необходимый уровень детальности карты.

Карта как набор слоев

Вы уже ознакомились с концепцией компьютерных карт, представляющих набор слоев в предыдущей главе. Каждая таблица MapInfo содержащая графические объекты может быть отображена как слой в окне карты. Например, Вы можете показать таблицу клиентов, таблицу уличной сети, и таблицу границ районов.



Представьте эти слои как прозрачные пленки, где каждый слой содержит различную картографическую информацию. Эти слои собранные стопкой один над другим и позволяют вам видеть все содержание карты одновременно. Например, один слой может содержать границы областей, второй слой может содержать символы, представляющие города, а третий слой представлять автострады. Эти прозрачные слои один над другим и образуют полную карту.

Теперь давайте перейдем к специфике создания карт.

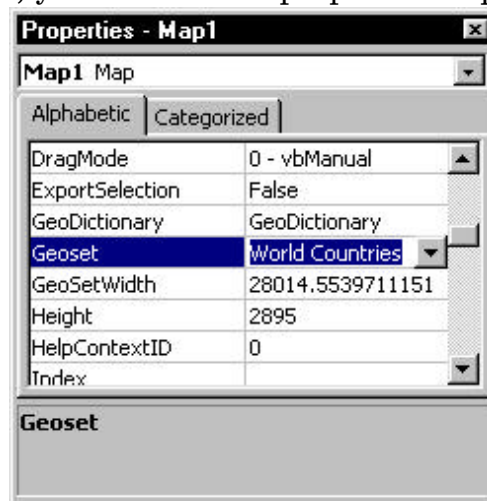
Коллекция слоев: строительные блоки Вашей карты

Коллекция Layers это свойство объекта Map и она содержит объекты Layer. Эти объекты представлены таблицами MapInfo и образуют Вашу карту. Каждый слой содержит различные объекты карты, такие как регионы, точки, линии или текст. Коллекция Layers имеет свойства и методы, используемые для таких операций как добавление и удаление объектов Layer из коллекции.

Как собрать коллекцию слоев

Один из путей работы с коллекцией слоев это загрузка geoset на этапе разработки приложения. Как мы рассказывали в предыдущей главе, geoset определяет коллекцию слоев карты и их настройки. Когда Вы помещаете объект Map на форму, по умолчанию MapX загружает [United States Geoset](#) (US.GST). Другими словами, Вы начинаете работу с коллекцией слоев определенной в [United States Geoset](#).

Допустим, Вы хотите написать приложение, работающее с картой стран мира, т.е. с коллекцией слоев, которая образует карту мира. Вы можете сделать это, указав на этапе разработки приложения World




Countries Geoset:

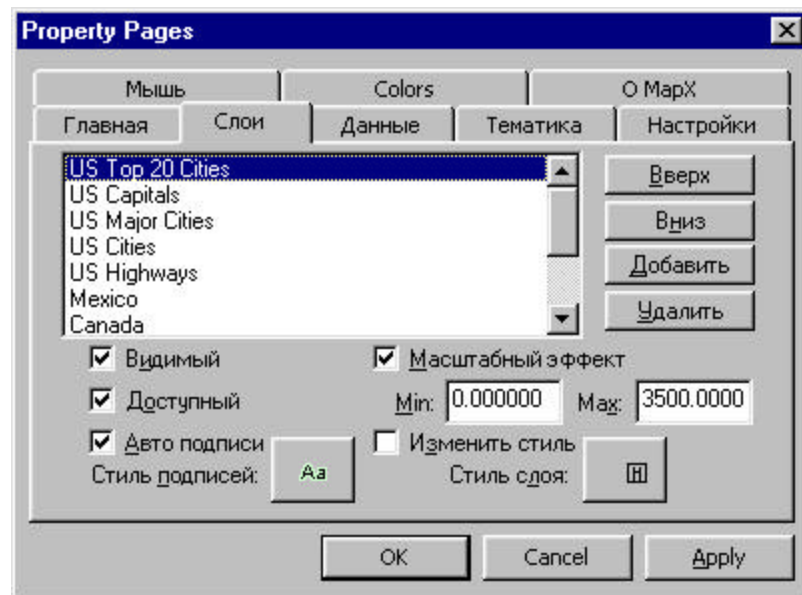
1. Выберите Geoset в окне [Properties](#).
2. Нажмите на стрелочке выпадающего списка.
3. Выберите World Countries из списка доступных Geosets.

Коллекция слоев определенная в World Countries Geoset будет загружаться, когда Вы будете запускать Ваше приложение и карта World Countries будет появляться в объекте Map.

Страница свойств

Самый быстрый путь увидеть все объекты Layer входящие в коллекцию на этапе разработки – использовать Страницу свойств (Property Page). Эта страница очень полезна для изменения свойств карты во время разработки и тестирования приложения. В данном случае мы будем использовать это, для того чтобы увидеть какие слои входят в коллекцию [World Countries](#).

1. Нажмите [Custom](#) в окне [Properties](#).
2. Нажмите  в строке [Custom](#). Появится [Страница свойств](#) (Property Page).
3. Нажмите на закладке [Layers](#) чтобы увидеть список слоев в коллекции.



Вы можете изменить многие свойства объекта Layer в коллекции, переупорядочить слои, добавить или удалить слои из коллекции.

Например, Вы можете сделать слой невидимым, когда карта отображается, выбрав слой в списке и сняв флажок у свойства **Видимый**. Когда карта будет отображаться, этот слой не будет виден. Конечно, Вы можете потом сделать этот слой видимым во время исполнения приложения программным путем, например:

```
Map1.Layers.Item(2).Visible = True
```

Мы будем больше работать со слоями программным путем в следующем разделе.

Некоторые свойства коллекции слоев

Каждая карта имеет коллекцию слоев. Коллекция Layers состоит из объектов Layer. Коллекция Layers имеет методы и свойства, используемые для добавления и удаления объектов Layer из коллекции.

Примечание: Справочник MapX и файл справки содержат полный список методов и свойств коллекции Layers.

Определение числа слоев в коллекции

Очень полезное свойство любой коллекции, свойство Count. Оно позволяет Вам узнать число экземпляров в коллекции, в данном случае число слоев на карте. Вы можете использовать это свойство, когда необходимо пройти в цикле через все экземпляры коллекции и например, получить имя каждого экземпляра:

```
Dim x as integer
For x = 1 to Map1.Layers.Count
    Print Map1.Layers(x).Name
Next
```

Извлечение слоя из коллекции

Свойство Item получает указанный объект Layer из коллекции. Свойство Item возвращает один из слоев как объект и это свойство принятое для коллекции Layers по умолчанию. Вы можете ссылаться на слой, используя его индексное значение 1, 2, и так далее или обращаться к слоям используя их имена, такие как Highways или Cities:

```
Dim lyr as Layer
Set lyr = Map1.Layers.Item("Highways")
```

-или-

```
Dim lyr as Layer  
Set lyr = Map1.Layers.Item(3)
```

В Ваших приложениях Вы часто будете ссылаться на объекты, методы и свойства коллекции Layers. Вы можете использовать тот факт, что свойство Item принимается по умолчанию, и вы можете не указывать его, для сокращения кода.

Например, каждая из строчек кода ниже делает идентичное присваивание (слой “Highways”).

```
Map1.Layers.Item(3).Visible = False  
Map1.Layers(3).Visible = False  
Map1.Layers.Item("Highways").Visible = False  
Map1.Layers("Highways").Visible = False
```

Определение географического охвата коллекции

Свойство Bounds возвращает объект Rectangle представляющий географический охват все слоев входящих в коллекцию (за исключением UserDraw слоя). Это свойство полезно, когда Вы хотите установить масштаб карты таким, чтобы были видны все объекты всех слоев:

```
'This sets the map bounds to the geographic extents  
of all layers in 'the collection effectively  
bringing the entire map into view.  
On Error Resume Next  
Set Map1.Bounds = Map1.Layers.Bounds  
End Sub
```

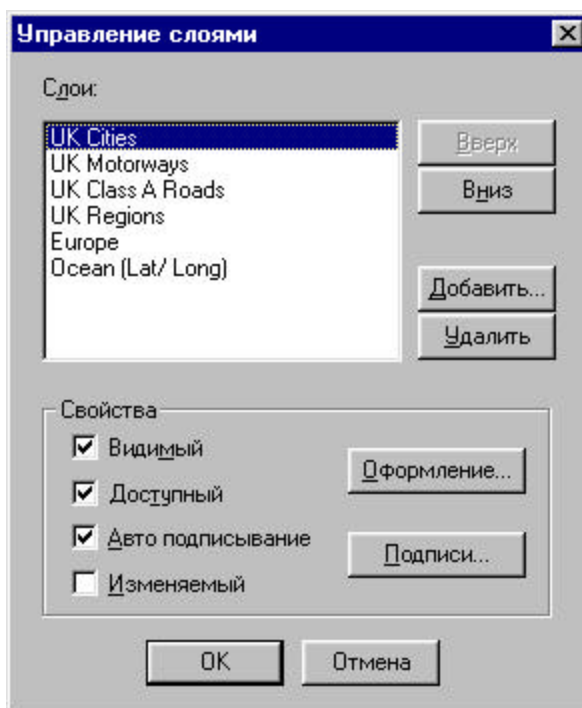
Некоторые методы коллекции слоев

Коллекция Layers имеет несколько методов, позволяющих контролировать какие слои из коллекции и как будут отображаться. Существует также метод позволяющий вызвать диалог [Управление слоями](#) позволяющий пользователям изменять настройки слоев. Эти методы также могут быть установлены на этапе проектирования с использованием [Страницы свойств](#), как это обсуждалось ранее в этой главе.

Примечание: Справочник MapX и файл справки содержат полный список методов и свойств коллекции Layers.

Использование диалога управления слоями

Метод [LayersDlg](#) представлен диалогом, в котором пользователь может добавить или удалить слой, изменить порядок слоев и изменить свойства слоя.



Если пользователь нажмет **ОК**, изменения сделанные в диалоге будут немедленно отражены на карте.

Map1.Layers.LayersDlg

Настройки диалога Управление Слоями

Диалог **Свойства отображения** позволяет Вам настроить стиль отображения каждого слоя в окне карты. В диалоге **Управление слоями**, пользователь может выбрать слой и нажать кнопку **Оформление**, чтобы открыть диалог **Свойства отображения**. В этом диалоге пользователь может изменить стиль оформления слоя, и установить масштабный диапазон в котором слой будет отображаться.



Режим отображения

Когда пользователь первый раз открывает карту, то границы, линии, точки и текст изображаются в соответствии с настройками по умолчанию в файле карты Geoset. Пользователь может изменить стиль отображения объектов, используя раздел [Режим отображения](#) диалога [Свойства отображения](#).

Например, пользователь хочет изменить стиль оформления улиц таким образом, чтобы они отображались, красной пунктирной линией. В диалоге **Управление слоями** пользователь должен будет выбрать слой улиц и нажать кнопку **Оформление**. Появится диалог **Свойства отображения**. Пользователь должен установить флажок **Однообразно**, что бы активизировать соответствующую большую серую кнопку. MapX показывает кнопки единообразного оформления, что бы все объекты слоя можно было бы оформить в едином стиле. Например, если слой содержит улицы, появится кнопка единообразного оформления линий. Нажатие на ней откроет диалог **Стиль линии**, в котором пользователь может изменить ширину, стиль и цвет линии для отображения улиц.

Для слоя границ, кнопка **Однообразно** откроет диалог **Стиль региона**, в котором пользователь может изменить тип границ и тип заливки полигонов. Диалог **Стиль символа** появляется, если пользователь хочет изменить стиль оформления для слоя содержащего символы или точки. Как и другие настройки отображения, **Однообразно** имеет действие только для текущей рабочей сессии. Чтобы сделать эти изменения постоянными, Вам необходимо модифицировать геонабор `geoset`.

Создание слоя

Метод **CreateLayer** позволяет Вам создать временный или постоянный слой в виде таблицы MapInfo. Метод возвращает объект Layer, объект который был добавлен к коллекции.

```
dim lyr as layer
set lyr = Map1.Layers.CreateLayer ("Temporary")
```

Добавление слоя

Метод **Add** добавляет слой к существующей коллекции и отображает его на карте.

Когда добавляете слой, Вы можете указать местоположение слоя в коллекции, используя необязательный параметр Position. Если параметр Position не указан, слой будет позиционирован автоматически, с учетом других слоев карты. Например, слой с точками помещается над слоем с регионами.

```
Map1.Layers.Add "C:\Data\Counties.tab"
```

-или-

```
Private Sub mnuAddLayers_Click()  
'используется common dialog для открытия таблицы  
MapInfo и добавления ее 'как слоя.  
    Dim sFile As String  
    With dlgCommonDialog  
        .DialogTitle = "Add Layer"  
        .Filter = "MapInfo Tables (*.tab)|*.tab"  
        .ShowOpen  
        If Len(.filename) = 0 Then  
            Exit Sub  
        End If  
        sFile = .filename  
    End With  
    On Error Resume Next  
    Map1.Layers.Add sFile  
End Sub
```

Удаление слоя

Метод **Remove** удаляет указанный слой из карты.

```
Map1.Layers.Remove 3
```

Удаление всех слоев

Метод **RemoveAll** удаляет все слои из карты.

```
Private Sub btnLayersRemoveAll_Click()
```

```
Dim nLayers As Integer
nLayers = Map1.Layers.Count
' remove all of the layers
Map1.Layers.RemoveAll
nLayers = Map1.Layers.Count
```

Упорядочивание слоев

Метод **Move** переупорядочивает слои в коллекции Layers. Первый параметр, откуда (верхний слой = 1) и второй - куда.

```
Map1.Layers.Move 1,2
```

Объект Layer

Объект Layer представляет векторные картографические данные в форме коллекции объектов карты имеющих predetermined тип, например, регионы, линии или символы. Обычно объект Layer соответствует географическим объектам из одной таблицы MapInfo. Каждый объект Layer в коллекции слоев независим от других. Их стиль оформления может быть изменен, может быть изменен масштабный диапазон, в котором слой будет видимым и т.п. и все это индивидуально, без какого-либо влияния на другие слои.

Вы можете манипулировать объектами слой в любое время в Вашей программе. Вы также можете использовать [Property Page](#), чтобы установить свойства в процессе разработки приложения, как это обсуждалось выше. Объект Layer имеет много свойств и методов. Методы, использующиеся для извлечения объектов из слоя, обсуждаются в главе: *Объекты и Выборки*.

Некоторые свойства слоя

Примечание: Справочник MapX и файл Справки содержат полный список методов и свойств объекта Layer.

Name

Имя слоя.

```
MsgBox Map1.Layers(4).Name
```

Visible

Делает слой видимым или невидимым.

```
Map1.Layers.Item(2).Visible = False
```

OverrideStyle

Указывает отменять ли принятый по умолчанию характеристики отображения слоя.

```
Map1.Layers(9).OverrideStyle = True
```

Style

Используемый стиль для слоя, если OverrideStyle - True.

```
Map1.Layers(9).Style = newstyleobject
```

ZoomLayer

Включает или выключает масштабный эффект для слоя.

```
Map1.Layers("States").ZoomLayer = True
```

ZoomMin

Определяет минимальный масштаб, при котором слой будет видимым.

```
Map1.Layers("States").ZoomMin = 45
```

ZoomMax

Определяет максимальный масштаб, при котором слой будет видимым.

```
Map1.Layers("States").ZoomMin = 580
```

Selectable

Указывает MapX определять, когда пользователь делает щелчок мышью на слое.

```
Map1.Layers("States").Selectable = True
```

AutoLabel

Указывает подписывать или не подписывать слой автоматически.

```
Map1.Layers("States").AutoLabel = True
```

В *Справочнике MapX* и файле справки вы можете найти полный список свойств слоя.

Порядок слоев

Слои карты в коллекции Layers отображаются в порядке увеличения индекса (т.е., Layers(1) это верхний слой, Layers(2) слой находящийся под слоем Layer(1), и т.д.), первым рисуется самый нижний слой карты и последним самый верхний. Важно установить порядок Ваших слоев правильно.

Например, предположим, Вы имеете точечный слой клиентов, и полигональный слой кварталов города. Если слои расположены в коллекции неверно, MapX сначала нарисует слой клиентов, а затем слой с границами кварталов, которые закроют точечный слой.

Вы можете изменить порядок следования слоев в карте на этапе проектирования, используя [Страницу свойств](#), как мы обсуждали это ранее. На закладке [Слои](#), выберите слой(и) порядок следования которого Вы хотите изменить и используйте кнопку [Вверх](#) или [Вниз](#) чтобы переместить слой(и) вверх или вниз относительно его текущего положения.

Чтобы дать пользователям возможность переупорядочивать слои во время выполнения приложения используйте метод [LayersDlg](#) отображающий диалог [Управление слоями](#), как обсуждалось ранее.

Порядок слоев также важен, когда Вы используете инструмент Select. Инструмент [Select](#) выбирает объекты из самого верхнего [Выбираемого](#) слоя. Если Вы имеете несколько объектов в одном и том же месте, может быть трудно выбрать именно тот объект, который Вы хотите. Вы можете переупорядочить слои таким образом, чтобы слой из которого Вы хотите выбрать объект был самым верхним. Инструменты обсуждаются в главе: "[Инструменты](#)".

Информация о слоях

Существует много случаев, когда вам потребуется получить информацию о слоях в коллекции. Здесь приведен простой фрагмент кода, который позволяет Вам последовательно пройти через все слои коллекции, определить типы представленных слоев и отобразить эту информацию в окне сообщений.

```
Dim lyr as Layer
For Each lyr in Map1.Layers
Select Case lyr.Type
    Case miLayerTypeNormal
        MsgBox "Layer " & lyr.Name & " обычный
слой"
    Case miLayerTypeRaster
        MsgBox "Layer " & lyr.Name & "
растровый слой"
    Case miLayerTypeSeamless
        MsgBox "Layer " & lyr.Name & " сшитый
слой"
    Case miLayerTypeUnknown
        MsgBox "Layer " & lyr.Name & "
неизвестный тип слоя"
    Case miLayerTypeUserDraw
        MsgBox "Layer " & lyr.Name & " user
draw слой"
    Case miLayerTypeDrilldown
        MsgBox "Layer " & lyr.Name & "
drilldown слой"
Next
```

В этом фрагменте кода используются предопределенные константы MapX-для всех типов слоев. Эти константы имеют общее название **LayerTypeConstants**. Все константы MapX перечислены в файле справки и *Справочном руководстве*.

Проверка типов объектов на слое

Этот фрагмент кода позволяет Вам последовательно пройти через все слои коллекции, используя индекс. Программа проверяет каждый слой в коллекции и определяется тип объектов представленных на слое. Объекты обсуждаются в этой главе далее.

```
Dim i as Integer
For i = 1 To Map1.Layers.Count
Select Case Map1.Layers(i).PredominantFeatureType
    Case miFeatureTypeRegion
        MsgBox "Layer " & lyr.Name & " содержит
регионы"
    Case miFeatureTypeLine
        MsgBox "Layer " & lyr.Name & " содержит
линии"
    Case miFeatureTypeSymbol
        MsgBox "Layer " & lyr.Name & " содержит
символы"
    Case miFeatureTypeMixed
        MsgBox "Layer " & lyr.Name & " содержит
смешанные объекты"
    Case miFeatureTypeUnknown
        MsgBox "Layer " & lyr.Name & " содержит
неизвестные объекты"
    Case miFeatureTypeText
```



```
MsgBox "Layer " & lyr.Name & "содержит  
текстовые объекты"
```

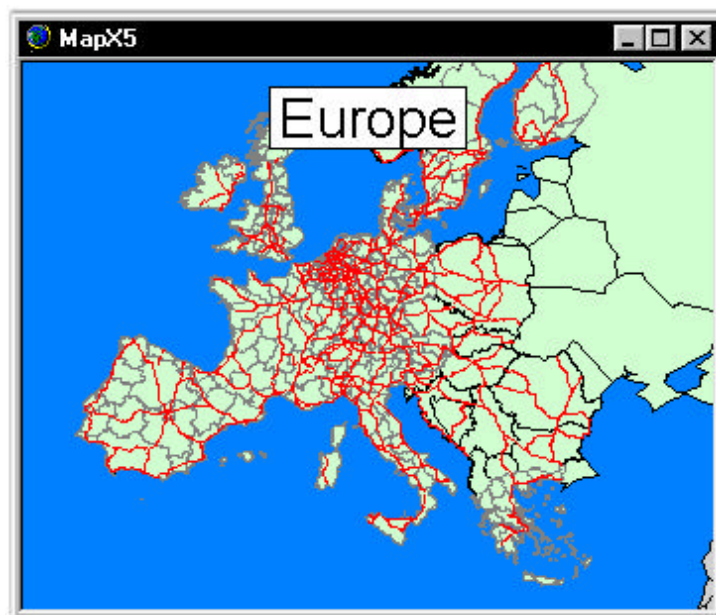
Next

Коллекция слоев имеет индекс по основанию единица, поэтому мы выполним итерацию от индекса равного единице до числа равного числу слоев в коллекции Layers. Мы также определили свойство **PredominantFeatureType** каждого слоя в коллекции, используя константы **FeatureType** для оценки типа.

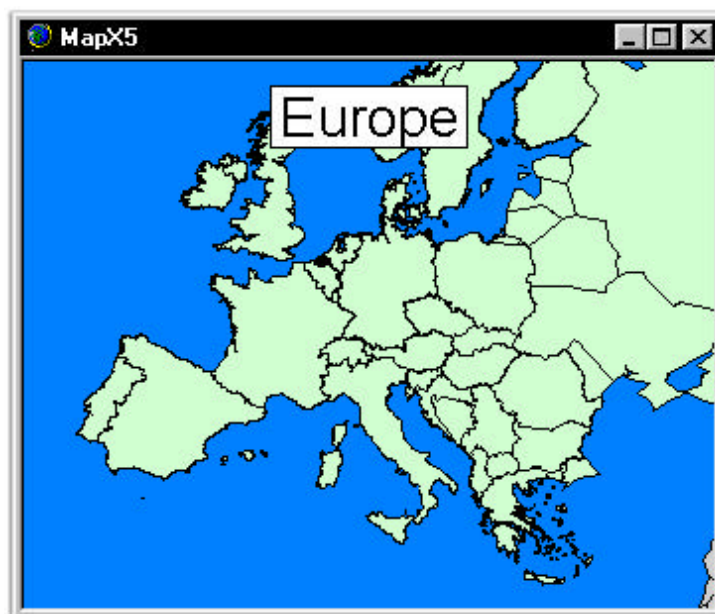
Масштабный эффект

Иногда Вам может потребоваться отображать слой только при определенном масштабе. Масштабный эффект позволяет Вам увидеть слой карты, когда масштаб карты попадет в заранее определенный диапазон. Вы можете установить различный масштабный эффект для каждого слоя.

Например, Ваша карта включает слой улиц, и при значительном уменьшении карты улицы становятся нечитаемыми.



используя масштабный эффект, Вы можете настроить свою карту таким образом, чтобы MapX автоматически скрывал слой улиц, если карта уменьшена до такого размера, что охватывает более 5 миль.



Нижеследующий пример кода добавляет слой к карте используя метод **Add** коллекции **Layer** и устанавливает масштабный эффект используя свойства объекта **Layer**.

```
Dim lyrStreets As Layer           `Создаем объект слой
`Устанавливаем таблицу как объект слоя и помещаем
слой на карту под номером 3.

Set lyrStreets = Map1.Layers.Add("Streets.tab", 3)
lyrStreets.ZoomLayer = True       `Включаем
масштабный эффект
lyrStreets.ZoomMin = 0            `Устанавливаем
мин. масштаб-0 миль
lyrStreets.ZoomMax = 5           `Устанавливаем макс.
Масштаб-5 миль
```

После того, как масштабный эффект для слоя установлен, когда масштаб карты будет попадать между минимальным и максимальными установленными уровнями, слой будет отображаться на карте. Вы можете изменять масштаб карты, используя инструмент **zoom** (обсуждается в главе 7) или метод **ZoomTo** объекта **Map**. В методе **ZoomTo** Вы устанавливаете масштаб и x, y координаты центра карты.

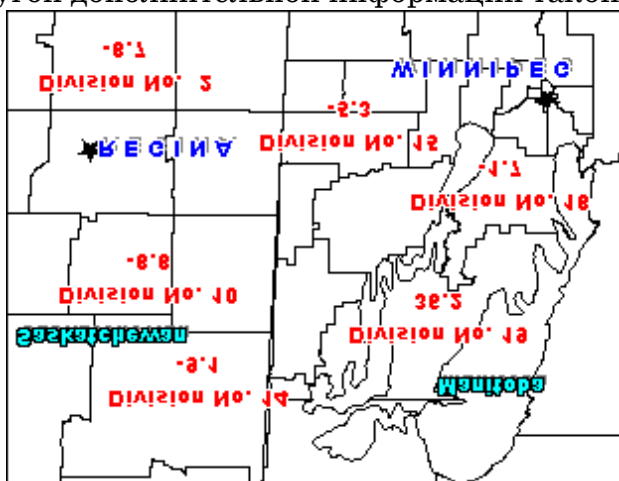
```
Map1.ZoomTo 3, -70.26, 44.05
```

Поскольку значение 3 мили попадает в заданный масштабный диапазон слой **Streets layer** будет центрирован по точке -70.26, 44.05 и отображать 3 мили по ширине окна карты.

Различные слои одной и той же карты могут отображаться при различных масштабах. Например, у Вас имеется слой улиц, слой границ графств и слой границ штатов. Вы хотите чтобы слой улиц был видимым только когда масштаб меньше чем 8 миль, слой графств отображался при масштабе от 20 до 200 миль, а слой штатов был виден только при масштабе более чем 100 миль. Вы можете установить различный масштабный эффект для каждого слоя Вашей карты.

Создание подписей для слоя

MapX имеет богатые возможности для подписывания географических объектов слоя атрибутивной информацией. Местоположение подписей базируется на положении центроидов географических объектов и другой дополнительной информации такой как [anchor point](#) и [offset](#).



Как атрибуты, подписи динамически связаны с их картографическими объектами. Если слой закрывается или делается невидимым, подписи также больше не отображаются. Если данные или географическая информация изменяются, подписи также изменяются. Если Вы создали подписи на основе выражения и изменили выражение, то подписи динамически будут заменены новыми.

Содержание подписи всегда определяется данными, ассоциированными с географическим объектом, независимо от того, как были созданы подписи автоматически, с использованием инструмента [Подпись \(Label\)](#) или метода [LabelAtPoint](#).

В дополнение к содержанию подписи, Вы можете контролировать положение, отображение и вид автоматически помещаемых подписей используя свойства объекта **LabelProperties**. Вы можете определить условия, при которых подписи будут отображаться, стиль их отображения, и местоположение всех подписей слоя относительно соответствующих им объектов.

Контроль отображения подписей

Чтобы автоматически нанести подписи на слой используйте установите свойство слоя **AutoLabel** равным **True**.

Каждый объект **Layer** имеет свойство **LabelProperties**, контролирующее различные аспекты подписей. Например, чтобы скрыть все подписи слоя, присвойте свойству **LabelProperties.Visible** значение **False**. Чтобы установить максимальное количество подписей отображаемых на карте установите свойство **LabelMax**. Свойство **Duplicate** контролирует могут ли объекты с одинаковыми именами, иметь повторяющиеся подписи на карте. Свойство **Overlap** разрешает или запрещает наложение подписей друг на друга; присвоив свойству **Overlap** значение **True**, Вы можете обнаружить, что большее число объектов подписано, но подписи из-за наложения стали трудно читаемыми.

Масштабирование подписей

Вы можете настроить подписи таким образом, чтобы они появлялись только в определенном масштабном диапазоне, подобно тому, как весь слой может быть видимым только при определенном масштабе. Чтобы указать масштабный диапазон для подписей установите для слоя свойства **LabelZoom**, **LabelZoomMax**, и **LabelZoomMin** для объекта **LabelProperties**.

Размещение подписей

Чтобы контролировать местоположение подписей используйте свойство **Position** (которое контролирует, где подпись помещается выше, ниже или сбоку от центроида объекта), и свойство **Offset** (определяющее насколько подпись удалена от объекта).

По умолчанию место прикрепления подписи зависит от предопределенного типа объекта слоя. Например, слой с полигонами по умолчанию имеет подписи, прикрепленные к центроидам полигонов.

Для объектов линия/полилиния (таких как слой улиц), Вы можете установить режим подписывания параллельно линейному объекту, задав свойству **Parallel** значение **True**.

Создание Указок

Указки это подписи с линией указывающей на объект к которому относится подпись. Это может быть весьма полезным, когда имеется много подписей на небольшой территории. Например, Вы хотите подписать карту Азии. На ней имеется много небольших стран, находящихся относительно близко друг к другу. Если Вы попытаетесь подписать эти страны, подписи будут накладываться и станут трудночитаемыми.

Чтобы отобразить линии указок, установите для свойства **LabelProperties.LineType** значение **miLineTypeSimple (1)** или **miLineTypeArrow (2)** для слоя который Вы хотите подписать.

Стиль подписей

Чтобы сделать изменение стиля всех подписей.

Вы так же можете сделать соответственное стилевое оформление. Внесите нужные Вам изменения. Когда Вы вернетесь в окно карты, выделенные подписи изменят свой стиль.

Можно также настроить фон под подписью. Можно сделать рамку вокруг подписи.

Интерактивное подписывание

Хотя в большинстве случаев Вы, вероятно, будете использовать автоматическое подписывание, в ряде случаев Вы может потребоваться создавать подписи по одной за раз, используя инструмент **Label**.

Инструмент **Label** один из встроенных стандартных инструментов MapX. Для активизации инструмента установите свойство

Map.CurrentTool равным **miLabelTool (1010)**; пользователи в этом случае получат возможность подписывать объекты карты щелчком на них.

Самый простой путь удалить все подписи в слое, использовать настройку **Visibility** (видимость). Это заблокирует прорисовку всех подписей на слое и автоматически сделанных подписей и подписей, созданных вручную. Что бы очистить только подписи, созданные пользователем (подписи нанесенные с использованием инструмента **Label**), используйте метод **ClearCustomLabels**.

```
Dim DS As Object
Dim DB As Object
Dim RS As Object
Dim Temp As Object
Set DB =
Workspaces(0).OpenDatabase("Mapstats.mdb")

Set RS = DB.OpenRecordset("USA")
Set DS = Map1.Datasets.Add(miDataSetDAO, RS)
Set Map1.Layers("usa").LabelProperties.Dataset =
DS
Set Temp = DS.Fields("GEONAME")
Set Map1.Layers("usa").LabelProperties.DataField =
Temp
```

Аннотации

Хотя подписывание объектов (labeling) решает большинство Ваших потребностей для работы с текстом, Вам может потребоваться создать текстовый объект, чтобы аннотировать карту. В отличие от подписей, текстовые аннотации не связаны с данными, Вы можете поместить аннотацию в любом месте карты, даже там где вообще нет объектов.

Чтобы добавить текстовую или символьную аннотацию к Вашей карте используйте методы и свойства объекта и коллекции **Annotations**.

Каждый объект Map имеет коллекцию **Annotations** (свойство **Map.Annotations**). Аннотации, будь то символы или текстовые объекты всегда рисуются на самом верху карты.

Аннотации обычно используются, чтобы поместить на карту текстовые сообщения или символы. Аннотации масштабируются когда масштаб карты изменяется. Аннотации не связаны с каким-либо конкретным слоем, и всегда помещаются наверху.

Обратите внимание на то, что объект **Annotation** не имеет свойств для установки местоположения, стиля символа и текста. Чтобы управлять этими атрибутами аннотаций, Вы можете использовать свойство **Annotation.Graphic** чтобы получить графический объект, а уже затем модифицировать свойства этого объекта.

Растровые изображения

Растровое изображение это тип компьютерной картинке, в которой изображение формируется рядами маленьких точек (пиксел).

Растровые изображения иногда еще называют **bitmaps**.

Аэрофотографии и спутниковые снимки - это наиболее часто используемые типы растровых данных в ГИС.

Отображение растра как слоя карты

В MapX Вы можете отображать растры как задний фон для карт, которые Вы создали. Однако сначала растровое изображение должно стать частью таблицы MapInfo. Затем Вы можете наложить поверх растрового изображения другую информацию, такую как сеть улиц, размещение покупателей и т.п.



Хотя растровое изображение может содержать карту, с ним нельзя ассоциировать данные. Оно предназначено для просмотра, как подложка.

Как показать растровое изображение

Каждое растровое изображение должно иметь соответствующий '.ТАВ' файл, в котором хранятся географические координаты изображения. .ТАВ файл создается во время регистрации изображения в MapInfo Professional.

Примечание: Когда Вы отображаете растровое изображение как слой карты, MapX автоматически устанавливает разворот и проекцию всех векторных слоев карты таким образом, чтобы достичь соответствия развороту и проекции растрового изображения.

Анимационный слой

Анимационный слой используется, когда объекты карты требуют частого обновления, например приложения работающие в реальном режиме времени. Например, Вы можете разработать приложение для управления парком автомашин, в котором точечными объектами будет представлено текущее местоположение каждой автомашины. Вы можете получать текущие координаты автомобиля, используя технологию GPS (Global Positioning Satellite), и затем обновлять точечные объекты, показывая текущее местоположение машины на карте. Для таких типов приложений, где объекты карты постоянно изменяются, значительно удобней использовать обновление и хранение данных, в анимационном слое, а не обычном, так как перерисовка карты будет происходить значительно быстрее.

Изначально `AnimationLayer` установлен равным нулю. Вы можете присоединить объекту `Layer` свойство, превращающее его в анимационный слой (это может быть обычный слой или `user draw` слой). Когда слою присвоено свойство `AnimationLayer`, он рисуется поверх всех слоев, включая слой аннотаций и запросов. При этом местоположения слоя в коллекции слоев остается прежним. “Летающие” объекты типа легенды будут отображаться сверху анимационного слоя, хотя они не будут перерисовываться каждый раз, когда перерисовывается слой анимации. Если обычный слой используется как анимационный, запросы (`selections`) и подписывание будут работать.

Пример

```
Set Map.Layers.AnimationLayer = Layers(3)
```

Свойство может быть использовано для определения какой слой в данный момент является анимационным (если существует вообще):

```
for each lyr in Map.Layers
    if Map.Layers.AnimationLayer = lyr the ...
end if
next
```

Чтобы выключить слой анимации присвойте ему значение ноль:

```
Set Map.Layers.AnimationLayer = nothing
```

Анимационный слой станет нормальным слоем, находящемся в том же месте в списке слоев.

Собственные слои пользователя

Метод **AddUserDraw Layer** коллекции **Layers** дает разработчику возможность создавать свои собственные слои карты. Он используется совместно событием **DrawUserLayer**, которое происходит, когда слою необходима перерисовка. Этот метод возвращает вновь созданный объект **Layer**. Вы можете иметь любое количество собственных слоев.

Как это работает

Сначала Вы добавляете пользовательский слой к Вашей коллекции слоев:

```
' this sets the UserDraw Layer to "My Layer"
Dim lyr as Layer
Set lyr = Map1.Layers.AddUserDrawLayer("My Layer",
1)
```

Затем Вы помещаете код для рисования слоя в событие **DrawUser Layer**. Когда Ваше приложение создает косметический (**UserDraw**) слой, используя метод **AddUserDrawLayer** коллекции слоев, то в приложение передается событие, когда окну требуется перерисовка.

Полный пример показан ниже.

```
' API DEFS должна быть декларирована в отдельном
модуле

Declare Function MoveToEx Lib "gdi32" Alias
"MoveToEx" (ByVal hdc As Long, ByVal x As Long,
ByVal y As Long, lpPoint As POINTAPI) As Long

Declare Function LineTo Lib "gdi32" Alias "LineTo"
(ByVal hdc As Long, ByVal x As Long, ByVal y As
Long) As Long
```

```

Declare Function SetMapMode Lib "gdi32" Alias
"SetMapMode" (ByVal hdc As Long, ByVal nMapMode As
Long) As Long
Type POINTAPI
    x As Long
    y As Long
End Type
Public Const MM_TWIPS = 6

' устанавливаем как UserDraw слой "My Layer"
Dim lyr as Layer
Set lyr = Map1.Layers.AddUserDrawLayer("My Layer",
1)
' Этот пример рисует линию между углами Wyoming
Private Sub Map1_DrawUserLayer(ByVal Layer As
Object, ByVal hDC As Long, ByVal RectFull As Object,
ByVal RectInvalid As Object)
Dim pt As POINTAPI

SetMapMode hDC, MM_TWIPS
dim PX as single
dim PY as single
X1 = -111.0542
Y1 = 45.0009
X2 = -104.0528
Y2 = 41.0018

if map1.ClipLine(X1,Y1,X2,Y2) then
    map1.ConvertCoord(PX, PY, X1,Y1, miMapToScreen)
    MoveToEx hDC , PX, -PY, pt
win api call
    map1.ConvertCoord(PX, PY, X2,Y2, miMapToScreen)
    LineTo hDC, PX, -PY ' win api call
end if

End Sub

```

7

Объекты и выборки

- ◆ Объекты и выборки
- ◆ Использование коллекции Features
- ◆ Коллекция Selection
- ◆ Редактирование объектов

Методы объектов **Selections** и **Features** позволяют Вам “пометить” или выбрать объекты отвечающие определенному критерию. Точка на карте представляющая город New York City это пример объекта **Feature**. Скажем, вам необходимо найти всех потенциальных клиентов проживающих в 5 милях от Sheep’s Head Bay, Brooklyn. После того, как Вы создали эту коллекцию данных, их можно обрабатывать, распечатывать, считать среднее, сохранять в файл или делать другие действия. В этой главе мы детально рассмотрим объекты и выборки.

Что такое объект карты?

Объект карты (feature) это картографический объект, представленный на карте точкой, линией или регионом. Например, карта США может содержать регионы для штатов, линии для автострад и точки для городов. В MapX, объект карты представлен объектом **Feature**.

Например, штат New York может быть объектом **Feature** типа регион, interstate 95 объектом **Feature** типа линия, а город New York объектом **Feature** типа точка.

Что такое коллекция объектов?

В MapX, различные слои образующие карту, обычно имеют в каждом слое однотипные объекты. Например, слой “US States” содержит объекты регионы, представляющие каждый штат, слой “US Highways” содержит линейные объекты представляющие автострады США, и слой “US Capitols” содержит точечные объекты представляющие столицы штатов. В MapX, все объекты слоя карты, или некоторая часть этих объектов представляется как коллекция **Features**. Многие методы объекта **Layer** возвращают коллекцию **Features**. Объекты из коллекции **Features** не “подсвечиваются” на Вашей карте, для этого используется коллекция **Selection**.

Что такое коллекция запросов?

Подобно коллекции **Features**, коллекция **Selection** также является коллекцией объектов **Feature**. Таким образом, коллекция **Selection** представляет объекты **Feature** которые в данный момент выбраны (или выбраны, когда пользователь щелкает на карте инструментом выбора, или выбраны различными методами выборки. Каждый объект **Layer** имеет собственную коллекцию **Selection (Layer.Selection)**. MapX автоматически выделяет все объекты в коллекции **Selection**.

Использование коллекции объектов (Features Collection)

Коллекция **Features** похожа на коллекцию **Selection**, обе коллекции являются коллекциями объектов **Feature**. Однако коллекция **Features** имеет другой набор методов и свойств и эти два типа коллекций имеют различное поведение. Выбранные объекты автоматически “подсвечиваются”, а объекты в коллекции **Features** нет.

Получение объектов на слое

Прежде чем делать что-нибудь с коллекцией **Features**, Вы должны создать ее. Методы объекта **Layer** создают коллекцию объектов **Feature**.

Следующие методы коллекции **Layers** обеспечивают различные пути получения коллекции **Features**.

Метод	Описание	Пример кода (Dim fs as Features ‘this creates a collection of Features)
AllFeatures	Возвращает коллекцию Features, содержащую все объекты слоя.	Set fs = Map1.Layers(2) _ .AllFeatures
NoFeatures	Возвращает пустую коллекцию Features для слоя.	Set fs = Map1.Layers(9).NoFeatures
SearchWithinDistance	Возвращает коллекцию Features, содержащую объекты, находящиеся не далее указанного	Set fs = Map1.Layers(3). _ SearchWithinDistance (objPoint, _ 36.5, miUnitMile, _ miSearchTypeCentroidWithin)

	расстояния от точечного объекта.	
SearchWithinFeature	Возвращает объект состоящий из объектов, находящихся в указанном объекте полигон.	Set fs = Map1.Layers(3). _ SearchWithinFeature _ (ftr, miUnitMile, _ miSearchTypeCentroidWithin)
SearchWithinRectangle	Возвращает коллекцию объектов попадающих в заданный прямоугольник.	Set fs = Map1.Layers(3). _ SearchWithinRectangle(miRect, _ miUnitMile, _ miSearchTypePartiallyWithin)
SearchAtPoint	Возвращает коллекцию объектов состоящую из объектов расположенных в указанной точке.	Set fs = Map1.Layers(3) .SearchAtPoint(objPoint)

Search Type Constants

MiSearchTypeCentroidWithin	Включает в поиск если центроид объекта находится внутри полигона.
MiSearchTypeEntirelyWithin	Включает в поиск, если объект находится в полигоне.
MiSearchTypePartiallyWithin	Включает в поиск, если любая часть объекта попадает внутрь полигона.

Следующий код является инструментом, использующим инструмент, который создает коллекцию объектов, находящихся в той точке, где пользователь щелкнул мышкой.

```

Private Sub Map1_ToolUsed(ByVal ToolNum As Integer,
ByVal X1 As Double, ByVal Y1 As Double, ByVal X2 As
Double, ByVal Y2 As Double, ByVal Distance As
Double, ByVal Shift As Boolean, ByVal Ctrl As
Boolean, EnableDefault As Boolean)

    Dim ftr As Feature      'создает объект Feature
    Dim fs As Features      'создает коллекцию
объектов
    Dim pt As New Point      'создает новую точку

    'the new point created has coordinate values set
when the
    user clicks on a feature
    pt.Set X1, Y1
    'If the tool used is the custom info tool then
create a
    collection of features from the States layer
    'and print the State Name feature into a message
box.
    If ToolNum = InfoTool Then
        'create the collecton of features at the point
the user clicked
        Set fs = Map1.Layers("USA").SearchAtPoint(pt)
        'print the State Name feature into a
message box.
        For Each ftr In fs
            MsgBox ftr.Name
        Next
    End If

```

Манипулирование коллекцией объектов

Следующие методы позволяют Вам манипулировать коллекцией **Features**, добавляя, удаляя, клонируя объекты в коллекции:

Метод	Описание
Add	Добавляет объект Feature или целую коллекцию Features в существующую коллекцию.
Clone	Делает копию коллекции как другой объект коллекции Features.
Common	Комбинирует эту коллекцию и другую коллекцию Features так что эта коллекция содержит только объекты, которые есть в обеих коллекциях (операция INTERSECT).
Remove	Удаляет объект Feature или все объекты из коллекции Features (операция SUBTRACT).
Replace	Перемещает содержимое коллекции с объектом Feature или все объекты из объекта коллекции Selection.

Объект Feature

Коллекция **Features** сделана из коллекции объектов **Feature**. Объект **Feature** соответствует объекту карты, например, символу, линии или полигону.

Объект **Feature** соответствует объектам на карте, таким как New York, Chicago, Louisiana, Cortland County или Highway I-10.

Автономные объекты

Методы объекта **Feature** позволяют Вам создавать и манипулировать автономными объектами. Когда Вы создаете автономный объект, Вы должны присоединить этот объект к карте прежде чем обращаться к его свойствам и методам. Прикрепление объекта к карте ассоциирует систему координат карты с объектом.

Следующий пример Visual Basic показывает как создавать автономные текстовые объекты, и затем менять его текст.

```
Dim f As New Feature
Dim fNew As Feature
' Add a new text object to layer 1
f.Attach Map1
f.Type = miFeatureTypeText
f.Point.Set Map1.CenterX, Map1.CenterY
f.Caption = "This is a text object"
Set fNew = Map1.Layers(1).AddFeature(f)

' Change the text caption
fNew.Caption = "Changed Text"
fNew.Update
```

Следующий пример Visual Basic демонстрирует как прикреплять автономный объект.

```
Dim f as new Feature
f.Attach Map1
f.Type = miFeatureTypeSymbol
f.Point.Set Map1.CenterX, Map1.CenterY
Map1.Layers(1).AddFeature f
```

Свойства объекта Feature

Следующие свойства определяют объект карты:

Свойство	Описание	Пример кода
CenterX	Координата X центроида объекта.	Print ftr.CenterX
CenterY	Координата Y центроида объекта.	MsgBox ftr.CenterY
FeatureID	Идентификатор ID объекта.	IVar = ftr.FeatureID

	Каждый объект в слое имеет его уникальный для данного слоя ID. Это значение целого типа.	
Length	Содержит длину объекта.	Print ftr.Length
Perimeter	Периметр объекта.	If ftr.Perimeter > 500 Then Print "Too Long" End If
Name	Имя объекта.	MsgBox ftr.name
Type	Тип объекта (точка, линия и т.п.).	ftr.Type = miFeatureTypeSymbol

Коллекция Selection (запросов)

Фундаментальная возможность MapX – выбор объектов карты, для осуществления дальнейших операций над ними. Пользователь может щелчком на карте выбрать один или более объектов (точек, линий, регионов и т.п.). MapX выделяет цветом все выбранные объекты.

Для проверки списка выбранных объектов, используйте коллекцию Selection, которая является коллекцией объектов **Feature**. Коллекция **Selection** также представляет различные методы (такие как **SelectByRadius**) которые позволяют реализовать различные типы выборки, например, выбор всех объектов, находящихся на определенном расстоянии от города (в окружность определенного радиуса). Выбранные объекты будут “подсвечены” на карте (выделены определенным цветом). Каждый слой имеет коллекцию выбранных объектов (**Layer.Selection**).

Коллекция Selection имеет методы для добавления или удаления объектов из коллекции. Таким образом, если Вы имеете существующие коллекции Selection или Features, Вы можете добавлять, удалять, копировать или искать пересечение двух коллекций.

В таблице представлены некоторые методы коллекции Selection.

Полный список смотрите в *Справочнике* или в *Электронной справке*:

Метод	Описание	Пример кода
ClearSelection	Отменяет выбор всех объектов этого слоя. Используйте Layers.ClearSelection чтобы очистить выборку во всех слоях.	Map1.Layers(“SalesReps1997”).Selection.ClearSelection

Clone	Делает копию коллекции как другой объект коллекции Selection.	Map1.Layers(2).Selection.Clone ftrs
Common	Комбинирует коллекцию и другой объект Selection так что эта коллекция содержит только те объекты, которые есть в обеих коллекциях (операция INTERSECT).	Map1.Layers(2).Selection.Common on Map1.Layers(4).Selection
Remove	Удаляет объект Feature или все объекты из объекта Selection из этой коллекции (операция SUBTRACT).	Map1.Layers(2).Selection.Remove fs
Replace	Перемещает содержимое коллекции с объектом Feature или все объекты из объекта коллекции Selection.	Map1.Layers("Boston").Selection.Replace lyr.AllFeatures
SelectAll	Выбирает все объекты в слое.	Map1.Layers("Cargo").Selection. .SelectAll
SelectByPoint	Выбирает объекты в указанной точке слоя.	Map1.Layers(5).Selection.Select ByPoint 75.14, 42.9, miSelectionAppend
SelectByRadius	Выбирает объекты из слоя попадающие в заданный радиус от точки.	Map1.Layers(4). Selection.SelectByRadius X, Y, Radius, miSelectionnew
SelectByRectangle	Выбирает объекты из слоя попадающие в заданный прямоугольник.	Map1.Layers(5).Selection.Select ByRectangle 98.7, 31.56, - 75.14, 42.9, miSelectionRemove
SelectByRegion	Выбирает объекты из слоя	Selection.SelectByRegion

	попадающие в заданный регион.	Layer, FeatureID, Flag
--	-------------------------------	------------------------

Константы Search Type

Следующие константы определяют, что делать с результатом Вашего запроса.

MiSelectionNew	Создать новую выборку.
MiSelectionAppend	Добавить к текущей выборке.
MiSelectionRemove	Удалить из текущей выборки.

Следующий код создает выборку в указанной точке:

```
Private Sub Command1_Click()  
    Map1.Layers(5).Selection.SelectByPoint -98,  
    31.56, miSelectionNew  
End Sub
```

Событие SelectionChanged

Это событие происходит, когда выборка изменяется. Это позволяет приложению определить, что выборка сделанная на карте изменилась, и отреагировать соответствующим образом. Выборка может измениться в результате применения пользователем инструмента selection, или использованием метода **Selection** объекта Layer.

```
Private Sub Map1_SelectionChanged()  
    ' *****  
    *****  
  
    'Any time the selection changes, update the listbox  
    with  
    'the currently selected records  
  
    ' *****  
    *****
```



```

Dim ftr As Feature      'Создание объекта feature
Dim lyr As Layer        'Создание объекта layer
'Within the current layer the user is selecting
from, print out a message
For Each lyr In Map1.Layers
For Each ftr in lyr.Selection
        MsgBox ("found: ") ftr.name
    Next
Next
End Sub

```

Редактирование объектов

MapX позволяет Вам создавать, изменять или удалять объекты слоя карты (точки, линии, регионы и т.п.). Любой слой основанный на стандартной таблице MapInfo (.tab) может быть отредактирован.

Примечание: MapX не поддерживает обработку транзакций.

Другими словами, когда Вы вносите изменения в объекты, добавляете/удаляете/редактируете, это происходит немедленно. Вам не требуется сохранять изменения позднее, но Вы не имеете возможности отменить операцию изменения.

Как создать новый объект

Существует два пути для создания новых объектов карты. Вы можете создать объект помещением нового объекта **Feature**, или создать объекты посредством операции (такой как построение буферной зоны) над существующими объектами, с использованием методов **FeatureFactory**. Этот тип объектов называется автономными объектами (**stand-alone**). Автономные объекты имеют некоторые ограничения: они не могут быть добавлены к какой-либо коллекции, и только те методы и свойства, которые использовались для определения допустимых объектов. Например, Вы не можете использовать свойство **Area** для автономного объекта, и автономный объект не имеет значения для свойства **Feature.Layer**.

Объект, который приходит из слоя, не является автономным и все свойства и методы будут работать.

Размещение нового объекта Feature

Примеры на Visual Basic

Следующий пример на Visual Basic создает символьный объект.

```
Dim newobj as new MapXLib.Feature 'автономный объект
Dim obj as MapXLib.Feature 'to hold feature added
to layer
newobj.Type = miFeatureTypeSymbol
'Use the map's default symbol style
newobj.Style = map1.DefaultStyle
' specify the x- and y-coordinates (long, lat)
newobj.Point.Set -104.3452, 34.91234

'now add the object to a layer
' obj will be the newly added object
' you can still use newobj to add more objects
set obj = map1.Layers(1).AddFeature(newobj)
```

Примеры на C++

Следующий пример C++ создает объект символ:

```
void CSampleProjectView::CreateSymbol() {
    CMapXFeature newobj; // Standalone object
    CMapXFeature obj; // to hold object added to layer

    // Create the new object and attach it to the
    map's coordinate system
    if(!newobj.CreateDispatch(newobj.GetClsid())) {
        TRACE0("Failed to Create Feature object");
        return;
    }
}
```

```

try {
    newObj.Attach(m_Map.GetDispatch());

    newObj.SetType(miFeatureTypeSymbol);
    // Use the map's default symbol style
    newObj.SetStyle(m_Map.GetDefaultStyle());
    newObj.GetPoint().Set(-104.3452,34.91234);
    // Now add the object to the layer
    // obj is the newly added object
    // you can still use newObj to add more objects
    obj =
m_Map.GetLayers().Item(1).AddFeature(newObj);
} catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
} catch (COleException *e) {
    e->ReportError();
    e->Delete();
}
}

```

Чтобы создать объект другого типа, установите тип **miFeatureTypeRegion**, **miFeatureTypeLine** или **miFeatureTypeText** и установите свойства приложимые к этому типу.

Методы FeatureFactory

Методы объекта **FeatureFactory** позволяют создавать новые объекты карты, или создавать объекты которые являются результатом операции (такой как построение буферной зоны) над существующими объектами.

Ниже перечислены методы объекта **FeatureFactory**:

- BufferFeatures
- CombineFeatures
- CreateArc
- CreateCircularRegion
- CreateEllipticalRegion
- CreateLine
- CreateRegion
- CreateSymbol
- CreateText
- EraseFeature
- IntersectFeatures
- IntersectionPoints
- IntersectionTest

Большинство этих методов возвращает самостоятельные объекты. Эти объекты автоматически присоединяются к карте (т.е., они уже ассоциированы с координатной системой). Другими словами, Вам не надо использовать метод **Attach** для объектов, возвращаемых этими методами.

Для получения объекта **FeatureFactory**, обратитесь к свойству **Map.FeatureFactory**.

```
Dim f As MapXLib.Feature
Dim p As New MapXLib.Point

' Use the map's current center as the point
coordinates
p.Set Map1.CenterX, Map1.CenterY

' создает текстовый объект и добавляет его к слою
(layer 1)
```

```
Set f =  
Map1.Layers(1).AddFeature(Map1.FeatureFactory.Create  
Text(p, "Some Text"))
```

Электронная справка и Справочник дают подробные инструкции как использовать различные методы **FeatureFactory**.

Как изменить существующие объекты

Когда Вы имеете указатель на объект **Feature**, он представляет реальный объект на слое; объект, который поступает из слоя не является автономным и все свойства и методы работают. Любые свойства и методы отражают объект на слое.

После начала изменения объекта путем настройки стилей или точек его прикрепления, Ваши изменения не возымеют эффект до тех пор пока Вы не 'обновите' объект. Такой подход позволяет делать много изменений объекта без ожидания обновления базы данных и перерисовки экрана после каждого изменения. Побочный эффект в том, что MBR, площадь, длина, и т.п. не будут изменены до тех пор пока Вы не обновите объект.

Один из следующих способов позволит Вам обновить объект:

- Если Вы изменили объект и хотите сохранить изменения, используйте метод **Feature.Update**.
- Если Вы хотите заменить объект другим объектом, используйте метод **Layer.UpdateFeature**.
- Когда объект обновлен, и старый MBR и новый MBR становятся недействительными, так что экран перерисовывается корректно.

Пример использования **Feature.Update** в Visual Basic

```
' Shift all selected objects by 1 degree  
Dim obj as MapXLib.Feature  
  
for each obj in map1.layers(1).selection
```

```

    obj.Offset(1.0, 0.0)
    obj.Update()
next

```

OR

```

for each obj in map1.layers(1).selection
    obj.Offset(1.0, 0.0)
    map1.layers(1).UpdateFeature(obj, obj)
next

```

Пример Feature.Update в C++

```

// Shifts all selected objects by 1 degree
void CSampleProjectView::ShiftObjects() {
    long i,ObjCount;
    CMapXFeature obj;
    try {
        ObjCount =
m_Map.GetLayers().Item(1).GetSelection().GetCount();
        for(i=1;i<=ObjCount;i++) {
            obj =
m_Map.GetLayers().Item(1).GetSelection().Item(i);
            obj.Offset(1.0,0.0);
            obj.Update();
        }
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

```

    }
}

//Or...
void CSampleProjectView::ShiftObjects2() {
    long i,ObjCount;
    CMapXFeature obj;
    try {
        ObjCount =
m_Map.GetLayers().Item(1).GetSelection().GetCount();
        for(i=1;i<=ObjCount;i++) {
            obj =
m_Map.GetLayers().Item(1).GetSelection().Item(i);
            obj.Offset(1.0,0.0);

            m_Map.GetLayers().Item(1).UpdateFeature(obj,obj);
        }
    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}
}

```

Если Вы выбираете не вызывать **Update** и обновить объект, в следующее время объект запрашивается из слоя и он будет иметь первоначальное значение.

Как удалить существующие объекты

Чтобы удалить объект используйте метод `Layer.DeleteFeature`. Объект и соответствующая ему строка будут удалены из таблицы MapInfo немедленно.

Пример `Layer.DeleteFeature` в Visual Basic

```
' удалить все выбранные из слоя USA объекты
Dim obj As MapXLib.Feature
Dim selectedFtrs As MapXLib.Features
Dim lyr As MapXLib.Layer

Set lyr = Map1.Layers("USA")
Set selectedFtrs = lyr.Selection
For Each obj In selectedFtrs
    lyr.DeleteFeature (obj)
Next
```

Пример `Layer.DeleteFeature` в C++

```
void
CSampleProjectView::DeleteAllSelectedObjectsFromLayer() {
    CMapXFeature obj;
    CMapXLayer lyr;
    long i, ObjCount;
    try {
        lyr = m_Map.GetLayers().Item("usa");
        ObjCount = lyr.GetSelection().GetCount();
        for(i=1; i<=ObjCount; i++) {
            obj = lyr.GetSelection().Item(i);
            lyr.DeleteFeature(obj);
        }
    }
```

```

    } catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    } catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

Как получить объект для редактирования

Вы можете позволить пользователю щелкнуть на объекте и выбрать его так. Затем, для доступа к выборке, Ваша программа может использовать коллекцию **Layer's Selection**. Коллекция **Selection** также представляет различные методы, позволяющие добавить объекты к коллекции, такие как **SelectByPoint**.

Объект **Layer** имеет различные методы (описанные выше), такие как **SearchAtPoint**, которые позволяют Вам получить коллекцию **Features**.

Вы можете использовать метод **Find.Search** чтобы осуществить запрос, который возвратит объект **FindFeature** (суперкласс объекта **Feature**).

Проверка частей полигонов или линий

Объект линия и объект полигон в MapX состоит из многих коллекций объектов **Point**. Эти коллекции коллекций **Points** доступны через коллекцию объекта **Feature.Parts**.

Создание объекта полилиния

Этот пример на Visual Basic демонстрирует, как использовать объект **Point**, коллекцию **Points** и объект **Parts** для создания объекта полилиния, добавляя по одному узлу.

```

' create a polyline feature with 3 points

```

```

Dim newobj as new MapXLib.Feature    'stand-alone
object
Dim obj as MapXLib.Feature    'to hold object added to
layer
Dim pts as new Points
Dim pt as new Point
newobj.Type = miFeatureTypeLine
'Use the map's default symbol style
newobj.Style = map1.DefaultStyle
'set the lines 3 points
pt.Set -101.023, 45.0452
pts.Add pt
pt.Set -102.023, 49.0452
pts.Add pt
pt.Set -100.0, 34.2564
pts.Add pt
newobj.Parts.Add pts
'now add the object to a layer
' obj will be the newly added object
set obj = map1.Layers(1).AddFeature(newobj)

```

Проверка узлов объекта

Этот пример на Visual Basic демонстрирует, как коллекция **Parts** и коллекция **Points** позволяют Вам в цикле пройти через все точки в выбранном объекте.

```

Dim obj as MapXLib.Feature
Dim pts as new Points
Dim pt as new Point
set obj = map1.layers(1).selection(1)

for each pts in obj.parts
  for each pt in pts
    debug.print pt.x, pt.y
  next
next

```

8

Поиск объектов на карте

- ◆ Поиск объектов на карте
- ◆ Объект Find
- ◆ Объект FindFeature

Метод **Find** объекта **Layer** позволяет осуществлять поиск на слое карты и находить определенные объекты на этом слое.

Объект Find

Объект **Object** позволяет найти местоположение объектов на карте. Вы можете найти интересующий Вас линейный, точечный или полигональный объект. Для того чтобы использовать метод **Find** слой, в котором Вы будете осуществлять поиск должен иметь поле с индексом. Смотрите Приложение: "*Использование утилиты Geodictionary Manager*" для более подробной информации о индексировании полей в таблице.

Например, если вы хотите найти город Albany в штате NY, Вы можете использовать метод **Find Object Search** для поиска городов в слое US _ Cities и слой штатов.

Метод	Описание	Пример кода
Search	Осуществляет поиск.	Set Found = Map1.Layers("US _ Cities").Find.Search("Albany", "NY")
SearchEx	Расширяет возможности поиска возвращая "ближайшее соответствие".	

Свойства объекта **Find** позволяют Вам установить параметры поиска.

Свойство	Описание
FindDataset	Поле Dataset с которым проводится сравнение. Если не указано, используется первичный ключ слоя.
FindField	Поле из Dataset с которым производится сравнение. Используется совместно со свойством FindDataset.

RefineDataset	Переопределяет, какие Dataset и Field будут использоваться вместо предыдущих.
RefineField	Поле для детализации. Берется объект Field.
RefineLayer	Слой, используемый для детального поиска. Берется объект Layer.
Abbreviations	Используется ли словарь аббревиатур (логическое).
ClosestAddr	Использование найденного совпадения с более подробным адресом (логическое).
OtherBoundary	Использует совпадение, найденное найденное в других границах, нежели указанное (логическое).

Уточнение границ

Уточнение границ в Вашем **Find.Search** используется для различения между множеством объектов с тем же именем. Например, если Вы путешествуете по Albany, Вы можете найти такое место и в New York, California или Georgia. Укажите штат, что бы уточнить границу “Albany, New York”

Следующий пример кода устанавливает слой USA в качестве уточняющей границы:

```
Private Sub Form_Load()  
    '*****  
    *****  
    'Set the USA to be the refining layer when doing the  
    find  
    '*****  
    Dim fdObject As MapXLib.FindFeature  
    'Set the Refining Layer  
    Set frmSelects.Map1.Layers("US Top 20 Cities") _  
    .Find.RefineLayer = frmSelects.Map1.Layers("USA")
```

Объект FindFeature

Метод **Find.Search** возвращает найденные объекты в виде объекта **FindFeature**. Объект **FindFeature** хранит свойства найденного объекта **Feature** как свои собственные свойства. В дополнение, объект **FindFeature** имеет свойство **FindRC** которое хранит результирующий код операции **Find**.

Объект **FindFeature** это суперкласс **Feature** с добавлением в виде более совпадающего стринга, который будет возвращен.

Свойство	Описание	Пример кода
FindRC	Результирующий код операции поиска.	MsgBox FoundObj.Find RC

Свойство **FindRC** это численное значение, показывающее почему объекты были или не были найдены.

Возвращаемые коды

Возвращает информацию **Find Object** такую как почему поиск удался или нет. Это числовой результирующий код. Таблица ниже содержит описание возвращаемых в результате поиска кодов.

Численное значение	Смысл
Ones Place	
xx1	Точное соответствие.
xx2	Используется подстановка из файла аббревиатур.
xx3 (-)	Точное соответствие не найдено.
xx4 (-)	Не указано имя объекта; соответствие не найдено.

Tens Place	
x1x	Сторона улицы не определена.
x2x (+ / -)	Номер дома был в диапазоне min/max.
x3x (+ / -)	Номер дома за пределами диапазона min/max.
x4x (+ / -)	Номер дома не указан.
x5x (-)	Улицы не пересекаются.
x6x (-)	Совпавшая строка не имеет объекта карты.
** Hundreds Place **	
1xx (+ / -)	Найденное имя только в одном регионе, отличающимся от заданного.
2xx (-)	Найденное имя в нескольких регионе, отличающихся от заданного.
3xx (+ / -)	Указан неуточненный регион, одно совпадение найдено.
4xx (-)	Не указан регион, найдено много совпадений.
5xx (+)	Найдено несколько совпадений в указанном регионе.

Как только код результата определен, Вы можете использовать результаты для выбранных случаев как показано в примере ниже. Пользователь вводит из текстового окна штат и город. Ввод попадает в метод **Find.Search**. Как только город найден в таблице городов, к карте добавляется аннотация и карта центрируется таким образом чтобы найденный объект был в центре карты.

```
'The code for a find command button
Private Sub cmdFind_Click()
    'Create a MapXLib for the feature found
    Dim FindCityObject As MapXLib.FindFeature
    'Sets the find feature object to the city the
    user types in
    Set FindCityObject = Map1.Layers("US Top 20 _
```



```

        Cities").Find.Search(txtCityName.Text,
txtStateName.Text)
    'Use result codes to determine if a city is
found
    If (FindCityObject.FindRC Mod 10 = 1) Then
        'Recenter the map to the city location
        Map1.ZoomTo 200, FindCityObject.CenterX, _
        FindCityObject.CenterY
    Else
        'Print out message box indicating city was
not found
        MsgBox ("City not found")
    End If
End Sub

```

Инструменты

- ♦ Стандартные инструменты
- ♦ Создание собственных инструментов
- ♦ Создание инструмента рисования полигона

Большинство картографических приложений представляет инструментарий помогающий решать наиболее общие задачи редактирования карт (такие как рисование линий на карте) и задачи навигации (такие как масштабирование). MapX представляет ряд таких наиболее используемых картографических инструментов, а также возможность создать свои собственные инструменты.

Стандартные инструменты

Используя MapX, Вы можете легко встраивать общие кнопки инструментов в свое приложение. MapX имеет встроенную поддержку некоторых наиболее используемых картографических инструментов, включая:

- Инструменты навигации (**Zoom-In, Zoom-Out, Pan, Center**) позволяющие пользователю менять масштаб и позиционирование карты.
- Инструмент **Подписывания** позволяющий пользователю щелкать на объекте карты и подписывать его.
- Несколько инструментов выбора, дающего разные пути выборов на карте.
- Инструменты создания объекта, которые позволяют создавать новые объекты карты.

Инструменты выбора обеспечивают встроенную поддержку клавиш (**SHIFT, CTRL**): Нажмите клавишу **SHIFT** пока используете инструмент выбора и произойдет отмена выбора объектов; нажмите **CTRL** пока используете инструмент выбора и инструмент добавит объект к выборке. MapX автоматически отображает различные курсоры, в зависимости от того, какая клавиша была нажата (плюс или минус), так что пользователь может понять эффект действия клавиши.

Пользовательские инструменты

Если Вам надо сделать кнопку инструмента, которого нет в MapX, Вы можете просто создать пользовательский инструмент с помощью метода **Map.CreateCustomTool**.

Когда Вы создаете пользовательский инструмент, Вы контролируете то, какой 'тип' инструмента Вы создаете – другими словами, Вы выбираете позволяет ли инструмент реагировать на щелчек мышки, или на щелчек и протягивание курсора при рисовке линии, щелкать и вести курсор для рисовки прямоугольника и др.. Вы так же выбираете какой курсор появится когда используется инструмент.

Определение активного инструмента

Что бы установить, какой инструмент будет использоваться, установите свойство **CurrentTool** объекта **Map**.

Для активации одного из стандартных инструментов, установите свойство на одну из констант **ToolConstants**. Например, для изменения инструмента в инструменте **Zoom In**:

```
Map1.CurrentTool = miZoomInTool
```

Для активации пользовательского инструмента, используете величину **ToolNumber** которую Вами определяется при использовании метода **CreateCustomTool**.

```
Map1.CurrentTool = 99
```

Инструменты редактирования объектов

Инструменты редактирования объекта позволяет пользователю создавать и изменять объекты в слое Map. Есть четыре стандартных инструмента создания объекта: **Add Point**, **Add Line**, **Add Polyline** и **Add Region**. Эти инструменты добавляют новый объект к тому слою, который определен в свойстве **Map.InsertionLayer**. Может быть только один слой для вставки объекта и по умолчанию он не задан. Установите текущий инструмент на инструмент создания объекта когда слой для вставки не указан и получите сообщение об ошибке.

MapX также поддерживает изменение существующих объектов карты. Для редакции объектов, свойство **Layer.Editable** должно быть установлено на true для любого слоя, которые Вы хотите изменить. Затем, встроенный инструмент **Arrow** может использоваться для удаления или изменения объектов в текущей выборке. Для перемещения выбранных объектов, просто щелкните на них и перемещайте мышку. Для изменения размера выбранных объектов, щелкните на них и перемещайте маркеры. Для удаления выбранных объектов, нажмите клавишу **Delete**.

Имеющиеся стандартные инструменты

Различные инструменты позволяют мышке решать разные задачи. Например, если текущий инструмент установлен на **miLabelTool**, то когда Вы щелкаете мышкой, будет помещена подпись на этот объект карты. Курсор мышки будет меняться в зависимости от типа инструмента.

Есть стандартные инструменты в MapX:

Инструмент	Константа	Описание
Add Line	miAddLineTool	Добавляет линейный объект к слою.
Add Point	miAddPointTool	Добавляет точечный объект к слою.
Add Polyline	miAddPolyLineTool	Добавляет объект полилинию к слою.
Add Region	miAddRegionTool	Добавляет объект регион к слою.
Arrow	miArrowTool	Щелкните на заголовке или аннотации. Так же можно перемещать или менять размер выделенных объектов на редактируемом слое.
Center	miCenterTool	Изменяет центр карты.
Label	miLabelTool	Щелкните на объекте что бы подписать объект.
Pan	miPanTool	Зацепите, что бы переместить карту.
Polygon Select	miPolygonSelectTool	Щелкните, что бы нарисовать полигон;

		объекты внутри полигона выделятся.
Radius Select	miRadiusSelectTool	Выбирает все объекты в круге заданного радиуса.
Rect Select	miRectSelectTool	Выбирает все объекты внутри прямоугольника.
Select Tool	miSelectTool	Щелкните чтобы выделить объект.
Symbol	miSymbolTool	Размещает аннотацию символа.
Text	miTextTool	Размещает аннотацию текста.
Zoom In	miZoomInTool	Увеличивающая лупа.
Zoom Out	miZoomOutTool	Уменьшающая лупа.

Создание собственных инструментов

Когда Вы создаете пользовательский инструмент для любого приложения, то это происходит за три шага:

1. Создать инструмент.
2. Написать обработчик инструмента (код который будет реально выполнять функцию инструмента).
3. Использовать инструмент (передать его в руки пользователя).

Создание инструмента

Для создания инструмента вызывайте метод `CreateCustomTool`.

В этом примере создается собственный инструмент **Линейка**.

Назначение инструмента **Линейка** в определении расстояния между двумя точками на карте. Во-первых, мы объявляем константу `RULERTOOLID` равной 500, она будет представлять наш инструмент. Затем, когда мы загрузим главную форму приложения, мы создадим инструмент.

```
Const RULERTOOLID = 500 'This goes in the General  
declarations.
```

```
Private Sub Form_Load()  
Map1.CreateCustomTool RULERTOOLID, miToolTypeLine,  
miSizeCursor  
End Sub
```

В этом вызове **CreateCustomTool**, мы указали три требуемых параметра: **ToolNumber**, **Type** и **Cursor**. **ToolNumber** это константа **RULERTOOLID**, параметр, который мы создали для представления инструмента. Параметр **Type** это **ToolTypeConstants**, определяющий поведение инструмента. В этом случае, это **miToolTypeLine** позволяющий пользователю щелкать мышкой и вести ее создавая линию. Параметр **Cursor** это **miSizeCursor**, который означает, что когда инструмент выбран, то он появится имеет указанный размер.

CreateCustomTool имеет два дополнительных параметра, которые так же используют значение **CursorConstants**. **CursorConstants** определяет курсоры так, Вы можете их с пользовательскими инструментами. Они определяют вид курсора, созданный Вами инструмент это **CurrentTool**.

Внимание: Смотрите *Справочник MapX* или *Электронную справку*, там есть полный список **CursorConstants**.

ShiftCursor определяет форму курсора инструмента, когда нажата клавиша **<SHIFT>**. **CtrlCursor** определяет форму курсора, когда нажата клавиша **<CTRL>**. Это полезно, если Вы хотите связать поведение курсора с этими клавишами.

Что бы сделать созданный инструмент активным, установите свойство **CurrentTool**. Например, можно поместить кнопку в форму Visual Basic и, когда пользователь нажимает ее, Вы устанавливаете свойство **CurrentTool**.

Возможные типы пользовательских инструментов

Константы **ToolTypeConstants** описывают тип инструмента, который Вы можете использовать для создания собственного инструмента. По сути дела они описывают поведение инструмента (т.е., **miToolTypeLine** позволяет пользователю рисовать линию; **miToolTypeCircle** позволяет пользователю рисовать окружность; и т.п.).

Константа	Поведение
miToolTypePoint	Определяет указанную точку.
miToolTypeLine	Рисует линию.
miToolTypeCircle	Рисует окружность.
miToolTypeMarquee	Рисует маркеры, окружающие выделенный объект.
miToolTypePoly	Рисует полилинию.
miToolTypePolygon	Рисует полигон.

После того, когда пользовательский инструмент **Ruler** создан, мы можем получить программный код для него.

Написание обработчика инструмента

Существует два варианта, когда код инструмента может выполняться: во время использования инструмента и после того как инструмент был использован. Для нашего инструмента **Линейка** мы будем выполнять код во время использования инструмента, потому что нам необходимо видеть результаты работы во время использования инструмента.

Путь, которым пользовательский инструмент **Линейка** действует: когда пользователь щелкает мышкой на карте, то отмечается стартовая точка действия инструмента. Пользователь ведет курсор по карте, не отпуская кнопки, затем останавливает мышку в конечной точке, отпуская клавишу. При этом помечается конечная точка инструмента измерения расстояния. Инструмент **Линейка** вычисляет расстояние между двумя точками. Что бы сделать это, нам надо выполнить код для инструмента **Линейка**, когда кнопка мышки нажата, и когда пользователь двигает мышку по карте, и останавливает ее в конечной точке. Чтобы получить местоположение начальной точки, в которой пользователь щелкнул мышкой, помещаем код в событие **MouseDown**:

```
Dim XDown As Double
Dim YDown As Double
```

```

Private Sub Map1_MouseDown(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)

    If Map1.CurrentTool = RULERTOOLID And Button =
vbLeftButton _
    Then
        Map1.ConvertCoord X, Y, XDown, YDown,
miScreenToMap
    End If
End Sub

```

Когда пользователь нажимает кнопку мыши происходит событие **MouseDown**. Код в событии **MouseDown** получает координаты X,Y точке в которой пользователь нажал кнопку, потом конвертирует их из экранных координат в координаты карты, и сохраняет в глобальных переменных **Xdown** и **Ydown**. **Xdown** и **Ydown** представляют стартовую точку для измерения расстояния.

Чтобы получить конечную точку, когда пользователь проведет над картой курсором мышки и поставит следующую точку, мы помещаем код в событие **MouseMove**:

```

Private Sub Map1_MouseMove(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)

    If Map1.CurrentTool = RULERTOOLID And Button =
vbLeftButton _
    Then
        Dim MapCoordX As Double, MapCoordY As Double
        Map1.MapUnit = miUnitMile
        Map1.ConvertCoord X, Y, MapCoordX, _
            MapCoordY, miScreenToMap
    End If
End Sub

```

```

        fMainForm.sbStatusBar.SimpleText =
Map1.Distance(XDown, _
                YDown, MapCoordX, MapCoordY) & " miles"
    End If
End Sub

```

Этот код будет выполняться непрерывно до тех пор, пока пользователь не остановит движения мыши и не отпустит кнопку. В последний момент этот код выполняется до конечной точки перед тем как мышка перестанет двигаться, и производится вычисление расстояния. Затем мы конвертируем экранные координаты точки в координаты карты и используя метод **Distance** объекта **Map** рассчитаем расстояние между двумя точками. Результат отобразится в строке сообщений формы.

Событие ToolUsed

Во многих случаях Вам понадобится выполнить какой-либо код после того как был использован инструмент. Например, после того как пользователь нарисовал окружность и отпустил кнопку, подсчитать, сколько клиентов попало в заданный радиус. Если инструменту необходима такая функциональность, используйте событие **ToolUsed** объекта **Map** для помещения Вашего кода.

Этот пример демонстрирует:

- Использование метода **Map.CreateCustomTool** для создания собственного инструмента "линия" (инструмент будет позволять пользователю "щелкнув и потянув" создать линию).
- Использование события **ToolUsed** для осуществления некоторых действий после того как пользователь использовал инструмент.

```

' Следующий код в событии Form_Load формы содержащей
карту (Map1) 'создает инструмент "линия".
Private Sub Form_Load()
    Map1.CreateCustomTool 1, miToolTypeLine,
miIconCursor

```

```
End Sub
```

```
' The following code (ToolUsed event) is called  
whenever any
```

```
' tool is used on the map.
```

```
' If the Current Tool is #1 (created above) then the  
code
```

```
' within the If-Then statement will be true, and we  
will set the caption of a label "lblDistance" equal  
to the Distance passed in with the "line-type" tool.
```

```
Private Sub Map1_ToolUsed(ByVal ToolNum As Integer,  
ByVal _
```

```
                    X1 As Double, ByVal Y1 As Double, ByVal
```

```
—
```

```
                    X2 As Double, ByVal Y2 As Double, ByVal
```

```
—
```

```
                    Distance As Double, ByVal Shift As _
```

```
                    Boolean, ByVal Ctrl As Boolean, _
```

```
                    EnableDefault As Boolean)
```

```
    If ToolNum = 1 then
```

```
        lblDistance.Caption = Str$(Distance)
```

```
    End If
```

```
End Sub
```

Дайте инструмент в руки пользователя

Чтобы "дать инструмент в руки пользователя", установите свойство

CurrentTool объекта **Map**:

```
Map1.CurrentTool = miZoomInTool
```

-или-

```
Map1.CurrentTool = 99
```

Создание инструмента рисования полигона (Polytools)

“Полиинструмент” это такой инструмент, который позволяет пользователю щелкать мышкой неоднократно -- например, рисовать полигон или полилинию.

Стандартный инструмент Polytools

MapX имеет стандартный инструмент выбора в полигоне. Этот инструмент позволяет пользователю нарисовать полигон, а затем выбрать все объекты центроид которых попадает в это полигон. Поиск идет только в выбираемых (selectable) слоях. Сшитые, растровые и косметические (Userdraw) слои игнорируются.

Чтобы активизировать этот инструмент установите свойство **CurrentTool** равным **miPolygonSelectTool** (value: 1010).

Когда этот инструмент используется, действует событие **PolyToolUsed**. Если пользователь завершил создание полигона (**miPolyToolEnd**) двойным щелчком, пробелом, или возвратом каретки, осуществляется поиск, и **miPolyToolEnd** (1) передается в событие **PolyToolUsed**. Если пользователь завершает рисование полигона нажатием клавиши **ESC**, поиск отменяется; **miPolyToolEndEscaped** (2) передается в событие **PolyToolUsed**. Пользователь может удалить все узлы линии (клавиша **backspace Delete**). Когда удален последний узел, передается флаг (2).

Собственный инструмент Polytools

Чтобы создать собственный инструмент **polytool**, вызывайте метод **CreateCustomTool**, и указывайте значение **ToolTypeConstants** of **miToolTypePoly**.

Чтобы сделать Ваш инструмент polytool активным инструментом, установите свойство **CurrentTool**.

Всегда, когда пользователь выбирает пользовательский инструмент и использует его на карте, MapX вызывает событие **PolyToolUsed**.

Следовательно, надо добавить код процедуры события **PolyToolUsed**, что бы инструмент подействовал.

Пример использования собственного инструмента Polytools

После того как Вы использовали метод создания инструмента polytool **CreateCustomTool**, следующий пример показывает как обрабатывать событие **PolyToolUsed**. Этот пример позволяет пользователю рисовать объект полигон или линию на слое Layer 1. Этот пример использует объекты feature. Также используется объект **Parts**, который описан в файле справки и *Справочнике*.

```
Private Sub map1_PolyToolUsed(ByVal ToolNum As Integer, _  
                                ByVal Flags As Long, ByVal pts As  
Object, _  
                                ByVal Shift As Boolean, ByVal Ctrl As  
_  
                                Boolean, EnableDef As Boolean)  
    If Flags = miPolyToolBegin Then  
        'Someone's beginning the use of a PolyTool...  
    ElseIf Flags = miPolyToolEnd Then  
        ' The user finished using a PolyTool by double  
clicking  
        If ToolNum = MY_SUPER_POLYGON_TOOL Then  
            ' They used MY_POLYGON_TOOL! Make a new  
            ' region feature and add it to the first  
layer  
            dim f As New Feature
```

```

        set f = map1.FeatureFactory.CreateRegion
(pts, _
    map1.defaultstyle)
    map1.Layers(1).AddFeature f

    ElseIf ToolNum = MY_SUPER_POLYLINE_TOOL Then
        dim f As New Feature
        map1.Layers(1).AddFeature f
    End If

    ElseIf Flags = miPolyToolEndEscaped Then
        ' The user hit 'Esc' or backspaced all the
nodes
        ' away... don't add anything in.
    End If

End Sub

```

10

Размещение данных на карте

- ◆ Размещение данных на Карте
- ◆ Что такое связывание данных?
- ◆ Преимущества помещения данных на карту
- ◆ Как добавлять данные к карте
- ◆ Объект DataSet
- ◆ Коллекция DataSets
- ◆ Метод DataSets.Add
- ◆ Метод Fields.Add
- ◆ Отображение Ваших данных в виде слоя точек
- ◆ Превращение нового слоя с точками в постоянный слой
- ◆ Использование геословаря при связывании данных
- ◆ Различные типы источников данных

Наборы данных **Datasets** помогут Вам связать пользовательские данные с Вашей картой. Например, если У Вас есть база данных Microsoft access по продажам в графствах и есть база данных Lotus Notes с торговыми точками, то можно связать такие данные с картой и посмотреть корреляцию между двумя типами данных.

Что такое связывание данных?

Связывание данных это процесс передачи данных из источника данных в MapX.

Существует много различных типов данных; MapX позволяет Вам связывать несколько различных типов **DataSources**. В MapX, данные представлены как объект **DataSet**. Первый аргумент метода **Datasets.Add** позволяет Вам указывать значение **DataSetTypeConstants**, которое определяет тип связываемых данных, которые Вам надо связать. Следующие типы данных можно связывать:

ADO	Этот тип связывания данных использует MS Active data objects ADO recordset.
DAO	Объект daoRecordset. Вы можете получить его из управления данными Visual Basic, из формы Access или при создании его в Visual Basic, Access, or C++.
Delphi	Этот тип использует источник данных Borland BDE.
Global Handle	Этот тип связывания позволяет Вам передавать блок данных с разделителей в виде табулятора.
Layer	Этот тип связывания данных позволяет Вам создавать набор dataset, который использует поля из таблицы MapInfo.
Notes View/ NotesQuery	Эти типы связывания данных имеют дело с Lotus Notes.
ODBC	MapX может использовать ODBC для получения данных из любых источников данных ODBC.
OLE Data	Это для контейнеров, таких как PowerBuilder.
Oracle Express Objects	Это обеспечивает доступ к datacube как к набору данных dataset.
RDO	Использует MS Remote Data Objects и объект RDO

	Resultset.
SafeArray	Набор данных COM dataset обеспечивающий статическое связывание данных из safearray.
Unbound	Если Вы не можете поддержать какой-либо другой формат, MapX обеспечит обращение с ним через ‘заднюю дверь’. С таким типом связывания данных, Вы можете установить цикл событий, в котором MapX будет опрашивать контейнер на предмет значений данных, по одной ячейки за раз.

Такие пользовательские, созданные пользователем типы связанных данных будут подробнее обсуждены далее в этой главе.

Преимущества помещения данных на карту

Существуют два главных способа связывания Ваших данных с картой:

- Вы можете просмотреть данные как объекты на карте. Если у Вас есть таблица Microsoft Access с местоположениями офисов по продажам для United States. Хотя связыванием данных Вы можете добавить Вашу таблицу как **Dataset** с методом **Datasets.Add** используя объект **BindLayer** как один из параметров. Будет создан новый метод на карте, показывающий точки на карте США. После того Вы добавили данные к карте, Вы можете использовать MapX что бы легко создавать приложения, ищущие ближайшее место продажи к заданному адресу покупателя. Поиск объектов на карте обсуждается в главе *"Поиск объектов на Карте"*.
- Вы можете связать Ваши атрибутивные данные с картой и затем использовать их для создания тематической карты. В данном случае таблица Microsoft Access мест продаж так же содержит колонку общих сумм продаж для каждого места. Используйте метод **Datasets.Add** и Вы сможете добавить колонку, содержащую суммы продаж к карте США. После добавления данных к карте, Вы можете создать тематическую карту сумм продаж по штатам. Тематическая картография обсуждается в главе *"Тематическая картография и анализ"*.

Как добавлять данные к карте

Связывание данных может быть осуществлено двумя способами:

- Если у Вас есть Visual Basic для управления связыванием данных, в процессе отладки Вы можете использовать свойство **DataSet** объекта **Map**.

-или-

- Свяжите данные на программном уровне, используя метод **DataSets.Add**. С этим методом Вы можете сообщить MapX какой источник данных будет использоваться, некоторую информацию об этом и какой слой будет связываться. С другой стороны, метод **Add** позволяет Вам связывать данные без явного указания природы данных; в этом случае, MapX анализирует Ваши данные и автоматически определяет наилучший путь связывания этих данных с картой.

Результат процесса связывания заключается в создании объекта **DataSet**. Этот объект **DataSet**, который добавляется к коллекции **DataSets**, содержит вычисленные значения для объектов на слое карты, с которым связываются данные. Например, если данные связаны с картой США, каждый штат будет иметь новые значения данных, которые будут использоваться для контроля того, как прорисовываются штаты. Если источник данных имеет множественные записи для каждого штата, значения могут быть суммированы, осреднены или подсчитаны. Объект **DataSet** имеет свойство **Value**, которое Вы можете использовать для доступа к вычисленным значениям для каждой строки (объекта) карты.

Для большинства типов связывания данных, `DataSource` (второй параметр для **`DataSets.Add`**) в действительности является OLE interface. MapX использует интерфейс для прямого доступа к данным из источника данных. В действительности они не передаются в **`DataSets.Add`**.

Объект **`DataSet`**

Объект **`DataSet`** это результат связывания данных из источника данных в MapX.

Объект **`DataSet`** содержит вычисленные значения для объектов на слое карты, с которым связываются данные. Источники данных могут быть DAO Recordset, ODBC data source, Global Handle (это путь передачи в блок с данными разделенными табулятором), OLE Data (для контейнеров, таких как PowerBuilder), и несвязанные Unbound Data (где MapX запрашивает контейнер о значениях данных, по ячейке за один раз).

Коллекция **DataSets**

Коллекция **DataSets** это объект, представляющий все объекты **DataSets** для карты. Коллекция **DataSets** имеет методы и свойства, используемые для добавления, удаления или работы с существующими объектами **DataSet** в коллекции.

Добавление объекта **DataSet** или удаление объекта **DataSet** из указанной коллекции сопровождается методами, перечисленными ниже:

Метод	Описание	Пример кода
Add	Создает указанный набор dataset и добавляет его к коллекции.	Set ds = Map1.Datasets.Add _ (miDataSetDAO, rs)
Remove	Удаляет указанный объект DataSet из коллекции Datasets.	Map1.Datasets.Remove 2

Метод DataSets.Add

С помощью метода **DataSets.Add**, Вы можете связать данные из источника данных с MapX. Это привязывает данные из внешнего источника к карте. Метод **DataSets.Add** позволяет Вам установить определенный объект **DataSet** и добавить его к коллекции объектов **DataSets**. Обратите внимание на синтаксис метода **Datasets.Add**.

Внимание: Дополнительные параметры в [квадратных скобках].

DataSets.Add Type, SourceData, [Name], [Geofield],
[SecondaryGeofield], [BindLayer], [Fields], [Dynamic]

Параметр Type

Этот параметр является тем типом объекта **dataset**, который добавляется. Этот параметр берет значение константы **DataSetTypeConstants**. Ниже следует список констант типов **DataSet**:

Константы типов DataSet

Тип	Описание
miDataSetADO	ADO
miDataSetDAO	Data Access Object
miDataSetDelphi	Delphi Native
miDataSetDelphi4	Delphi4
miDataSetGlobalHandle	Tab delimited data
miDataSetLayer	MapInfo Table
miDataSetNotesQuery	Lotus Notes Query
miDataSetNotesView	Lotus Notes View
miDataSetODBC	ODBC Database
miDataSetOEO	Oracle express Objects
miDataSetOLEData	OLE datasource
miDataSetRDO	RDO

miDataSetSafeArray	Safe Array
miDataSetUnbound	MapX запрашивает данные из контейнера.

Ниже более детально описаны отдельные типы данных.

Параметр SourceData

Этот параметр является ссылкой на данные и он зависит от типа объекта **Dataset**. Для большинства типов связываемых данных, **DataSource** (второй параметр для **DataSets.Add**) в действительности является OLE interface. MapX использует интерфейс для прямого доступа к данным из источника. В действительности данные не передаются в **DataSets.Add**.

Здесь показаны допустимые источники данных для каждого типа наборов данных:

Тип Dataset	Допустимый источник данных
miDataSetADO	таблица ADO
miDataSetDAO	объект DAORecordset.
miDataSetDelphi	Delphi Native
miDataSetDelphi4	Delphi4 recordset
miDataSetGlobalHandle	Вариант типа VT_I4 и lVal эквивалентного глобальному обработчику данных (global memory handle).
miDataSetLayer	Таблица MapInfo.
miDataSetNotesQuery	Lotus Notes Query
miDataSetNotesView	Lotus Notes View
miDataSetODBC	Объект ODBCQueryInfo.
miDataSetOEO	Oracle express Objects
miDataSetOLEData	Игнорируется методом Datasets.Add.
miDataSetRDO	Таблица RDO

miDataSetSafeArray	Safe Array
miDataSetUnbound	Пусто (вместо этого, используйте событие RequestData для доступа к данным, чей формат известен только программисту).

Параметр Name

Этот параметр является строкой, которая единственным образом идентифицирует набор данных **dataset**. Это дополнительный параметр, если он не указан, имя используется в форме **DataSetN** где N это порядковый номер в коллекции **Datasets**.

Параметр Geofield

Этот параметр является именем или индексом колонки в источнике данных, которая содержит географическую информацию. Если этот параметр не указан, MapX ищет все по всем полям, что бы определить какая колонка в источнике данных содержит географическую информацию, как указано в GeoDictionary. Это не будет обязательно поле с географическими данными, это может быть колонка с уникальными ключами, такими как почтовые коды. Таким образом, если Вы знаете какая колонка в источнике данных содержит географическую информацию, то Вам надо это указать.

Если Вы собираетесь просматривать данные как объекты карты, колонка **GeoField** в источнике данных должна быть уникальной. Колонка **GeoField** будет использоваться для именования объектов на новом слое точек. Не уникальные значения будут словаться в одну точку путем добавления к новому слою точек в тех местах где встречены двойные точки, значения в таких местах будут объединяться.

Если коллекция **Fields** определена, параметр **Geofield** указывает на колонку в коллекции **Fields** раньше чем на такую же в источнике данных. Смотрите описание параметра **Fields** ниже.

Второй параметр **Geofield**

Этот параметр требуется только когда слой, который связан с набором данных **dataset** имеет неуникальную ключевую колонку. Например, использование таблицы MapInfo “United States Counties” для связывания данных требует второго параметра **geofield**, поскольку имена графств не являются уникальными для одного штата.

Несколько штатов в таблице имеют имена графств “Warren” или “Washington”, среди прочих. Таким образом, необходима дополнительная информация для уточнения в процессе связывания данных. Вы хотите связать данные с Warren, штат New York или с Warren, штат New Jersey? Указав колонку “County” в качестве **Geofield** и колонку “State” в качестве **Secondary Geofield**, MapX сможет разобраться какие данные будут связываться с Warren, штат New York и какие данные будут связываться с Warren, штат New Jersey. Произойдет уточненное связывание данных с графствами в каждом штате. (Этот подход идентичен процессу “уточнения границ” (**Refining Boundary**) при создании объекта **Find**; смотрите раздел “Поиск объектов на карте”.)

Если коллекция **Fields** определена, параметр **SecondaryGeofield** указывает в первую очередь на колонку в коллекции **Fields**, а не на колонку в источнике данных. Смотрите ниже описание параметра **Fields**.

Параметр **BindLayer**

Этот параметр указывает к какому слою карты надо прикрепить Ваши данные, если Вы связываете атрибутивные данные, или указывает на **BindLayerObject** если входные данные имеют географические ссылки на файл ссылок (такой как файл почтовых индексов Zip Codes) или содержит значения Долгота/Широта. Это дополнительный параметр, и если он не указан, MapX ищет слои в GeoDictionary что бы прикрепить к ним данные.

Если Вы знаете какой слой карты будет связан с данными, Вы должны указать его. Когда осуществляется подбор **BindLayer**, **Geofields** должен быть уникальным. Только первый объект из неуникальных данных пройдет процедуру подбора. Остальные будут проигнорированы. Более детально **BindLayer** обсуждается далее в этой главе.

Параметр **Fields**

Этот параметр является объектом **Fields**, представляющим собой коллекцию объектов **Field**. Объекты **Field** используются для описания того, какие поля берутся из источника данных, и какая функция объединения данных используется в том случае, если более чем одна запись в источнике данных присваивается одному объекту на карте. Таким образом будет установлена коллекция полей (колонок), которая будет использоваться для связывания с картой. Это дополнительный параметр, и если он не указан, то связываются все колонки, и значения данных суммируются, если более чем одна запись соответствует одному объекту. Коллекция **Fields** обсуждается более детально позднее.

Если коллекция **Fields** определена, то параметры **Geofield** и **SecondaryGeofield** ссылаются в первую очередь на колонки в коллекции а не в источнике данных.

Параметр Dynamic

Этот параметр является логической величиной, которая контролирует является ли динамической связь данных. Это дополнительный параметр; если он пропущен, по умолчанию будет значение False, означающее, что связь является статической (то есть, MapX будет копировать необходимые данные, когда база данных открыта). Если Вы укажете True, MapX получит “живой” доступ к данным, только тогда, когда требуются данные (например, при подписывании). Если Вы указываете True но набор данных dataset не поддерживает динамические колонки, появится сообщение об исключительной ситуации.

Простой пример DataSets.Add

Следующий код открывает таблицу US_Cust из базы данных MapStats MSAccess, которая включена в папку DATA Вашей директории MapX 4.0. Предполагается, что Map1 это карта США.

```
Dim ds As Dataset
Dim db As Database
Dim rs As Recordset

'Open the MapStats.MDB Databasse
Set db = DBEngine.Workspaces(0).OpenDatabase( _
    "C:\Program Files\MapInfo MapX
4.0\Data\Mapstats.mdb")
'rs это SourceData recordset который является
таблицей MS Access
Set rs = db.OpenRecordset("US_Cust")
'Add the dataset to the map
Set ds = Map1.Datasets.Add miDataSetDAO, rs
```

Использование коллекции Fields

Источник данных может иметь много колонок. MapX получает издержки от каждой связываемой колонки данных, поэтому Вам связывать только те колонки, которые необходимы на карте (такие данные, которые Вы хотите использовать в тематических построениях или для подписывания). Используйте последний параметр **DataSets.Add (Fields)** для настройки коллекции **Fields** полей (колонок) которые Вам надо связать с картой.

Внимание: Когда декларируется переменная полей, то объявите это как "MapXLib.Fields". Это необходимо для предотвращения конфликтов с объектом DAO "Fields".

`Dim flds As New MapXLib.Fields`

Вы можете получить доступ к коллекции наборов данных **Fields** посредством свойства **Dataset.Fields**.

Метод	Описание	Пример кода
Add	Добавляет поле к коллекции Fields.	Flds.Add "Sales", "Sum_of_Sales" _ , miAggregationSum
Remove	Удаляет поле из коллекции Fields.	Flds.Remove 3
RemoveAll	Удаляет все объекты Field из коллекции.	Flds.RemoveAll

Метод **Fields.Add**

Метод **Fields.Add** позволяет Вам добавлять колонку данных из источника данных в качестве поля в коллекцию **Fields**. Коллекция **Fields** создается для использования с методом **DataSets.Add** (или с методом **Themes.Add**; смотрите раздел Тематическая картография и анализ). Параметр **Fields** метода **Datasets.Add** берет коллекцию **Fields**, и строится с использованием этого метода **Add**. Метод **Add** не может быть использован коллекцией **Fields** после того, как набор данных **dataset** создан.

Синтаксис **Fields.Add** показан ниже. (**Внимание:** дополнительные параметры в [квадратных скобках]).

Fields.Add DataSourceCol, Name, [AggregateFunction], [Type]

Части	Описание
DataSourceCol	Имя или индекс колонки источника данных.
Name	Имя добавляемого поля.
AggregateFunction	Используемая функция объединения. Берется значение AggregationFunctionConstants
Type	Тип данных в колонке. Берется значение FieldTypeConstants . Параметр type используется только с Unbound DataSets (несвязанным набором данных). Он игнорируется для всех других типов DataSet .

Объединение данных

Параметр **Aggregate Function** метода **Fields.Add** определяет как **MapX** вычисляет значение **Field** в случае когда на один объект приходится несколько записей.

Например, если данные выглядят так:

Штат	Число продаж
CA	120
NY	100
CA	50
CA	110

Здесь есть три значения продаж в Калифорнии, и после прикрепления значений данных к Калифорнии, возникнет необходимость сообщить MapX каким методом будет осуществляться объединение, или суммированием, или осреднением или подсчетом числа продаж.

Параметр **Aggregate** метода **Fields.Add** по умолчанию равен **miAggregationIndividual** для колонок со строковыми значениями и равен **miAggregationSum** для числовых значений.

Для объединения используются следующие константы:

Константы функции объединения

Константа	Описание
MiAggregationSum	Суммирует данные.
MiAggregationAverage	Берет среднее значение.
MiAggregationIndividual	Выбирает в каждой отдельной записи.
MiAggregationCount	Подсчитывает число записей.

Пример

Следующий пример показывает как создать коллекцию полей из источника данных **DAO recordset** и затем добавить **dataset** используя коллекцию **Fields**.

```
Private Sub cmdOK_Click()  
    Dim ds As MapXLib.Dataset  
    Dim rs As Recordset
```

```

Dim i as integer
Dim flds As New MapXLib.Fields

'Открытие набора записей recordset выбранного из
списка
    Set rs = db.OpenRecordset(DatasetsList.Text)

'Использование константы AggregationFunction
выбранной из списка
    Dim AggregateTypeNum As Integer
    Select Case AggregateType.ListIndex
    Case 0
        AggregateTypeNum = miAggregationSum
    Case 1
        AggregateTypeNum = miAggregationAverage
    Case 2
        AggregateTypeNum = miAggregationCount
    Case 3
        AggregateTypeNum = miAggregationIndividual
    Case 4
        AggregateTypeNum = miAggregationAuto

    End Select

'Цикл по полям набора записей DAO Recordset
' и добавление их к коллекции MapX Fields.
    For i = 0 To rs.Fields.Count - 1
        flds.Add rs.Fields(i).Name,
rs.Fields(i).Name, _
        AggregateTypeNum
    Next

```



```

        'Добавление набора данных, использование
коллекции полей
        Set ds = Map1.Datasets.Add( _
            miDataSetDAO, rs, "US State Data", , , ,
flds)

        MsgBox "Dataset " & DatasetsList.Text & " Added"
        Unload Me
    End Sub

```

Отображение Ваших данных в виде слоя точек (BindLayer)

Связывание данных методом **BindLayer** это такой тип связывания, при котором источник данных используется для создания нового слоя точек, размещенных в соответствии с данными. Это происходит в двух случаях:

- Когда Ваши данные содержат координаты x и y (такие как широта и долгота).

Точки создаются в определенных местах. Смотрите *“Отображение данных с координатами X/Y в качестве слоя точек на карте,”* ниже.

- Когда данные содержат ссылочную информацию, например почтовые индексы (Zip Codes).

Точки создаются в приблизительных местах размещения этих почтовых индексов (их центроидов). Смотрите раздел *“Отображение данных с почтовыми индексами в виде слоя точек”*, ниже.

Объект **Bindlayer** требуется когда связываются данные, содержащие X/Y координаты или точечную информацию, такую как почтовые коды, и Вы хотите увидеть точки, созданные на новом или существующем слое.

Когда Вы добавляете набор данных **dataset** используя **BindLayerObject**, то точки автоматически отображаются на карте. Это не тот случай, когда добавляется **dataset** без **BindLayerObject**. Так ведет себя только **BindLayerObjects**.

Точки будут изображены в виде серых звезд. Если надо поменять стиль, то установите свойство **Layer.StyleOverride** на значение **True** и поменяйте свойства стиля. Изменения стиля не отображаются пока **StyleOverride** установлен на **True**.

Свойство объекта **BindLayer**

Свойство	Описание	Значение
LayerType	Указывает тип слоя, с которым будут связаны данные.	A BindLayerTypeConstant: MiBindLayerTypeNormal MiBindLayerTypeXY MiBindLayerTypePointRef
CoordSys	Указывает систему координат в которой будет создан слой.	Объект CoordSys
FileSpec	Позволяет Вам указать имя и место размещение файла, так что метод Datasets.Add может создавать постоянный слой вместо временного слоя.	Текст, определяющий путь и имя файла.
KeyLength	Положительное число, представляющее размер символьной колонки в результирующем слое.	Целочисленное значение (1-254).
LayerName	Определяет имя слоя, который связывается с данными, если LayerType имеет значение miBindLayerTypeNormal. Это имя только созданного слоя, если LayerType имеет значение miBindLayerTypeXY или miBindLayerTypePointRef.	Строковая величина.
RefColumn1	Поле содержащее долготу, когда LayerType имеет значение miBindLayerTypeXY, или поле, содержащее ссылочное значение	Строковое или целое значение, ссылающееся на колонку (начиная с 1).

	(такое как ZIP Code) если <code>LayerType</code> имеет значение <code>miBindLayerTypePointRef</code> .	
<code>RefColumn2</code>	Определяет поле, содержащее широту когда <code>LayerType</code> имеет значение <code>miBindLayerTypeXY</code> .	Строковое или целое значение, ссылающееся на колонку (начиная с 1).
<code>ReferenceLayer</code>	Указывает имя ссылочного файла, используемого если <code>BindLayer.LayerType</code> имеет значение <code>miBindLayerTypePointRef</code> .	Строковая величина.

Порядок операций для использования объекта **BindLayer**:

1. Создайте объект **BindLayer**.
2. Используйте этот объект когда при добавлении набора **dataset** путем передачи как параметр **BindLayer** в метод `datasets.Add`.

Как создать объект **BindLayer** и отобразить Ваши X / Y или данные ссылающиеся на точки будет показано ниже.

Отображение данных с X / Y в виде слоя точек

Если Ваши данные содержат координаты X / Y, Вы можете создать и показать слой точечных объектов посредством связывания данных.

Используйте объект **BindLayer** и установите :

- Его свойство `LayerType` равным `miBindLayerTypeXY`.
- Его свойство `RefColumn1` должно быть именем или индексом (начиная с 1) колонки, содержащей значение координаты X.
- Его свойство `RefColumn2` должно быть именем или индексом (начиная с 1) колонки, содержащей значение координаты Y.

Пример

Этот пример делает следующее:

- Открывает таблицу US_Cust Microsoft Access в Вашей базе данных Mapstats (пример).
- Создает объект **BindLayer** из таблицы US_Cust указывает тип связывания данных **BindLayer (miBindLayerTypeXY)** и индекс колонок в таблице US_Cust, содержащих координаты X и Y.
- Добавляет US_Cust как набор данных dataset, создает временный слой точек, и показывает каждого покупателя из таблицы US_Cust в виде точки на карте.

В этом примере предполагается, что Map1 это карта США.

```
Dim BindLayerObject As New MapXLib.BindLayer
Dim flds As New MapXLib.Fields

' Получение набора записей
' для картирования базы данных MapStats Access
' имеющей колонки с широтой и долготой
Set db = DBEngine.Workspaces(0).OpenDatabase( _
"C:\Program Files\MapInfo MapX
4.0\Data\Mapstats.mdb")

Set rs = db.OpenRecordset("US_Cust")
rs.MoveLast
Debug.Print "Record Set has " & rs.RecordCount &
" records."

' Заполнение в объекте Bindlayer
BindLayerObject.LayerName = "US Customers" 'Name
of new layer
BindLayerObject.RefColumn1 = 8 ' "Longitude"
' Использует номер колонки (начиная с 1) или имя
колонки
' BindLayerObject.RefColumn2 = 9 ' "Latitude"
BindLayerObject.LayerType = miBindLayerTypeXY
```

```

'Type for X/Y binding

Debug.Print "Finished setting up Bind Layer"

Set ds = Map1.Datasets.Add(miDataSetDAO, rs, "U.S.
Customers" _
, "City", "State", BindLayerObject)

Debug.Print "Finished adding dataset"

```

Отображение данных с почтовыми индексами в виде слоя точек

Если Вы имеете таблицу данных, не содержащую колонки с координатами X / Y, но с ссылочной информацией, такой как почтовые индексы (ZIP Codes), Вы можете показать данные в виде точек на карте. Применение ссылочной информации позволит Вам связать данные с отдельным слоем со ссылками, содержащим координаты X / Y для почтовых индексов (ZIP Code). Это делается указанием ссылочного слоя в объекте BindLayer когда устанавливается объект **BindLayer**, как показано ниже.

MapX применяет таблицу с именем Zipcodes.cpf в директории \MapInfo MapX 4.0\Maps которую Вы можете использовать как ссылочный слой для почтовых индексов США (ZIP Codes). Zipcodes.cpf так же имеет “дружественное имя” (описание, которое присваивается слою через геословарь) и которое звучит так: “US 5 Digit Zipcode Centers.”

Для создания и отображения слоя точечных объектов, построенных через связывания индексированных данных (ZIP Code), используйте объект **BindLayer** и настройте его так :

- Его свойство **LayerType** должно иметь значение **miBindLayerTypePointRef**.

- Его свойство **RefColumn1** будет именем или индексом(начиная с нуля) колонки в Вашей таблице, содержащей почтовый индекс (zip code) (или другие ссылки на точечные объекты).
- Его **ReferenceLayer** должно быть именем файла с ссылками. Файл с ссылками содержит координаты x и y каждого объекта, на который производится ссылка ZIP Code. Свойство **ReferenceLayer** может быть установлено на имя файла-слоя (полное имя файла) или быть установлено на “дружественное имя” слоя (описание, которое присваивается слою через геословарь).

Внимание: Ссылочный слой не нуждается в отображении на карте, но он должен быть установлен в геословаре.

Свойство **RefColumn2** используется в связывании данных почтовых индексов, ZIP Code, поскольку они уникальны (неповторяются).

Пример

Этот пример показывает как связывать почтовые индексы (ZIP Code). Предполагается, что уже используется таблица с почтовыми индексами (ZIP Codes) открытая как `Data2.Recordset`, и четвертая колонка в таблице пользователя содержит почтовые индексы ZIP Codes.

Этот пример делает следующее:

- Устанавливает объект **BindLayer**, используя "US 5 Digit Zipcode Centers" в качестве ссылочного слоя.
- Добавляет набор данных dataset.

Когда набор данных добавлен, MapX создает временный слой с точками для таблицы с адресами покупателей, где каждый покупатель по почтовому индексу получит временный слой точек для таблицы покупателей с X / Y координатами, соответствующими координатам их почтовых индексов (ZIP Code). В добавок, MapX будет добавлять колонку **Geofield** к новосозданному слою и затем свяжет с ним любые атрибутивные данные, указанные в наборе записей **Recordset**. Имя нового слоя получается из **BindLayerObject.LayerName** и будет следующим: "Customers By Zipcode centers."

```
Dim MyDataset as Dataset
```

```
' Set up the bind layer object
```

```
Dim BindLayerObject As New MapXLib.BindLayer
```

```
BindLayerObject.LayerName = "Customer's By Zipcode  
centers"
```

```
BindLayerObject.RefColumn1 = 3
```

```
' Третья колонка в источнике данных содержит  
информацию об zip кодах
```

```
BindLayerObject.LayerType = miBindLayerTypePointRef
```

```
BindLayerObject.ReferenceLayer = "US 5 Digit Zipcode  
Centers"
```

```
' Значения из колонки под именем "GEOABBR" будут
```

```
' использоваться как geofield для новой таблицы  
точек
```

```
Set MyDataset = Map.Datasets.Add(miDataSetDAO,  
Data2.Recordset, "Zipcode Dataset", "GEOABBR", ,  
BindLayerObject)
```

```
MyDataset.Themes.Add
```


Превращение нового слоя с точками в постоянный слой

Можно легко сделать из нового слоя точек, созданных с помощью объекта **BindLayer**, постоянную таблицу MapInfo. Используйте свойство **BindLayer.Filespec** или укажите имя и местоположение файла, так что метод **Datasets.Add** сможет создать постоянный слой из временного слоя. Если Вы не присваиваете это свойство, то слой будет временным. Это свойство применяется и к связыванию данных через колонки с координатами X / Y и к связыванию через ссылочные точки (zip code). Следующий пример создает постоянную таблицу через связывание данных с использованием координат X / Y.

```
Dim BindLayerObject As New MapXLib.BindLayer
Dim ds as Dataset

'Где будет сохранена постоянная таблица
BindLayerObject.FileSpec =
"C:\MapInfo\Data\Dealers.TAB"
BindLayerObject.LayerType = miBindLayerTypeXY
BindLayerObject.LayerName = "US Dealers"
'Имя используемого слоя
BindLayerObject.RefColumn1 = ("LONG")
'Имя колонки с долготой
BindLayerObject.RefColumn2 = ("LAT")
'Имя колонки с широтой
'Добавление набора данных и создание постоянного
слоя
Set ds = Map1.Datasets.Add(miDataSetDAO, rs, "My
Dealers", _ Dealer, BindLayer)
```

Использование геословаря при связывании данных

MapX использует файл GeoDictionary (названный по умолчанию **GEODICT.DCT**) что бы хранить информацию относящуюся к связыванию данных. MapX поставляется с двумя утилитами— **GeoDictionaryManager40.EXE** и **GeosetManager40.exe**—которые помогут Вам управлять геословарем GeoDictionary.

Когда Вы используете метод **DataSets.Add**, MapX может анализировать Ваши данные что бы определить как могут быть эти данные связаны с картой. Этот метод автоматически делает следующее:

- Определяет какая колонка в источнике данных содержит географическую информацию.
- Определяет какой слой карты будет связан с данными.

Обе эти возможности являются дополнительными. Есть параметры, через которые Вы можете точно указать какая колонка в источнике данных содержит географическую информацию, и какой слой будет связываться с данными. Если Вы знаете нужную колонку и слой, укажите их обязательно, это сильно облегчит работу.

Что бы связать данные со слоем карты в MapX, должны выполняться следующие условия:

1. Колонка карты с географическими ключами должна быть проиндексирована. Большинство карт поступают с уже индексированными ключевыми колонками, но бывают ситуации, когда надо самому выбрать, какие колонки являются ключевыми. Например, таблица United States имеет как минимум 3 ключевых колонки: имя штата (“New York”), аббревиатура штата (“NY”), и FIPS коды (36). Карта United States поставляемая MapInfo имеет первые две в качестве ключевых колонок, поэтому запрашиваемые данные из источника данных должны иметь или аббревиатуру штатов или их названия.

2. Карта и ее ключевые колонки должны быть указаны в GeoDictionary.

Некоторые ключевые колонки из слоя карты могут не содержать уникальные значения. Пример этого - имена графств в США: многие штаты имеют графства с названиями “Warren” или “Washington”. Таким образом, когда строка данных имеет имя “Washington,” MapX нуждается в более точной информации, например в каком это штате. Для примера с графствами США, с этой целью уточнения используется колонка State. Параметр **SecondaryGeoColumn** используется для указания колонки в источнике данных, которая содержит уточняющую информацию. GeoDictionary содержит информацию о том, какие карты содержат информацию о неуникальных ключах, и какие карты используются для их уточнения.

События MapX используемые при связывании данных

После того как колонка определена как географическая колонка из источника данных и определен слой карты, который будет связан с данными, начинается связывание. Каждая строка источника данных ‘сопоставляется’ с объектом и данные связываются. Если строка содержит географическую колонку, значение из которой не совпадает с объектом (например [“NA” вместо “MA”] или просто значение, которого нет на карте [“Puerto Rico”]), то событие **DataMismatch** запускается чтобы оповестить контейнер. Вы можете проигнорировать событие **DataMismatch**, в этом случае MapX просто проигнорирует такую строку таблицы.

Во время автоматического связывания данных, MapX вызывает событие **ResolveDataBind** если данные неоднозначны.

Во время связывания "несвязанных" данных, используйте событие **RequestData** для организации цикла, так что контейнер будет поставлять данные в MapX по одной ячейке за каждый проход цикла.

Обновление наборов данных **Datasets**

После того как данные из источника объединены и хранятся в MapX, и если эти данные в самом источнике изменяются, то MapX не воспримет автоматически и какие изменения. Метод **DataSet.Refresh** может быть использован для того что бы MapX заново перечитал данные из источника и переобъединил их и заново сохранил. На это однако тратится время. **DataSet.Refresh** не работает для **DataSets** который создает новые слои точек. Действие метода заключается в простом удалении слоя и новом добавлении набора данных dataset, или в осуществлении следующих шагов:

1. Создание нового слоя точек с использованием **Datasets.Add** с **BindLayer** имеющим тип **miBindLayerTypeXY**. Создание нового слоя только с колонками X и Y при передаче объекта **Fields**, с указанными только этими двумя колонками, в метод **Datasets.Add**.
2. Связывание остатка данных с этим вновь созданным слоем. Используется **Datasets.Add** с нормальным связыванием, используется поле X для параметра **Geofield** и поле Y в качестве параметра **SecondaryGeofield**. Второй набор данных dataset может быть обновлен.

Внимание: Этот метод не добавляет новых объектов к слою когда добавляются новые строки к источнику данных.

Различные типы источников данных

MapX позволяет связывать несколько различных типов **DataSources**. Первый аргумент метода **Datasets.Add** позволяет Вам указать значение **DatasetTypeConstants**, которое определяет тип связываемых данных, с которыми Вы работаете.

Каждый из типов источников данных описан ниже.

miDataSetDAO (Data Access Object Recordset)

Параметр **SourceData** метода **Datasets.Add** должен быть объектом **daoRecordset**. Вы можете получить его из **Visual Basic data control**, формы Access, или при создании его в Visual Basic, Access или C++.

Пример

```
'Этот пример демонстрирует простейшее использование
метода Datasets.Add

'. При таком его использовании Вы указываете только
источник данных и

'тип. MapX автоматически определяет какая колонка
представляет

'географическую информацию, и какой слой будет
связан с этой информацией
'

'Так же демонстрируется простейшее использование
метода Themes.Add

' При этом, когда все аргументы пропущены, MapX
'автоматически определяет имя для тематического слоя
, какая колонка
```

```

'используется для тематики и какой тип тематики
создается. В
'этом случае, MapX создаст слой диапазонов по первой
числовой колонке
'в таблице USA (TOTPOPHIS).
Private Sub Command1_Click()
    Dim db As Database
    Dim rs As Recordset
    Dim ds As Dataset

    'Сначала получаем таблицу, содержащую источник
данных
        Set db = DBEngine.Workspaces(0).OpenDatabase( _
            "C:\Program Files\MapInfo MapX
4.0\Data\Mapstats.mdb")
        Set rs = db.OpenRecordset("USA")

    'Эта строка создаст набор данных с именем "Dataset1"
        Set ds = Map1.Datasets.Add(miDataSetDAO, rs)

    'Создание тематического слоя по умолчанию из этого
набора данных
        ds.Themes.Add
End Sub

```

miDataSetODBC (Open DataBase Connectivity data source)

MapX может использовать ODBC для получения данных из любого источника данных ODBC. Вам надо указать строку соединения, имя источника данных, и строку SQL для выполнения с использованием объекта **ODBCQueryInfo**.

Свойство **ODBCQueryInfo.ConnectionString** устанавливает строку соединения с источником данных ODBC. Например, укажите “ODBC;”. Вы можете так же включить “uid=”, “pwd=” или DLG=. Если необходимая информация пропущена, будет подсказка о драйвере ODBC для **DataSource**.

“DLG=” контролирует отображение диалога с соединением: DLG=0 означает никогда не показывать этот диалог, DLG=1 означает всегда показывать его, DLG=2 показывает диалог только в тех случаях, когда он нужен (то есть если не вся необходимая информация введена).

Более подробно об объекте **ODBCQuery Info** смотрите в *Электронной Справке или в Справочнике*.

Пример

```
Private Sub Command2_Click()  
    Dim ds As Dataset  
    Dim parm As New ODBCQueryInfo  
    Dim fd As Field  
  
    parm.SqlQuery = "select * from usa"  
    parm.DataSource = "usa" ' name of odbc  
                           datasource  
    parm.ConnectionString = "ODBC;" ' can be  
    left blank, or can include uid= or pwd= or DLG=  
  
    Set ds = Map.Datasets.Add(miDatasetODBC, parm, "odbc  
Dataset")  
    For Each fd In ds.Fields  
        MsgBox fd.Name  
    Next  
    If Not (ds Is Nothing) Then ds.Themes.Add  
End Sub
```


miDataSetLayer (таблица MapInfo)

Этот тип связывания данных позволяет Вам создавать набор данных **dataset** который использует поля из таблицы MapInfo.

Пример

```
Dim FoundObj As FindFeature
Dim Layer As Layer
Dim ds As Dataset
Dim Findds As Dataset

' Добавление таблицы MapInfo в качестве слоя
Map1.Layers.Add "c:\program files\mapinfo _
    professional\data\usa\states.tab"

Set Layer = Map1.Layers("states")
Set Findds = Map1.Datasets.Add(miDataSetLayer,
    Layer)
Set Map1.Layers("states").Find.FindDataset = Findds
Set Map1.Layers("states").Find.FindField =
    Findds.Fields(2)
Set FoundObj=Map1.Layers("states").Find.Search("New
    York")

If (FoundObj.FindRC Mod 10 = 1) Then
    Map1.Zoom = 500
    Map1.CenterX = FoundObj.CenterX
    Map1.CenterY = FoundObj.CenterY
Else
    MsgBox "No exact match found. " +
        Str$(FoundObj.FindRC)
```

```
End If
```

miDataSetGlobalHandle (данные с разделителем tab)

Этот тип связывания данных позволяет Вам посылать их в блок данных с разделителем - табулятором. Параметр **SourceData** метода **Datasets.Add** должен быть переменной с типом VT_I4 и lVal равным обработке памяти глобальных данных. Формат каждой строки в блоке глобальной памяти:

```
Field TAB Field TAB Field CRLF
Field TAB Field TAB Field CRLF
. . .
```

где **Field** это строковая величина в кавычках или числовое значение без кавычек; TAB это символ 0x09; и CRLF это возврат каретки: 0x0D 0x0A.

Пример: Связывание данных с использованием C++

Ниже приведена один из обработчиков команд меню для добавления данных из файла использующие тип **miDataSetGlobalHandle**. Формат данных должен быть таким: кавычки вокруг строковых значений, табулятор между полями и перевод каретки в конце каждой записи. Например:

```
"\"NY\" \"t105.34\t100\t1\r\n"
"\"MA\" \"t245.19\t200\t2\r\n"
"\"NY\" \"t195.0\t300\t3\r\n"
"\"AK\" \"t195.0\t125\t4\r\n"
"\"CA\" \"t56.453\t200\t5\r\n";
```

Функция **CMapxSampleView::OnMapAdddata()** обрабатывает разделы меню из меню Карта. Она подсказывает имя файла, содержащего данные в форме, описанной выше, читает файл и добавляет его к коллекции карт набора данных.

```
void CMapxSampleView::OnMapAdddata()
{
```

```

:
:
    CFileDialog dlgFile(TRUE, "*.txt", NULL, 0,
szDataFilter, this);

    if (dlgFile.DoModal() == IDCANCEL)    //
Пользователь отменяет диалог
        return;

    // Читает файл в виде строковых величин и копирует
его в глобальную память
    // буфера
:
:
    // Размещение буфера памяти, копирования строковых
значений в него
:
:
    // Объявление переменных, которые будут
параметрами для
    // DataSets.Add()
    short Type;
    VARIANT SourceData, Name, GeoField,
SecondaryGeoField;
    VARIANT BindLayerName, Fields;
    CString strName= "TestData";

// установка дополнительных параметров; большинство
не будет использовано
// Внимание: Вы можете так же использовать строчку
//COptionalVariant SecondaryGeoField;
// вместо

```

```

SecondaryGeoField.vt = VT_ERROR;
SecondaryGeoField.scode = DISP_E_PARAMNOTFOUND;

// позволим марх автоматически определить geofield
GeoField.vt = VT_ERROR;
GeoField.scode = DISP_E_PARAMNOTFOUND;

// позволим марх найти какой слой связывается

BindLayerName.vt = VT_ERROR;
BindLayerName.scode = DISP_E_PARAMNOTFOUND;

// используем все поля по умолчанию
Fields.vt = VT_ERROR;
Fields.scode = DISP_E_PARAMNOTFOUND;

// установим имя нашего набора данных
Name.vt = VT_BSTR;
// Помните, что SysAlloc() это строинг; он входит в
BSTR
Name.bstrVal = strName.AllocSysString();

// установим источник данных - без проверки ошибки
его размещения
Type = miDataSetGlobalHandle;
SourceData.vt = VT_I4;
SourceData.lVal = (long)hGlobalData;

try {

```

```

        // теперь добавим набор данных к коллекции
datasets
        CMapXDataset ds =
m_ctrlMapX.GetDatasets().Add(Type,
        SourceData, Name, GeoField,
SecondaryGeoField,
        BindLayerName, Fields);
    }
    catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    }
}

```

miDataSetOLEData (PowerBuilder)

Это для контейнеров, таких как PowerBuilder, которые передают данные в MapX в формате, описанном в **miDataGlobalHandle** когда инициализируется управление MapX. Затем вызывается метод **Datasets.Add** с **miDataSetOLEData** в качестве типа, что означает для MapX создание набора данных с использованием предварительно переданных данных.

miDataSetUnbound (альтернативный метод)

Если Вы не можете поддерживать один из следующих форматов, MapX обеспечит Вам альтернативный подход для связывания данных, что позволит организовать цикл событий, где MapX опрашивает контейнер на предмет значений данных, по одной ячейке за раз. MapX пошлет событие **RequestData**, с номером колонки и строки, откуда запрашиваются данные. Вы можете затем использовать любой метод получения данных и подачи их через это событие обратно в MapX.

Пример: Использование RequestData для связывания “несвязанных” наборов данных

```
'Этот пример демонстрирует использование
Datasets.Add и
' события RequestData поддерживаемого MapX для
построения несвязанного набора данных
'   , который позволяет MapX получать доступ к
данным, чей формат
'   известен только программисту. В этом примере,
данные
'   в форме двумерного массива в VB.
Private Const kNumberOfRows = 3
Dim theData(1 To 3, 1 To 2) As Variant

Private Sub Form_Load()
    theData(1, 1) = "ME"      'Fill in the data to
    theData(2, 1) = "NH"     'be used by RequestData
    theData(3, 1) = "VT"
    theData(1, 2) = 100
    theData(2, 2) = 200
    theData(3, 2) = 300

    'Увеличение слоя New England
    Map1.ZoomTo 800, -70.26, 44.05
End Sub

Private Sub Command1_Click()
    Dim flds As New MapXLib.Fields
    Dim ds As Dataset

    'Описание структуры несвязанного набора данных:
```

```

        flds.Add "State", "State",
miAggregationIndividual, _
            miTypeString
        flds.Add "Sales", "Sales", miAggregationSum,
miTypeNumeric

'Создание несвязанного набора данных. Событие
"RequestData" будет запускаться для получения
используемых данных
'
        Set ds = Map1.Datasets.Add(miDataSetUnbound,
Nothing, _
            "My Dataset", "State", , , flds)

'Создание тематического слоя, базирующегося на
колонке "Sales" в
'несвязанном наборе данных
        ds.Themes.Add miThemeGradSymbol, "Sales", "My
Theme"
End Sub

Private Sub Map1_RequestData(ByVal DataSetName As
String, _
                                ByVal Row As Long, ByVal
Field _
                                As Integer, Value As
Variant, _
                                Done As Boolean)

        Done = false

```

```
    If DataSetName <> "My Dataset" Or Row >  
kNumberOfRows Then  
        Done = True  
    Else  
        Value = theData(Row, Field)  
    End If  
End Sub
```

miDataSetNotesView, miDataSetNotesQuery (Lotus Notes)

Эти типы связывания данных предназначены для Lotus Notes. Более подробно смотрите главу "*Начинаем работу с MapX*".

11

Доступ к данным СУБД (DBMS)

Доступ к данным СУБД
Применение метода Layers.Add с использованием объектов LayersInfo
Доступ к удаленным таблицам через .tab файл
Создание точечных объектов для данных СУБД по значениям в колонках X/Y
Поддержка Oracle8i
Параметры DBMS LayerInfo
Доступ к атрибутивным данным
О производительности
Кэширование
Каталог карт MapInfo
Загрузка пространственных данных в СУБД
Создание каталога карт MapInfo вручную
Геокодирование таблицы DBMS
Синтаксис определения стиля Symbol, Pen, Brush
Устранение неисправностей

Доступ к данным СУБД (DBMS)

MapX обеспечивает прямой доступ к серверам пространственных данных (Spatial Server Access). Это полезное свойство, которое позволяет разрабатывать приложения использующие непосредственно данные, хранящиеся на серверах пространственных данных, таких как MapInfo's SpatialWare, работающих в составе баз данных Oracle, Informix, DB2, или в базе данных Oracle 8i Spatial. Серверы пространственных данных позволяют организациям размещать картографическую информацию в базе данных предприятия, что обеспечивает централизованное и безопасное хранение всей производственной информации. Серверы пространственных данных SpatialWare дополнительно обеспечивают расширенную обработку запросов, повышают производительность сервера и оптимизируют организацию хранения пространственных данных.

Доступ к удаленным пространственным данным

С помощью MapX можно показывать данные, хранящиеся на разных серверах СУБД, в виде слоев карт. Список серверов включает::

- MS Access/SQL Server/другие источники данных ODBC .
- Серверы пространственных данных, такие как — SpatialWare для Oracle, SpatialWare Informix DataBlade или SpatialWare DB2 Extender.
- MapInfo Geocoding DataBlades для Informix.
- Oracle8i Spatial.

Добавить слой карты на основе данных, хранящихся в СУБД, можно, применяя метод [Layers.Add](#). Существуют два способа:

- В случае, если запрос должен быть динамически вычислен при его выполнении, нужно применять метод [Layers.Add](#) с использованием объектов LayerInfo. Такой подход соответствует случаю обработки данных, которые до этого не обрабатывались для показа на карте. Например, в приложении, которое строит карту расположения торговых точек, имеющих оборот, значение которого вводится пользователем. Значение величины оборота торговой точки полученное из приложения не может быть известно при разработке приложения и должно включаться в запрос перед его исполнением. Этот способ заменяет ранее использовавшийся метод [AddServerLayer](#).
- В случае, если значения получаемые в результате выполнения запроса могут быть вычислены заранее, целесообразно использовать Layers.Add напрямую через .tab файл. Например, приложение должно показать на карте все работоспособные пожарные гидранты. Запрос к базе данных, выбирающий все пожарные гидранты имеющие в поле 'работоспособность' ("service") значение истинно (true), может быть построен и выполнен заранее. Файл .tab может быть помещен в геонабор (geoset), при этом гарантируется загрузка необходимых данных при запуске любого приложения использующего этот геонабор (geoset).

Подробнее оба этих способа добавления пространственных данных описаны в нижеследующих разделах. После того, как Вы выбрали способ, с помощью которого будете обращаться к пространственным данным, используйте рекомендации из соответствующего раздела.

Внимание: MapX 4.0 не может открывать таблицы SpatialWare 3.0 или более ранних версий.

Применение метода `Layers.Add` с использованием объектов `LayersInfo`

В случае если `Layers.Add` успешно завершился, но данные не были показаны на карте, в первую очередь необходимо проверить порядок показа слоев. Нужный слой может быть прорисован на карте и затем закрыт другим слоем. Порядок слоев можно изменять при помощи свойства `LayerInfo`. Если же слой СУБД является самым верхним или единственным слоем в окне карты, самой вероятной причиной отсутствия графических объектов на карте является использование слишком малого значения стандартного масштаба показа карты. Настроить стандартное значение масштаба показа карты хранящейся на удаленном сервере можно установкой соответствующего значения в `MapInfo_MapCatalog` при помощи программы `MapBasic` (`misetmbr.mbx`).

```
Dim lInfo As Object
Dim lStr As String
Set lInfo = CreateObject("mapx.layerinfo.4")
lInfo.Type = 4          ' layer type is RDB
lInfo.AddParameter "NAME", "city_125"          '
layer name
lInfo.AddParameter "TOOLKIT", "ODBC"          ' в случае
использования прямого соединения к базе данных
Oracle следует использовать идентификатор "ORAINET"
lInfo.AddParameter "CONNECTSTRING",
"DRIVER={SpatialWare 32 Bit
Driver4.00};HOST=Champagne;UID=oracle;UPWD=secret;U
ID=GEORGETOWN; PWD=secret;OSID=DB1;DLG=0"
lInfo.AddParameter "QUERY", "SELECT * FROM CITY_125"
lInfo.AddParameter "CACHE", "ON"
lInfo.AddParameter "MBRSEARCH", "ON"
```

```
g_map.Layers.Add lInfo  
Set lInfo = Nothing
```

Доступ к удаленным таблицам через .tab файл

MapX имеет доступ к данным СУБД "напрямую" или через связанную таблицу MapInfo Professional. Однако, связанная таблица доступна только для чтения и не может быть обновлена при помощи MapX. На самом деле отображаются данные из удаленной таблицы и информация из локальной копии удаленной таблицы не используется.

Для обеспечения доступа к удаленным картам можно создать .tab файл. Такой файл создается при выполнении команд меню MapInfo Professional **Файл > Открыть таблицу DBMS**.

Файл .tab представляет собой текстовый файл; его можно создать при помощи любого текстового. После того как Вы создали файл .tab, доступ из приложения к нему организовать как к любому файлу MapInfo или программно, используя метод [Layers.Add](#), или применяя утилиту менеджер Geoset.

Пример (ODBC)

Ниже приводится образец файла .tab, обеспечивающий доступ к слою SpatialWare при помощи ODBC.

```
!table  
!version 500  
!charset WindowsLatin1  
  
Definition Table  
Type ODBC  
begin_metadata
```

```

"\IsReadOnly" = "FALSE"
"\DATALINK" = ""
"\DATALINK\Query" = "SELECT * FROM CITY_125"
"\DATALINK\ConnectionString" = DRIVER={SpatialWare
32 Bit
Driver4.00};HOST=Champagne;UUID=oracle;UPWD=secret;U
ID=GEORGETOWN; PWD=secret;OSID=DB1;DLG=0
"\DATALINK\Toolkit" = "ODBC"
"\CACHE" = "OFF"
"\MBRSEARCH" = "ON"
end_metadata

```

Пример (Oracle Direct)

Ниже приводится образец файла .tab, обеспечивающий доступ к слою Oracle Spatial при обращении напрямую.

Внимание: пример Oracle файла .tab явно использует пароль, установленный для соединения. При отсутствии пароля, отрабатывается ошибка. MapX не имеет встроенного диалога ввода пароля, в отличие от существующего в ODBC

```

!table
!version 550
!charset WindowsLatin1

```

Definition Table

```

Type ODBC
begin_metadata
"\IsReadOnly" = "FALSE"
"\DATALINK" = ""
"\DATALINK\Query" = "select "NAME", "SW_MEMBER",
"MI_SQL_RECNUM", "OBJECT" from
"MAPX"."RDBSDATA"

```

```

"\DATA LINK\ConnectionString" =
"SRVR=superior;UID=mapx;PWD=mapx"
"\DATA LINK\Toolkit" = "ORAINET"
"\CACHE" = "OFF"
"\MBRSEARCH" = "ON"
end_metadata

```

Создание точечных объектов для данных СУБД по значениям в колонках X/Y

Слой карты можно создать на основе таблицы СУБД, которая содержит координаты X/Y. Необходимо создать Map Catalog, зарегистрировать таблицы как SpatialType 4.0 и указать две колонки - контейнеры координат. Колонки должны быть проиндексированы. После этого, при установлении нового соединения к таблице СУБД через ODBC, можно обращаться в запросах к новым колонкам как к объектам типа "Object".

Пример

```

Private Sub AddMSAccessPointLayer()

On Error GoTo TRAP
    Dim lInfo As Object
    Set lInfo = CreateObject("mapx.layerinfo.4")
    lInfo.Type = 4          ' layer type is RDB
    Dim ConnectStr As String
    ConnectStr = "DRIVER={Microsoft Access Driver
(*.mdb)};DBQ=C:\Programs\MapInfo\Data\USA\ACCESS\US_
CUST.MDB;Drive rId=25;FIL=MS
Access;MaxBufferSize=512;PageTimeout=5"

```



```

lInfo.AddParameter "connectstring", ConnectStr
lInfo.AddParameter "name", "TempConnect"
lInfo.AddParameter "toolkit", "ORAINET"
lInfo.AddParameter "query", "select
`MI_SQL_RECNUM`, `OBJECT`, `Capital`,
`Province_Name`, `Pop_1991`, from `Can_caps`"
lInfo.AddParameter "cache", "OFF"
lInfo.AddParameter "mbrsearch", "ON"
g_map.Layers.Add lInfo
Set lInfo = Nothing
Exit Sub

```

TRAP:

```

MsgBox Err.Number & ", " & Err.Description

```

End Sub

Загрузить таблицу MapInfo, содержащую точечные объекты, например, CITY_125, в MSAccess или SQL Server можно при помощи программы MIUpload.MBX (находится в каталоге TOOLS рабочего каталога MapInfo Professional) в MapInfo Professional.

Поддержка Oracle8i

Поддержка Oracle8i является новой особенностью MapX 4.0. Её можно установить в дополнение к соединению MapX с базами данных через ODBC. Oracle8i Spatial — это новая реализация пространственной базы данных, выпущенная корпорацией Oracle. Хотя она и имеет общие черты с более ранними реализациями Oracle SDO, она имеет и существенные отличия. Oracle8i Spatial обслуживает хранимую в Oracle SDO информацию средствами общими для любых реляционных схем. Следует учитывать, что MapInfo не может поддерживать связь с реляционной схемой Oracle SDO через Oracle Call Interface (OCI). MapX может поддерживать несколько одновременных соединений к Oracle8i через OCI и к другим базам данных через ODBC. С помощью MapX нельзя загрузить из базы данных Oracle8i пространственные геометрические таблицы через ODBC соединения, использующие существующие в настоящее время драйверы ODBC компании Intersolv.

Требования к Oracle8i

Для того чтобы соединиться с Oracle8i из MapInfo, необходимо наличие установленного клиентского ПО Oracle8i v. 8.1.5. Подробнее смотрите в документации по Oracle.

О диалоге установления соединения

При применении Oracle8i не существует окна диалога, которое может быть использовано для ввода параметров установления соединения с этой базой данных. Поэтому необходимо явно вводить полную строку установления соединения. Если не передать идентификатора пользователя и пароль, то соединение установлено не будет. Поскольку при создании таблицы DBMS ".tab" файл, полученный при обращении к базе данных из MapInfo Professional, не содержит пароля (из соображений безопасности), то прямое использование такого файла не приведет к установлению соединения. Эту проблему можно обойти с помощью создания простого диалога запрашивающего пароль пользователя и выполняющего метод Layers.Add к слою таблицы СУБД (DBMS) используя layerinfo и передавая строку установления соединения к нужной базе данных. Строка установления соединения с Oracle должна иметь следующий вид:

```
"SRVR=superior;UID=user1;PWD=secret;"
```

Добавьте временный слой, использующий этот метод. Необходимое соединение с базой данных будет добавлено в набор используемых соединений. После этого оно будет использоваться всеми файлами ".tab" связанными с той же базой данных открытыми пользователем с таким же идентификатором, поскольку этот пользователь уже был опознан.

Для файлов ".tab" добавленных в геонабор (geoset) может потребоваться включение необходимых действий, обеспечивающих регистрацию пользователей обращающихся к базам данных в процедуры обрабатываемые при запуске приложения. Могут быть использованы соединения и регистрации на нескольких серверах и таблицы открываемые на них будут обрабатываться.

Как избежать ошибки тесселяции в 8.1.5

Пространственные запросы Oracle8i могут не выполняться из-за ошибки тесселяции (разбиения пространства на четверти, содержащие геометрические объекты, для организации пространственных индексов). Такие ошибки могут проявляться при слишком больших значениях масштаба. Нижеприведенный пример позволяет избежать таких ошибок:

1. Шаг 1 нужно проделать только один раз.

Используя имя пользователя Oracle MDSYS выполните запрос:

```
CREATE VIEW all_sdo_index_metadata as
  select SDO_INDEX_OWNER, SDO_INDEX_TYPE,
         SDO_INDEX_NAME, SDO_INDEX_TABLE,
         SDO_INDEX_PRIMARY,
         SDO_TSNAME, SDO_COLUMN_NAME,
         SDO_RTREE_HEIGHT, SDO_RTREE_NUM_NODES,
         SDO_RTREE_DIMENSIONALITY, SDO_RTREE_FANOUT,
         SDO_RTREE_ROOT, SDO_RTREE_SEQ_NAME,
         SDO_LEVEL, SDO_NUMTILES, SDO_MAXLEVEL, _
         SDO_COMMIT_INTERVAL,
         SDO_FIXED_META, SDO_TABLESPACE,
         SDO_INITIAL_EXTENT,
         SDO_NEXT_EXTENT, SDO_PCTINCREASE,
         SDO_MIN_EXTENTS,
         SDO_MAX_EXTENTS
from SDO_INDEX_METADATA_TABLE
where
  (exists
    (select table_name from all_tables
     where table_name=sdo_index_table and
     owner=sdo_index_owner)
   or
```

```

exists
    (select view_name from all_views
     where view_name=sdo_index_table and
owner=sdo_index_owner)
    or
exists
    (select table_name from all_object_tables
     where table_name=sdo_index_table and
owner=sdo_index_owner)
);
grant select on all_sdo_index_metadata to public;
create public synonym all_sdo_index_metadata for
all_sdo_index_metadata;
2. В схеме принадлежащей пользователю, который хочет
    предоставить возможность выбора по отношению к своим
    таблицам, выполните:
grant select on sdo_geom_metadata to <user>;
grant select on <table> to <user>;
grant select on <sdo_index_table> to <user>;

```

Параметры DBMS LayerInfo

И файлы ".tab" и объекты LayerInfo могут использовать одинаковые параметры. Однако, правила поименования имен параметров немного отличаются.

Параметр	Описание
Query	<p>Это запрос к серверу СУБД (DBMS), результатом выполнения которого будет таблица данных для приложения. Если необходимо, чтобы таблица была показана на карте, то запрос должен обращаться к колонкам, содержащим пространственные объекты (или псевдо колонке OBJECT). Пользователи SpatialWare могут включать в запрос специфичные для GISSQL стандарта предложения SQL и строки, с помощью которых возвращаются пространственные объекты. Кроме того, пользователи SpatialWare должны включать в запрос обращения к колонкам SW_GEOMETRY и SW_MEMBER, которые содержат географические объекты и индексы.</p> <p>Внимание: Если имя таблицы зависит от регистра в котором оно набрано и если в качестве идентификатора имен используется символ двойной кавычки ("), то двойные кавычки в запросе должны вводиться дважды. Например: "select * from ""tableowner"". ""tablename"""</p>
ConnectionString	<p>Содержит информацию необходимую для подключения к СУБД (DBMS). При использовании ODBC подключения обращение к источнику данных определяет строку подключения. Если указана полностью строка подключения то подключение устанавливается без появления окна диалога. Если же, при обращении к источнику данных не передается никакой строки подключения, то вызывается системный диалог "Выбор источника данных".</p> <p>При обращении к Oracle Spatial, необходимо в приложении указывать полную строку подключения, включая имя</p>

	<p>идентификатор пользователя (userid) и пароль. Ключевое слово DLG в этом случае неприменимо и диалог не появляется. Строка подключения к Oracle напрямую должна иметь такой вид:</p> <p>"SRVR=superior;UID=mapx;PWD=secret"</p> <p>Внимание: для законченных приложений рекомендуется использовать полностью заполненную строку подключения, с тем чтобы это подключение могло использоваться многократно.</p>
Cache	<p>Может принимать значения "On"/"Off". Стандартно используется значение "On". С помощью этого параметра разрешается или запрещается кеширование данных при отображении слоев. Кеширование увеличивает производительность при последовательных изменениях масштаба показа карты, перемещении окна карты (внутри района попавшего в кеш), подписывании и построении тематических карт. Если кешировать слишком много насыщенных объектами слоев, то памяти для создания кеша может не хватить и его содержание будет потеряно..</p>
MBRSearch	<p>Может иметь значения "ON"/"OFF". Стандартно используется "ON".</p> <p>При отображении данных на карте, MapX выбирает их из области ограниченной описывающим всю карту прямоугольником. Для слоев СУБД (DBMS) используется пространственный предикат, который добавляется к запросу определяющему допустимые значения таблицы. Для некоторых типов запросов или слоев, которыми они отображаются на карте, могут возникнуть сложности с использованием пространственных индексов. В этом случае применение этого предиката может замедлить обработку запроса или привести к невыполнению запроса. Отключая использование этого предиката, можно избежать таких ситуаций.</p>
Name	<p>Необходимый строковый параметр определяющий имя слоя или его синоним.</p>
ToolKit	<p>Может иметь значения "ODBC"/"ORAINET". По умолчанию</p>

	<p>применяется "ODBC"</p> <p>Определяет способ подключения к серверу СУБД (DBMS). Для прямого подключения к Oracle необходимо указать "ORAINET" и использовать строку соединения клиентской части ORAINET, для остальных следует использовать ODBC.</p>
--	---

Формат строки подключения к СУБД (DBMS)

Формат строки подключения к ODBC совпадает с аналогичным для подключения связанных таблиц ODBC в MapInfo Professional. Строка состоит из нескольких операторов разделенных точкой с запятой (;). Каждый оператор представляет собой пару определителей — Ключ-Значение. Важные ключи перечислены ниже:

Ключ	Описание
DSN=	<p>Объявляет имя источника данных ODBC.</p> <p>Внимание: при использовании оператора DSN указанное в нем имя источника данных ODBC должно совпадать с именем источника данных использованном на машине клиента. Обратите внимание, что имена источников данных утапливающие подключение к одному и тому же серверу у разных пользователей могут иметь разные имена. Если заранее не известно какое имя источника данных будет использовано, следует использовать оператор DRIVER вместо оператора DSN.</p>
DRIVER=	<p>Объявляет точное имя установленного драйвера SpatialWare или Informix Universal Sever. Может быть использован вместо оператора DSN. Пример:</p> <p>DRIVER={SpatialWare 32 Bit Driver}</p> <p>Внимание: Informix 2.80.0861 не поддерживает обращение к нему через оператор DRIVER=.</p>
UID=	Объявляет, в случае необходимости, имя пользователя

	обращающегося к источнику данных.
PWD=	Объявляет, в случае необходимости, пароль обращающегося к источнику данных пользователя.
DLG=	Число, с помощью которого можно управлять появлением диалога "Выбор источника данных": 0 – Исключает возможность появления этого диалога (является обязательным при построении тематических карт MapX). 1 – Показывает диалог установления подключения. 2 – Показывает диалог подключения, но только при необходимости (в случае, если не вся необходимая информация была передана) — используется по умолчанию.
SRVR=	Определяет значение передаваемое в служебную программу Oracle8i EasyConfig. Необходимый оператор для при подключении к Oracle8i.

Подключения можно использовать повторно, избегая таким образом многократных вызовов диалогов. Если передать полную строку установления подключения, выдержав правильную последовательность передачи параметров (см. "Примеры строк подключения"), которая в точности воспроизводит уже использованную для подключения из набора уже установленных, то такое соединение будет использовано для двух таблиц. Пароль может не использоваться при установлении повторного совместного подключения.

Примеры строк подключения

Ниже приводятся примеры установления подключений через ODBC драйверы к IDS/UDO и SpatialWare.

Строка подключения к IDS/UDO

```
DRIVER={INFORMIX 2.80 32  
BIT};UID=informix;PWD=secret;DATABASE=sw;HOST=adak;S  
ERVER=adak_tli ;SERVICE=sqlexec;PROTOCOL=onsoctcp;
```

Строка подключения к SpatialWare:

```
DRIVER={SpatialWare 32 Bit  
Driver3.50};HOST=fire;UID=oracle;UPWD=secret;UID=GE  
ORGETOWN;PWD=G EORGETOWN;OSID=QASW1
```

Доступ к атрибутивным данным

Для того чтобы, получить данные из всех колонок, можно составить запрос вида "Select * From tablename". Использовать маскирующий символ "*" (звездочка) необязательно. Можно явно указать именно те колонки, которые нужно использовать. Для того чтобы увеличить производительность следует ограничивать глубину запроса только теми колонками, которые нужны.

Для того чтобы обеспечить доступ к атрибутивным данным из слоев СУБД (DBMS), содержащих пространственные объекты, следует использовать метод [datasets.add](#) к набору данных типа [midatsetlayer](#) (получая атрибуты пространственных объектов уже существующих слоев).

Добавляя слой СУБД (DBMS), для лучшей производительности следует в запросе обращаться только к тем колонкам, которые должны быть использованы в приложении. Такими являются: колонки содержащие пространственные объекты и колонка(и), по которым построены индексы. Эти колонки запрашиваются автоматически, даже если они не объявлены формально в запросе. Кроме того, глубину запроса следует ограничивать колонками, значения в которых будут использованы для создания подписей или тематических карт. Можно включать в запрос псевдо колонку OBJECT, обращаясь к любой таблице с присоединенной геоинформацией. При этом будут запрашиваться колонки реально содержащие пространственные данные. Такой способ формирования запроса является обязательным при обращении к колонке MapMarker MDIGEOADDRESS или к таблице с колонками координат X/Y.

Примечание: Можно использовать любое выражение или функцию, выполняемое на сервере, которое обеспечивает определение колонки. Кроме того избегайте использования в реальном приложении запросов вида `select * from tab`.

Следующий пример кода программы показывает как открыть файл MapInfo (*.tab) и после этого выбрать удаленные данные и связать их с отображением на слое. В результате становится возможным формировать подписи и строить тематические карты по колонкам удаленной таблицы:

```
Dim Lay As Layer
Dim dsname As String
Set Lay = Map1.Layers.Add(filename)
dsname = Lay.Name
Map1.Datasets.Add miDataSetLayer, Lay, dsname
```

О производительности

Подключение к серверу баз данных может длиться несколько секунд. Эта задержка наблюдается при первичном установлении подключения. Операция "Открыть" тоже может выполняться несколько секунд. Эта единственная задержка наблюдается каждый раз, когда открывается новая таблица.

Скорость показа карты зависит от объема данных полученных с сервера. В некоторых случаях скорость показа данных, построенных на основе таблиц сервера, заметно медленнее, чем из локального файла. Скорость зависит также от того, попали ли данные, необходимые для показа на карте, в кеш MapX.

Кэширование

Для увеличения производительности слой СУБД (DBMS) может быть кэширован. Кэширование полезно если приложение использующее данные СУБД требует перерисовок, построения тематических карт и создания подписей. По умолчанию кэширование включено. Для кэширования требуется достаточно много памяти, поэтому приходится иногда выбирать какие данные кэшировать, а какие нет.

Слой ODBC кэшируется в выделенной памяти при считывании и отображении в окне карты приложения MapX. При последующих перерисовках нужные данные получаются из кэша, а не запрашиваются вновь из базы данных с сервера. Кэширование способно существенным образом увеличить производительность при перерисовках.

При навигации по карте или при изменении масштаба показа карты таким образом, что данные попавшие в кэш в соответствии с МОП (минимальным описывающим прямоугольником) окна карты, кэш будет автоматически обновлен из данных с сервера в соответствии с новым МОП.

При необходимости обновить данные в кэше без навигации по карте или изменения масштаба показа окна карты данными, которые потенциально могли измениться и лежат внутри этой географической области, следует использовать метод `Layer.Refresh`.

Внимание: Если поместить в кэш слишком много данных или слишком много слоев, то может начать использоваться виртуальная память и преимущества по производительности скорее всего исчезнут.

Каталог карт MapInfo

Если необходимо обеспечить доступ приложения MapX к удаленным пространственным данным, то в составе СУБД (DBMS) должна присутствовать специальная таблица — каталог карт, под именем MapInfo Map Catalog. Одна такая таблица должна быть создана для каждой базы данных таблицы которой должны быть показаны в виде слоев карт в приложениях MapX. Каталог карт должен содержать информацию о пространственных колонках в каждой из таблиц с геоинформацией к которым может быть получен доступ в базе данных.

Приложение MapX может использовать уже существующий в базе данных на сервере каталог карт MapCatalog (один и тот же каталог может использоваться разными клиентскими программами MapInfo, включая MapInfo Professional). Если в базе данных не существует каталога карт, то придется его создать.

Использование MapInfo Professional для настройки каталога карт

Если Вы являетесь пользователем MapInfo Professional, то эту программу можно использовать для создания и настройки каталога карт MapInfo Map Catalog. В состав MapInfo Professional входят служебные программы MapBasic, такие как MIODBCAT.MBX, которые помогают в создании и настройке каталога карт.

Каталог карт MapInfo Map Catalog созданный с помощью MapInfo Professional может использоваться приложениями MapX. Например, при использовании корпоративных данных один пользователь - администратор базы данных с помощью MapInfo Professional создает и настраивает каталог карт, а неограниченное число пользователей будет использовать приложения MapX, читающие записи из этого каталога карт.

Загрузка пространственных данных в СУБД

Если у Вас есть пространственные данные в виде таблиц MapInfo, то существует возможность импортировать их в базу данных SpatialWare при помощи :

- служебной программы MISWUP32.MBX, которая находится в каталоге, создаваемом при установке клиентского программного обеспечения SpatialWare
- или служебной программы EasyLoader для Informix Universal Server
- DB2.
- Oracle Spatial

Создание каталога карт MapInfo вручную

В случае, если Вам необходимо создать Каталог Карт MapInfo вручную и Вы не являетесь пользователем MapInfo Professional, то делать это должен Администратор базы данных

1. Создайте пользователя MAPINFO для конкретной базы данных, где расположены таблицы, которым предполагается хранить геоинформацию.
2. Создайте в базе данных таблицу MAPINFO_MAPCATALOG. Оператор Create Table, применяемый в конкретной СУБД для создания таблицы, должен быть эквивалентным следующему оператору MapInfo

```
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)  
)
```

Внимание: Структура таблицы должна точно соответствовать вышеприведенному оператору. Единственное изменение возможно для СУБД, поддерживающих типы данных Varchar или Text, которые могут применяться вместо типа Char.

3. Создайте уникальный индекс для колонок TABLENAME и OWNERNAME так, что только одной таблице для каждого владельца могла быть присвоена геоинформация.

```
create index mapcat_i1  
on mapinfo.mapinfo_catalog (OwnerName,TableName)
```

4. Предоставьте привилегии доступа Select (выборка), Update (обновление) и Insert (вставка) для таблицы MAPINFO_MAPSCATALOG, что позволит пользователям присваивать таблицам ODBC геоинформацию. Привилегия Delete (удаление) должна быть предоставлена администратору базы данных.

Геокодирование таблицы DBMS

Для каждой таблицы содержащей пространственные данные, с которыми будет работать приложение MapX, требуется добавить запись в таблицу каталога карт MAPINFO_MAPCATALOG. Если Вы не имеете возможности использовать MapInfo Professional для создания и управления каталогом карт, то придется вручную добавить необходимые записи. Ниже описываются синтаксис и возможные значения для конкретных баз данных.

Имя колонки	Значения	Примеры
SPATIALTYPE	<p>Внимание: в этой колонке описываются формат хранения и индексирования пространственных объектов и тип(ы) поддерживаемых или не поддерживаемых данных конкретной базой данных. Цифра левее десятичной точки описывает формат пространственного объекта. Цифра правее десятичной точки описывает тип пространственного объекта, который может храниться в конкретной базе данных.</p> <p>Формат пространственных объектов MapInfo</p> <p>1.x: Слой точек в колонках X/Y индексированный методом micode (разбиением на квадранты по ключу)</p> <ul style="list-style-type: none">• 2.x: Oracle MD/SDO версии 1 HNCODE_ - Не поддерживается• 3.x: Oracle MD/SDO версии 1	5.3

	<p>HNCODE_PARTITIONED - Не поддерживается</p> <p>4.x: Слой точек в колонках X/Y</p> <ul style="list-style-type: none"> • 5.x: SpatialWare for Oracle • 6.x: Ingres SOL - Не поддерживается • 7.x: Sybase SQS - Не поддерживается • 8.x: Oracle SDO версии 2 - Не поддерживается <p>9.x: модуль MapInfo Geocoding DataBlade SpatialWare для точек</p> <ul style="list-style-type: none"> • 10.x: модуль MapInfo Geocoding DataBlade по XY • 11.x: SpatialWare IUS datablade • 12.x: SpatialWare Extender для DB2 • 13.x: колонка Oracle Spatial <p>Spatial Object Type</p> <ul style="list-style-type: none"> • x.0: только точки • x.1: только линии • x.2: только полигоны • x.3: все типы поддерживаются 	
TABLERNAME	Имя таблицы.	DRAINAGE
OWNERNAME	Владелец таблицы.	GEORGETOWN
SPATIALCOLUMN	<p>Имя колонки, если такая существует, в которой хранится пространственная информация:</p> <ul style="list-style-type: none"> • SW_GEOMETRY (при использовании SpatialWare Type/IUS) • NULL_COLUMN (при использовании прямого хранения координат XY) • MI_SQL_MICODE (при использовании метода MI Code) 	SW_GEOMETRY

	<ul style="list-style-type: none"> • или имя колонки IUS, DB2или Oracle типа ST_SPATIAL. • имя колонки Oracle 8i SDO_GEOMETRY. 	
DB_X_LL	Координата X нижнего левого угла описывающего прямоугольника в единицах соответствующих COORDINATESYSTEM (см. ниже)..	360
DB_Y_LL	Координата Y нижнего левого угла описывающего прямоугольника.	90
DB_X_UR	Координата X правого верхнего угла описывающего прямоугольника.	360
DB_Y_UR	Координата Y правого верхнего угла описывающего прямоугольника.	90
COORDINATESYSTEM	Строка символов эквивалентное предложению MapInfo CoordSys (в самом начале не нужно использовать ключевое слово "CoordSys"), определяющее проекцию, единицы измерения и т.п. Для карт в проекции Широта/Долгота следует указывать "Earth Projection 1,0"	Earth Projection 1,0
SYMBOL	Предложение MapInfo Symbol (если на слое есть только точки) или предложение Symbol, вслед за которым предложение Pen (описывающее стиль оформления линейных объектов), предложение Pen (описывающее стиль оформления границ площадных объектов) и, последнее, предложение Brush.	Symbol(35,0,12) Pen(1,2,0) Pen(1,2,0) Brush(2,255,255)
XCOLUMNNAME	Для таблиц, в которых координаты X/Y хранятся напрямую в колонках, следует указать имя колонки с X координатами. Если такой колонки не существует (т.е. для хранения пространственных координат	NO_COLUMN

	используется единственная колонка таблицы вместо пары колонок XY), то следует либо явно указать	
YCOLUMNNAME	Для таблиц, в которых координаты X/Y хранятся напрямую в колонках, следует указать имя колонки с Y координатами или ввести значение NO_COLUMN.	NO_COLUMN

Синтаксис определения стиля Symbol, Pen, Brush

Если Вы в ручную создали таблицу каталога карт MAPINFO_MAPCATALOG, обеспечивающую доступ к удаленным пространственным данным, то в ней необходимо указать стиль оформления символов и, возможно, стиль оформления линий и заполнения областей.

Объявление стиля символов

Для того, чтобы объявить стиль оформления символов, необходимо использовать предложение Symbol. Существует три варианта структуры предложения Symbol: первый —, для символов типа MapInfo 3.0, второй —, для символов из шрифтов TrueType, и третий —, для растровых символов.

Синтаксис	Пример
Symbol(форма, цвет, размер) или Symbol(форма, цвет, размер, шрифт, стиль шрифта, поворот) или Symbol(имя файла	Symbol(35,0,12) Symbol(64,255,12,"MapInfo Weather" ,17,0)

растра,цвет,размер,настрройки)	Symbol("sign.bmp", 255, 18, 0)
--------------------------------	--------------------------------

Объявление стиля линий

Следует использовать предложение Pen для объявления стиля оформления линий. В каталоге карт Map Catalog может потребоваться объявить два разных стиля оформления линий: один для линейных объектов, другой для границ областей

Синтаксис Pen	Пример
Pen(толщина, штриховка, цвет)	Pen(1, 2, 0)

Объявление стиля областей

Следует использовать предложение Brush для объявления стиля оформления замкнутых полигонов (областей)

Синтаксис Brush	Пример
Brush(штриховка,цвет,цвет фона)	Brush(2, 255, 65535)

Устранение неисправностей

В случае если возникнут проблемы с использованием приложений MapX и SpatialWare, Вы можете использовать таблицу, приведенную ниже, для анализа ситуации и поиска способа решения проблемы.

Описание проблемы	Возможная причина	Решение
Сообщение MapX "Слой не согласован."	Попытка Связать данные со слоем SpatialWare.	Связывание с данными из слоев SpatialWare средствами MapX в текущей версии не поддерживается.
Объекта по указанному индексу не существует.	<ul style="list-style-type: none"> Запрос был осуществлен к несуществующей таблице Пространственного объекта отвечающего критериям запроса не получено. Запрос был сделан не к таблице SpatialWare. 	<ul style="list-style-type: none"> Проверить что таблица с таким именем существует и ее имя набрано в правильном регистре. Кроме того она должна содержать геоинформацию. Для создания таблиц SpatialWare используйте служебную программу загрузки пространственных данных Upload. Проверьте запрос, может быть он содержит синтаксические ошибки. Кроме того, убедитесь, что в запросе содержится обращение к колонке, объявленной содержащей пространственные объекты в

		MapInfo_Mapcatalog.
Значение MapX datasets.rowcount равно нулю.	Набор данных (dataset) был создан на основе данных сервера СУБД (DBMS).	Значение параметра datasets.rowcount для наборов данных (dataset), созданных на основе данных сервера СУБД (DBMS), всегда будет равно нулю. При итерации последовательно по набору данных (dataset) возвращается код ошибки, который можно использовать как флаг состояния превышения диапазона значений для такого набора данных.
Карта имеет неправильный масштаб показа. Например, масштаб показа установлен таким, что все объекты сливаются в одну точку.	МОП (минимальный описывающий прямоугольник) для данных SpatialWare определяется в таблице MapInfo_Mapcatalog, значения точек углов которого установлено таким образом, что приводит к нежелательному масштабу показа карты в приложении.	Отредактируйте координаты (DB_X_LL, DB_X_UR, DB_Y_LL, DB_Y_UR) в таблице MapInfo_MapCatalog. Можно использовать служебную программу MapInfo Professional MSETMBR.MBX.

12

Тематическая картография и анализ

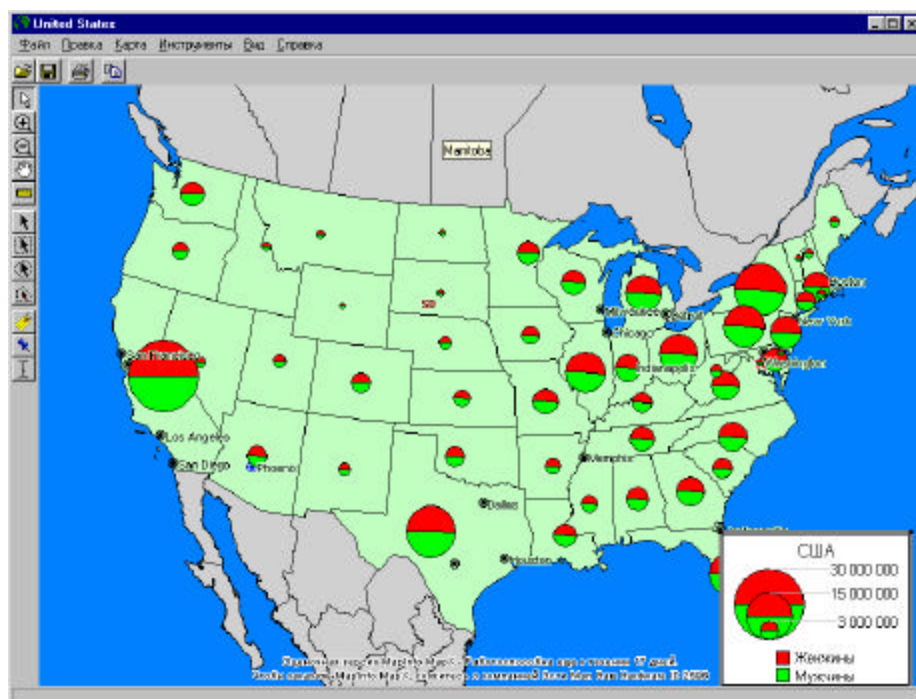
- ◆ Тематическая картография и анализ
- ◆ Что такое тематическая картография
- ◆ Разработка собственной тематической карты
- ◆ Типы тематических карт
- ◆ Работа с тематическими картами
- ◆ Настройка тематической легенды

Тематическая картография это мощный инструмент анализа и визуализации данных. Вы можете придать графическую форму данным, которые Вы видите на карте. Диаграммы и графики, а также тематическая раскраска карты значительно облегчают анализ пространственных данных.

Используя MapX, Вы можете создавать приложения с тематическими картами следующих типов: диапазоны значений, размерные символы, плотность точек, отдельные значения, круговые и столбчатые диаграммы.

Что такое тематическая картография?

Тематическая картография это процесс раскраски карты в соответствии со значениями каких либо параметров. Тематика это обычно только часть Ваших данных, которые извлекаются из набора данных. Тематические слои представляют данные визуализированы цветами, оттенками, штриховкой, символами или диаграммами круговыми или столбчатыми. Вы создаете разные тематические карты, присваивая графическим объектам эти цвета, штриховки или символы, в зависимости от определенных значений в данных. Столбчатые графики и круговые диаграммы позволяют делать данные сопоставимые для каждой записи в наборе данных. Есть много способов проиллюстрировать данные тематическими картами. Наиболее известный пример это карта погоды. Там где красный цвет, Вы знаете, что и температура там высокая (в градусах); где Вы видите синий цвет, там холодно (в градусах значения низкие). Тематическая картография также позволяет Вам обнаружить тренд в данных, что трудно увидеть в табличных данных. Используя методы и свойства в коллекции **Themes**, объекте **Theme** и объекте **ThemeProperties**, Вы можете создать и настроить Вашу собственную тематическую раскраску. Объект **Legend** содержит тематические ключи (theme's key), описывающие каким значениям соответствуют тематические цвета, символы и их размеры.



Разработка собственной тематической карты

Перед тем как Вы создадите тематическую карту, важно знать о тех элементах, из которых она состоит и как эти элементы сочетаются. В этом разделе обсуждаются тематические переменные, источник получения Ваших данных, способы использования данных из других таблиц, оформление и отображение тематических слоев.

Тематические переменные

Данные, которые Вы изображаете на тематической карты называются тематическими переменными. В зависимости от типа тематического анализа, на карте можно отразить одну или более тематических переменных. Диапазоны значений, растровая тематическая поверхность, размерные символы, плотность точек и отдельные значения - все работают только с одной переменной. Столбчатые и круговые диаграммы могут показывать более одной переменной одновременно.

Вы также можете создать двувариантную тематическую карту, где один объект карты, например символ представляет два значения - одно цветом символа а второе размером символа.

Где взять данные

Перед созданием тематической карты, надо решить, какая информация будет для этого использоваться и где она расположена. Данные используемые для создания тематики поступают из объекта **Field** или **Field** из вашего набора данных. Эти поля передаются как параметр **Fields** метода **Themes.Add** коллекции **Themes**.

Тематическая коллекция Themes

Каждый набор данных имеет тематическую коллекцию (collection of themes). Коллекция **Themes** создает, подсчитывает, добавляет или удаляет объект **Theme** из тематической коллекции.

Методы	Описание	Пример кода
Add	Создает тематический объект и добавляет его к коллекции Themes для обычного набора данных	Map1.Datasets(1).Themes.Add _ miThemeRanges “TotPop”, “My _ Ranges Theme”
Remove	Удаляет указанный тематический объект из коллекции.	Map1.Datasets(1).Themes.Remove “My _ Ranges Theme”
RemoveAll	Удаляет все тематические объекты из коллекции	Map1.Datasets(1).Themes.RemoveAll

Создание тематического объекта Theme

Если Вам надо раскрасить карту в тематические цвета используя данные из набора данных, то создайте объект **Theme** для набора данных с помощью метода **Themes.Add**. После того как набор данных добавлен к Вашей карте, создание тематического объекта может быть сделано одной строкой кода:

```
Map1.Datasets(1).Themes.Add miThemeRanges “TotPop”,  
—  
“My Ranges Theme”
```

Этот код создает тематический диапазон для первого набора в коллекции Datasets используя поле “TotPop” .

Метод **Themes.Add** описан ниже. Необязательные параметры в квадратных скобках [].

Синтаксис

Themes.Add [Type], [Field], [Name]

Часть	Описание
Type	Определяет тип создаваемого тематического слоя. Берется значение ThemeTypeConstants. Это необязательный параметр и если он не указан (или указан как miThemeAuto), MapX будет пытаться выбрать по умолчанию тематику, базирующуюся не числе передаваемых полей. Если MapX не может по умолчанию выбрать тип тематической карты, сгенерируется ошибка.
Field(s)	Указывает поле или поля для тематической карты. Поле может быть указано по имени, индексу или по объекту Field. Если Вы создаете тематический слой используя несколько переменных (например круговые или столбчатые диаграммы), в коллекцию Fields передается массив имен полей, индексов или объектов Field. Это необязательный параметр и если он не указан, MapX использует первое числовое поле набора данных DataSet.
Name	Указывает имя тематической карты. Это строковый параметр. Это необязательный параметр, и если он не указан, MapX генерирует имя, такое как StatesBySales.

Как MapX выбирает по умолчанию метод тематической картографии

Если параметр Type метода **Themes.Add** не указан (или указан как **miThemeAuto**), MapX будет пытаться выбрать подходящий вариант тематики основанный на числе полей, передаваемых для построения тематики, как описано ниже.

Но. Полей	Тип слоя	Алгоритм
более чем 1	Points, Lines, Regions	По умолчанию тип тематики это круговые диаграммы. Если тематический слой со множеством переменных уже существует,

		попытайтесь создать тематический слой с одной переменной, используя первое поле в коллекции полей (см. Алгоритмы ниже).
1	Points	Если связанное поле имеет значения, объединенные как индивидуальные, попытайтесь создать тематический слой отдельных значений. Если тематические слои диапазонов или отдельных значений уже существует, то создать тематический слой по умолчанию не удастся. Если связанное поле имеет значения не объединенные как индивидуальные, попытайтесь создать слой размерных символов. Если и такой тематический объект существует, попытайтесь создать диапазоны. Если диапазоны или отдельные значения уже существуют, то создать тематический слой по умолчанию не удастся.
1	Lines	Если связанное поле имеет значения, объединенные как индивидуальные, попытайтесь создать тематический слой отдельных значений. Если тематические слои диапазонов или отдельных значений уже существует, то создать тематический слой по умолчанию не удастся. Если связанное поле имеет значения не объединенные как индивидуальные, попытайтесь создать диапазоны. Если диапазоны или индивидуальные значения

		уже существуют, попытайтесь создать слой размерных символов. Если тематику уже существует, не удастся создать тематический слой по умолчанию.
1	Regions	Если связанное поле имеет значения, объединенные как индивидуальные, попытайтесь создать тематический слой отдельных значений. Если тематические слои диапазонов или отдельных значений уже существует, то создать тематический слой по умолчанию не удастся. Если связанное поле имеет значения не объединенные как индивидуальные, попытайтесь создать диапазоны. Если диапазоны или отдельные значения уже существуют, попытайтесь создать слой плотности точек. Если слой плотности точек уже существует, попытайтесь создать слой размерных символов. Если тематику уже существует, не удастся создать тематический слой по умолчанию.

Внимание: Круговые, столбчатые диаграммы и размерные символы это тематические объекты.

Определение типа тематического слоя определяется спецификой слоя. Другими словами, тематика Слоя А не рассматривается когда делается попытка определить тип тематики по умолчанию для новой тематики на слое В.

Типы тематических констант

Вы можете создать следующие типы тематики:

Тип	Описание
<code>miThemeRanged</code>	Диапазоны
<code>miThemeBarChart</code>	Столбчатые диаграммы
<code>miThemePieChart</code>	Круговые диаграммы
<code>miThemeGradSymbol</code>	Размерные символы
<code>miThemeDotDensity</code>	Плотность точек
<code>miThemeIndividualValue</code>	Индивидуальные значения
<code>miThemeAuto</code>	MapX “подбирает наилучшим образом” тематику

Каждый тип тематики имеет собственное назначение и уникальные атрибуты. Например, используйте **`miThemeRanged`**, и сможете тематически раскрасить диапазоны на карте мира в соответствии с плотностью населения. Можно раскрасить страны оттенками красного, самый темный будет соответствовать наибольшей плотности населения, а розовый - наименьшей плотности. Так Вы наглядно изобразите плотность населения планеты по странам.

Вы не ограничены в возможности представления числовых значений с помощью тематической картографии. Номинальные величины тоже могут быть изображены тематически. Например, у Вас есть набор данных по подземным кабелям. Те кабели, которые не обслуживались за последние шесть месяцев, подписаны преимущественным статусом. Используя **`miThemeIndividualValue`**, можно раскрасить кабели по приоритету их статуса. Все записи с одинаковым значением статуса получат одинаковую раскраску. Смотрите более подробную информацию в конце главы.

После того как тематический слой **Theme** добавлен к указанной коллекции, объект **Theme** создается и вы можете управлять свойствами объекта.

Объект Theme

Объект **Theme** устанавливает свойства тематического слоя внутри коллекции **Themes**. Изменение объекта **Theme** осуществляется методами, свойства определяют как будет выглядеть тематический слой, тип тематики и др.

Свойства	Описание
AutoRecompute	Контролирует, перевычисляются ли диапазоны тематического слоя. Это логическая величина, по умолчанию равна TRUE.
ComputeTheme	Контролирует, вычисляются ли тематические слои. Это логическое свойство и по умолчанию равно True. При значении True будут рассчитываться тематические слои для данных. Если установлено False будут созданы невидимые тематические объекты с 20 диапазонами для карты индивидуальных значений и 5 диапазонами для карты диапазонов. Вы можете затем вручную установить минимальное и максимальное значение для настройки тематического слоя.
DataMax	Определяет максимальное значение для установки тематических диапазонов или рассчитывает диапазоны равного размера когда свойство Theme.ComputeTheme или параметр ComputeTheme метода Themes.Add установлены на FALSE
DataMin	Определяет минимальное значение для установки тематических диапазонов или рассчитывает диапазоны равного размера когда свойство Theme.ComputeTheme или параметр ComputeTheme метода Themes.Add установлены на FALSE.
Fields	Возвращает доступную только для чтения коллекцию Fields, представляющую установку полей используемых в наборе данных, на которых базируется тематический слой.
Layers	Свойство только для чтения, возвращающее Объект Layer, представляет слой, на котором базируется тематический слой.
Legend	Каждый объект Theme имеет Объект Legend

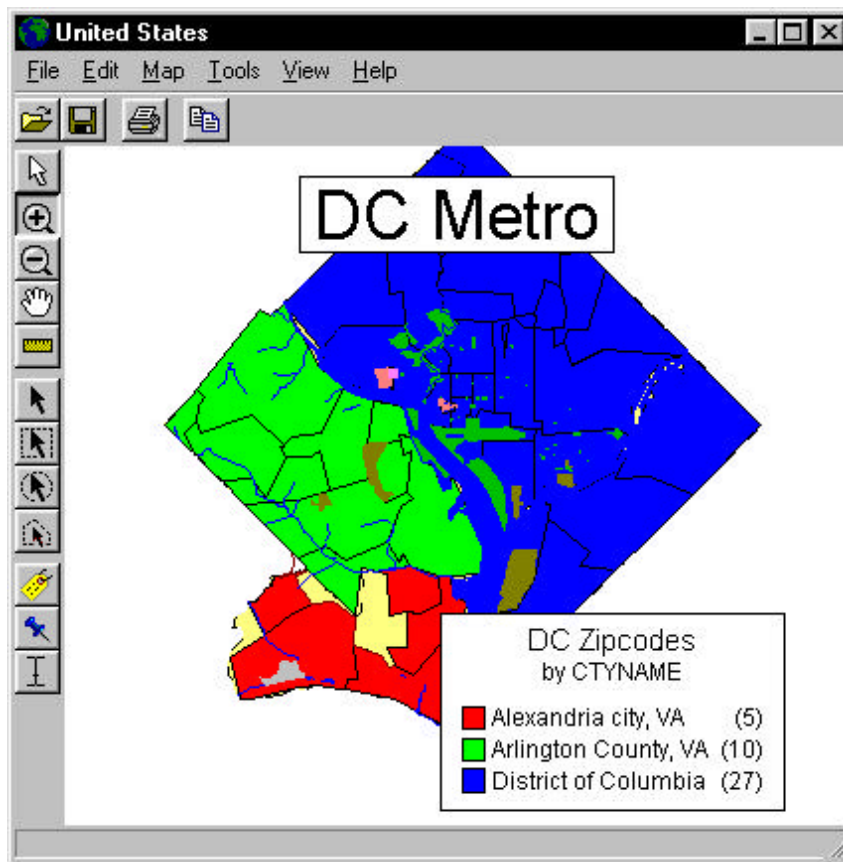
	(Theme.Legend). Объект Legend содержит свойства для контроля изображения тематической легенды. Каждый объект ThemeCategory (RangeCategory, IndividualValueCategory или MultiVarCategory) имеет вход в легенду, содержащуюся в объекте LegendText.
Name	Имя объекта Theme. Имя должно быть уникальным в коллекции Themes. Это свойство доступное для чтения/записи и определяется как параметр метода Themes.Add или генерируется программой MapX когда создается тематический слой. Это свойство по умолчанию для объекта Theme.
ThemeProperties	Объект, содержащий свойства, относящиеся к тематическому слою. Устанавливаемые свойства, содержащиеся в объекте ThemeProperties определяются типом объекта Theme (тип тематической карты).
Type	Тип тематической карты для объекта Theme. Это значение ThemeTypeConstants и свойство открыто только для чтения.
Visible	Устанавливает, является ли видимым тематический слой. Это логическое значение, и по умолчанию равно TRUE .

Типы тематических карт

Этот раздел посвящен общей информации об основных типах тематических карт.

Карты отдельных значений

Карты отдельных значений показывают линии, точки или границы, раскрашенные в соответствии с отдельными значениями в обычном поле набора данных. Вы можете использовать и числовые и номинальные величины в карте отдельных значений. MapX присваивает каждому уникальному значению его собственный цвет или символ. Когда карта отдельных значений использует типы символов, символы берутся из стиля заявленного по умолчанию.



Например, дистрибьютор безалкогольных напитков создал таблицу с супермаркетами, которые закупают такие напитки по почтовым кодам Zip Code в округе Колумбия. Каждый супермаркет продает напитки по своей цене. Если дистрибьютор раскрасит границы областей почтовых кодов по цене, используя отдельные значения, все заведения, продающие напитки по 49 центов будут раскрашены одним цветом, все заведения продающие их по 51 центу, будут раскрашены другим цветом и т.д. Каждому уникальному значению присваивается собственный цвет. Дистрибьютор сможет увидеть распределение цен среди супермаркетов по площади и сможет определить где следует увеличить объем продаж исходя из сложившихся цен.

Если Вы раскрашиваете точки, линии или границы используя номинальные данные, то сможете раскрасить только по отдельным значениям. Номинальные данные или нечисловые данные (имена, типы ресторанных кухонь или марки автомобилей) или числовые данные где числа представляют нечисловые данные (например, идентификаторы). Данные рассматриваются как числовые данные и могут использоваться в картах диапазонов и отдельных значений.

Например, у Вас есть результаты потребительского опроса. Один вопрос был такой “Ваше основное занятие воскресенье вечером?”

Возможные ответы:

1. Сон
2. Просмотр TV
3. Выпивка
4. Чтение
5. Спортивные занятия
6. Посещение музеев и галерей
7. Поход в кино

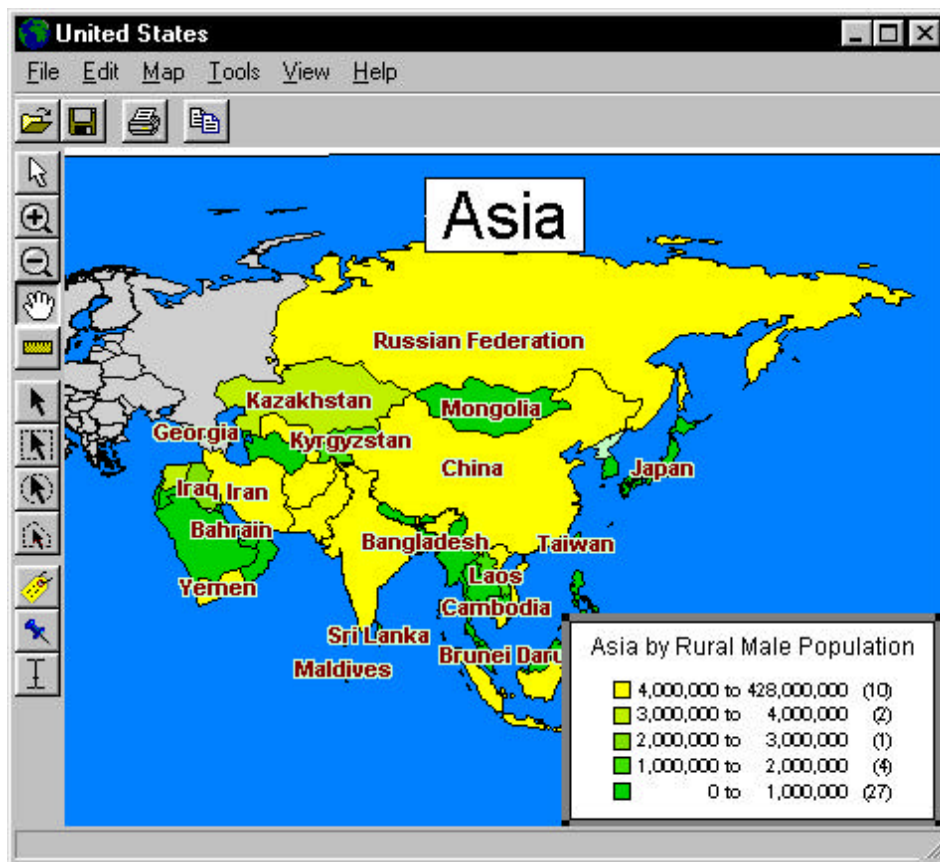
Вам надо раскрасить карту в соответствии с ответами потребителей. Поле SUNDAY Вашего набора данных содержит числа, соответствующие номерам ответов. Таким образом, эти числа в колонке являются скорее индексами а не численными значениями. Поход в кино не больше чем спортивные занятия, хотя $7 > 5$. Когда числа используются вместо значений, надо использовать тематическую карту отдельных значений.

Карта отдельных значений настраивается через коллекцию **IndividualValueCategories**, являющуюся коллекцией объектов **IndividualValueCategory** — один объект для каждого уникального значения. Чтобы получить коллекцию **IndividualValueCategories**, обратитесь к свойству **ThemeProperties.IndividualValueCategories**. Более подробную информацию смотрите в *Электронной справке* или в *Справочнике*.

Работа с тематическими картами

Карты диапазонов

Когда Вы создаете карту диапазонов, MapX группирует все строки набора данных в диапазоны и присваивает каждому объекту строки цвет, символ или линию в соответствии с соответствующим диапазоном. Например, у Вас есть набор данных от станций наблюдения за погодой для визуализации, и Вы хотите раскрасить пункты наблюдений в соответствии с количеством выпавшего снега.



В соответствие со свойствами карт диапазонов, MapX группирует количество выпавшего снега в диапазоны. Все станции наблюдения, где снега от нуля до 5 дюймов, попадают в один диапазон. Станции, где снега от 5 до 10 дюймов, попадают в следующий диапазон. Станции, где снега от 10 до 15 дюймов, попадают в третий диапазон, а там, где снега более 15 дюймов, попадут в четвертый диапазон.

Все записи в наборе данных сортируются по диапазонам и им присваивается цвет данного диапазона. Например, самый последний диапазон, где снега более 15 дюймов закрашивается красным цветом. Другие диапазоны закрашиваются в более светлые цвета, а самый первый диапазон будет серым (цвет по умолчанию). Когда Вы отображаете карту, цвета сразу покажут где сколько снега.

Диапазоны также полезны когда размер полигона прямо не зависит от численного значения тематической переменной.

Типы выделения диапазонов

MapX может создавать диапазоны автоматически используя пять методов: [Равное количество записей](#), [Равный разброс значений](#), [Естественные группы](#), [На базе дисперсии](#) и [Квантили](#). Диапазоны устанавливаются свойством **DistMeth** объекта **ThemeProperties**. Свойство **DistMeth** может быть установлено на любую из следующих констант **DistribMethodConstants**.

DistribMethodConstants

Равное количество записей (**miEqualCountPerRange**) имеет одинаковое количество записей в каждом диапазоне. Если Вы хотите, что бы MapX сгруппировал 100 записей в 4 диапазона, используя равное количество записей, MapX вычисляет диапазоны так, что примерно 25 записей попадут в каждый диапазон, в зависимости от фактора округления.

При использовании равного количества записей (или любого другого метода), важно следить за экстремальными значениями, которые могут влиять на тематическую карту. Например, если Вы раскрашиваете диапазоны по методу равного количества записей:

John	5000	Andrea	7000
Penny	6000	Kyle	5500
Miguel	4500	Angela	7500
Linda	5000	Elroy	6000
Ben	100	Mark	7000

Ben и Miguel группируются в одном диапазоне (поскольку у них наименьшие значения). Это не приведет к очень хорошим результатам, так как не будет видно что у Ben значение на порядок меньше чем у других.

Равный разброс значений (**miEqualRangeSize**) разделяет записи на диапазоны равных значений. Например, у Вас есть поле в таблице с данными в диапазоне от 1 до 100. Вам надо создать тематическую карту с четырьмя равными диапазонами. MapX разделит так 1–25, 26–50, 51–75 и 76–100.

Помните, что MapX может создавать диапазоны без попадающих туда данных, в зависимости от распределения данных. Например, Вы раскрашиваете базу данных в соответствии с равным разбросом значений:

John	100	Andrea	90
Penny	6	Kyle	1
Miguel	4	Angela	92
Linda	95	Elroy	89
Ben	10	Mark	10

MapX создает четыре диапазона (1–25, 26–50, 51–75 и 76–100). Обратите внимание, что только два диапазона имеют реальные записи (1–25 и 76–100).

Естественные группы и Квантили это способы показать данные вне зависимости от их распределения.

Естественные группы (**miNaturalBreak**) создают диапазоны в соответствии с алгоритмом, использующим среднее внутри каждого диапазона. Он распределяет данные так, что среднее в каждом диапазоне наиболее близко к каждому значению в этом диапазоне. Это обеспечивает то, что диапазоны наилучшим образом представлены своими средними значениями и что данные внутри каждого диапазона наиболее близки друг к другу.

Когда Вы создаете диапазоны используя метод **На базе дисперсий (miStandardDeviation)**, средний диапазон разбивает данные по среднему значению и диапазоны выше и ниже среднего диапазона отстоят на единицу стандартного отклонения от среднего.

Вы так же можете определить диапазоны используя метод **Вручную (miCustomRanges)**. Вы указываете собственные диапазоны в коллекции **RangeCategories** (и MapX не будет вычислять диапазоны).

Пример настройки диапазонов вручную

Карты тематических диапазонов устанавливаются через коллекцию **RangeCategories**, являющуюся коллекцией объектов **RangeCategory** — один объект для каждого диапазона, отсортированные в восходящем порядке. Каждый объект в коллекции описывает один диапазон (стиль, которым он изображается, его минимальное и максимальное значение и другое).

```
' Этот пример устанавливает пользовательские диапазоны
для тематического слоя
' базирующегося на значениях в dataset и
контролирующего форму.
```

```

' Тематика с диапазонами была уже создана и сделаны
настройки
' ThemeProperties для этого слоя.
Public Sub GetThemeInfo()
    Dim rc As MapXLib.RangeCategory
    Dim thm As MapXLib.Theme
    Dim iNumRanges As Integer, iCount As Integer
    Dim dIncrement As Double, dCurRange As Double

    Set thm = Form1.Map1.Datasets(1).Themes(1)

    ' Подготовка тематического слоя путем пересчета для
каждого изменения
    ' значения свойства поскольку мы будем устанавливать
численные свойства.
    thm.AutoRecompute = False
    thm.ThemeProperties.DistMethod =
miCustomRanges
    ' Следующая строка получает значение от управляющего
ползунка
    iNumRanges = Form1.sldNumRanges.Value
    thm.ThemeProperties.NumRanges = iNumRanges
    ' Эти настройки объекта rc будут первой
RangeCategory в
    ' коллекции RangeCategories для этого тематического
слоя диапазонов.
    Set rc = thm.ThemeProperties.RangeCategories(1)
    ' Эти настройки цветов для объекта RangeCategory
будут
    ' цветом управления PictureBox
    rc.Style.RegionColor =
Form1.picMinColor.BackColor

```

```

        ' Эти настройки объекта rc будут последним Range
Category в
        ' коллекции RangeCategories для этого тематического
слоя диапазонов.
        Set rc =
thm.ThemeProperties.RangeCategories(iNumRanges)
        ' Эти настройки цвета для объекта RangeCategory
будут
        ' цветом управления PictureBox.
        rc.Style.RegionColor =
Form1.picMaxColor.BackColor
        ' Этот раздел кода создает пользовательские
диапазоны для тематического слоя
        ' последовательно для каждого объекта RangeCategory
        ' устанавливая значения свойств Min и Max.
        dIncrement = Int(Val(Form1.txtmaxvalue.Text) /
iNumRanges) + 1
        dCurRange = 0
        For iCount = 1 To iNumRanges

thm.ThemeProperties.RangeCategories(iCount).Min =
dCurRange
            dCurRange = dCurRange + dIncrement - 1

thm.ThemeProperties.RangeCategories(iCount).Max =
dCurRange
            dCurRange = dCurRange + 1
        Next
        ' Этот блок вернет автоматически пересчитанные
свойства
        ' для объектов легенды и тематического слоя, затем
перерисует карту.
        thm.Legend.LegendTexts.AutoGenerate = True

```

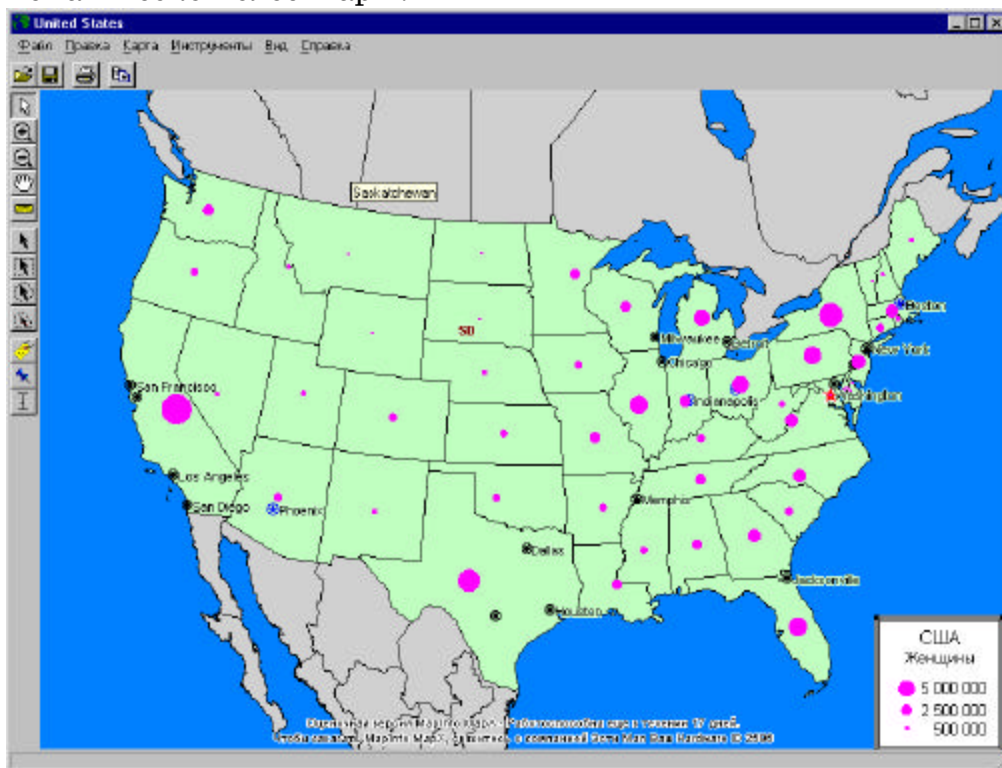
```

thm.AutoRecompute = True
Form1.map1.Refresh
End Sub

```

Карты размерных символов

Карты размерных символов используют символы для представления различных значений. Вы можете использовать размерные символы независимо от типа используемых данных. Используйте размерные символы, например, что бы показать продажи по территории. В зависимости от объема продаж будет изменяться размер символа на тематическом слое MapX.



Или Вы можете представить в виде размерных символов степень потребительского спроса в разных местах, степень интереса будет пропорциональна размеру символа.

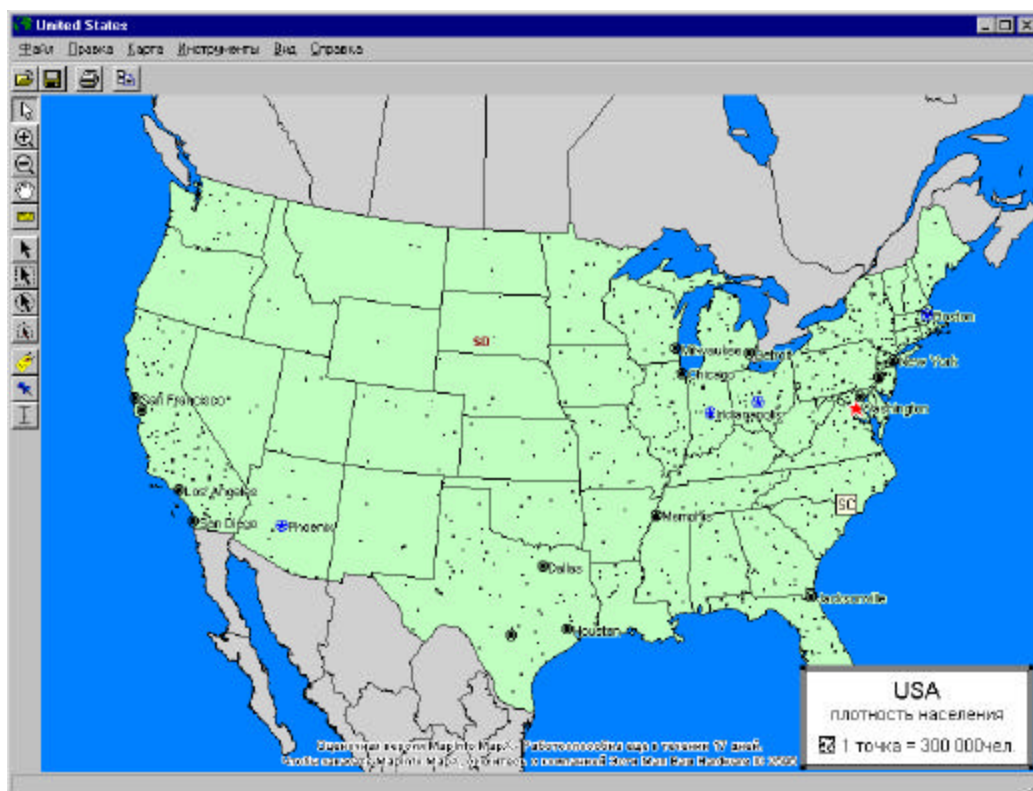
Размерные символы хорошо применять при использовании числовых данных. Если Вы работаете с таблицей ресторанов, нет смысла отражать размерными символами тип кухни в каждом ресторане. Таким образом, размерные символы лучше применять если надо показать количество гамбургеров, проданных в 20 различных ресторанах.

Есть два свойства, настраивающих карту размерных символов: **Data Value** и **SymbolStyle**. Свойство **SymbolStyle** контролирует используемый символ, такое как размер символа, зависящий от свойства **Data Value**. Все значения между наибольшим значением и нулем имеют интерполированное значение размера символа. Свойство **SymbolStyle** это объект **Style**, который контролирует все такие аспекты оформления как цвет, шрифт, вращение и др.. Можно применить собственный растр **bitmap** в качестве символа.

Объект **Style** имеет возможность выбора символа (**Style.PickSymbol**) при чем открывается диалог **Стиль Символа**. В диалоге выбираются свойства стиля символа. Объект **Style** обновляется в соответствии со сделанными настройками при нажатии кнопки **ОК** в диалоге.

Карты плотности точек

Карты плотности точек используют точки для представления значений данных привязанных к границам (замкнутым) или полигонам. Общее число точек в соответствует численному значению тематической переменной, относящийся к данному полигону. Если у Вас есть 10000 пожилых людей на страну, и каждая точка представляет 100 человек, то здесь будет 100 точек внутри границы страны.



Плотность точек обычно полезна для показа таких данных у которых одна точка представляет большое количество таких данных: население, число точек общепита, число дистрибьютеров и др.

Например, если у Вас есть таблица с данными о населении и границами районов, можете использовать плотность точек для отображения плотности точек в каждом районе. Есть два тематических свойства для контроля слоев с плотностью точек. Вы можете указать величину соответствующую одной точке. Например, есть таблица со статистикой населения, разделенной на границы. В ней есть 20000 студентов в графстве Rensselaer, New York. Если Вы раскрашиваете графство Rensselaer в соответствии с числом студентов, используя метод плотности точек, каждая точка будет представлять 200 студентов. В этом случае, в графстве Rensselaer будет 100 точек.

Когда Вы увеличиваете число значений соответствующих каждой точке, Вы уменьшаете число самих точек. Можно изменить настройки так, что одна точка будет представлять 400 студентов. В этом случае будет только 50 точек в графстве Rensselaer.

```
Map1.Datasets(1).Themes("My DotDensity _  
Theme").ThemeProperties.ValuePerDot = 400
```

Вторая настройка используется для изменения размера точек. Если Вы работаете с большим количеством значений которые порождают много точек, то задайте точки малыми, чтобы лучше их видеть. Если данные невелики и точек мало, можно увеличить их размер.

```
Map1.Datasets(1).Themes("My DotDensity _  
Theme").ThemeProperties.DotSize = miDotSizeLarge
```

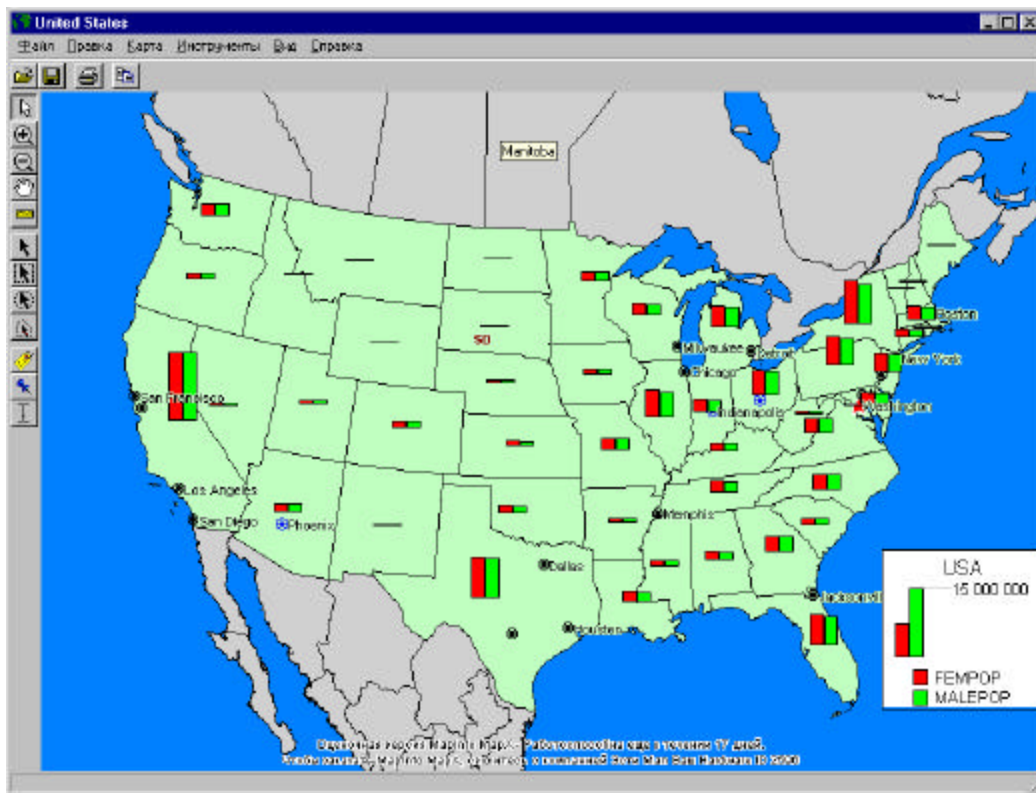
Внимание: Распределение точек случайно в пределах области.

Например, если Вы раскрашиваете штаты в соответствии с количеством населения, точки расположатся равномерно по штату Нью-Йорк, а не в районе города Нью-Йорк, где на самом деле проживает большинство.

Карты со столбчатыми диаграммами

В отличие от тематических карт с одной переменной, таких как диапазоны и размерные символы, карты столбчатых диаграмм позволяют анализировать и отображать несколько переменных одновременно. Столбчатая диаграмма строится для каждого объекта и привязывается к его центроиду, на ней изображаются столбцами разные переменные, что позволяет сравнивать их друг с другом по их высоте. Можно также анализировать те же переменные по разным объектам карты.

Например, у Вас есть таблица границ штатов U.S., содержащая сведения о женском и мужском населении. Используя столбчатые диаграммы, можно создать тематическую карту со столбчатыми диаграммами имеющими по два столбца для каждого штата: один столбец для мужчин и один для женщин. Вы можете сравнивать разницу в половом составе населения для каждого штата или можете сравнивать различия по разным штатам. Для лучших результатов, используйте не более четырех столбцов для одной диаграммы.



Вы можете настраивать пять параметров для столбчатых диаграмм: **DataValue**, **Size**, **Independent**, **MultiVarCategories** и **Width**.

Свойство **DataValue** работает в сопряжении со свойством **Size** для контроля размера тематических столбцов. По умолчанию значение этого свойства устанавливается на наибольшее значение из данных.

Свойство **Size** работает в сопряжении со свойством **DataValue** для контроля размера тематических диаграмм. Оно указывает высоту столбцов в “бумажных” единицах (PaperUnit). Имеет двойную точность и по умолчанию равно .25 дюймов.

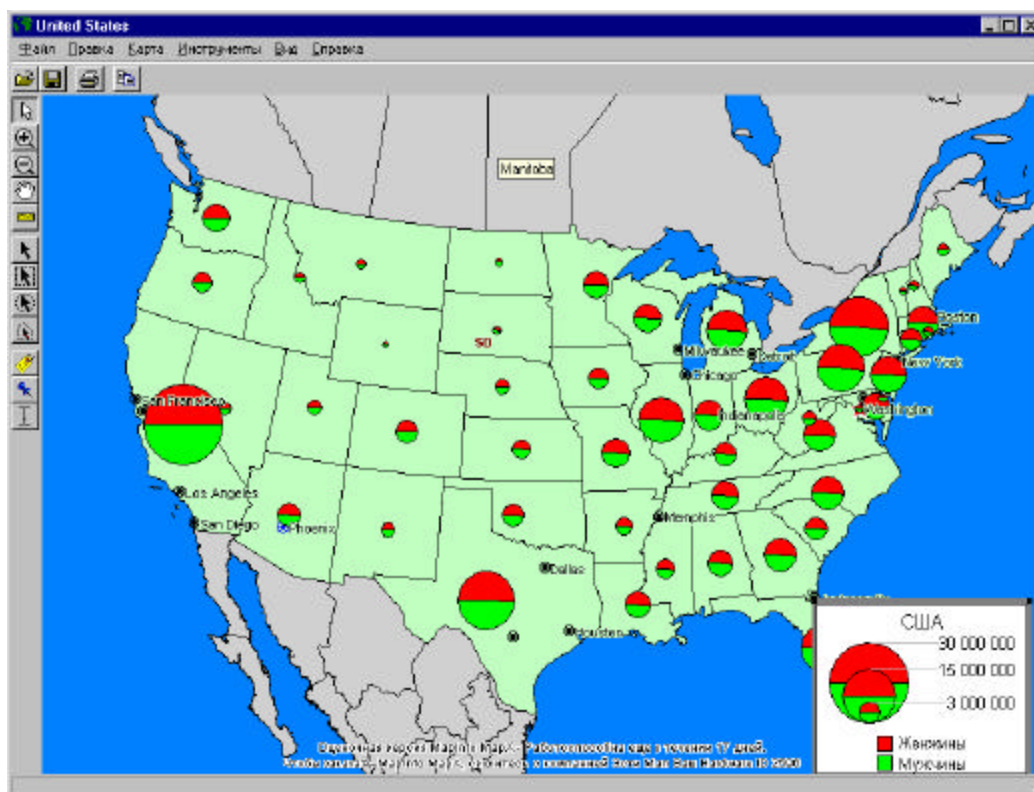
Свойство **Width** указывает ширину каждого столбца в диаграмме в “бумажных” единицах. Это величина двойной точности и по умолчанию равна .25 дюйма.

Свойство **Independent** контролирует, являются ли данные для столбцов независимыми (не сравнимыми, в смысле единиц измерения, такими как Population и Avg Income). Это логическая величина и по умолчанию равна False. Разработчик должен установить это свойство на True для столбчатых диаграмм с несколькими столбцами, в который отдельные столбцы соответствуют независимым полям, с разными единицами измерений, или если единицы одни но различие сильно велико. Примером может быть столбчатая диаграмма, у которой один столбец представляет население штата а другой показатели экспорта. Население будет достигать миллионов, а позиция в экспорте от 1 до 50. Если свойство **Independent** установлено на True, наивысшее значение населения по высоте столбца будет равно наивысшей высоте столбца с максимальной позицией экспорта.

Тематическая карта столбчатых диаграмм выставлена в OLE через коллекцию **MultivarCategories**, которая является коллекцией объектов **MultivarCategory**. Коллекция содержит один объект для каждого столбца в столбчатой диаграмме. Для более подробной информации о коллекции **MultivarCategories**, смотрите *Электронную Справку* или *Справочник*.

Тематические карты круговых диаграмм

Тематические карты с круговыми диаграммами также позволяют изображать по несколько переменных. Подобно высоте столбцов в столбчатых диаграммах, в круговых диаграммах Вы сравниваете сектора круга. Круговые диаграммы также позволяют Вам сравнивать части целого.



И круговые и столбчатые диаграммы обычно используются для анализа демографических данных. Например, есть набор данных для США. Вышли наборы данных показывают население по нескольким группам. Используя круговые диаграммы, Вы можете показать население каждой демографической группы и увидеть, какую часть от всего населения она составляет. Такой подход позволит увидеть распределение демографических групп и для отдельных штатов и для всей страны. Вы можете проанализировать как меняется от штата к штату вклад конкретной демографической группы. Для лучших результатов, используйте не более 4-6 тематических переменных в круговой диаграмме.

Существует четыре тематических свойства, контролирующих столбчатые диаграммы: **DataValue**, **Size**, **Graduated** и **MultiVarCategories**.

Свойство **DataValue** работает в соединении со свойством **Size** и контролирует зависимость размера тематических графиков от значений данных. По умолчанию значение для этого свойства устанавливается равным наибольшему значению из тематических данных.

Свойство **Size** работает в соединении со свойством **DataValue** для контроля того какой размер будет у тематических графиков. Это свойство указывает высоту тематического графика в “бумажных” единицах (PaperUnit). Это величина двойной точности, по умолчанию равная .25.

Свойство **Graduated** контролирует является ли размер круговой диаграммы зависимым от суммарного значения. Это логическая величина и по умолчанию равна True.

Тематическая карта круговых диаграмм выставлена в OLE через коллекцию **MultivarCategories**, которая является коллекцией объектов **MultivarCategory**. Коллекция содержит один объект для каждого столбца в столбчатой диаграмме. Для более подробной информации о коллекции **MultivarCategories**, смотрите *Электронную Справку* или *Справочник*.

Бивариантная тематическая картография

Бивариантная тематическая картография использует точечные или линейные объекты для отображения двух тематических переменных. Например, звезда может представлять одну переменную, такую как число подростков, а синяя закраска представляет годовые суммы их покупок. Для создания бивариантных карт в MapX, Вы создаете две тематических карты и слой над ними, на котором и отобразятся объекты этих двух переменных.

Типы карт и переменных

Для бивариантных карт подходят только карты диапазонов и индивидуальных значений. Вы можете выбрать между двумя комбинациями для бивариантных карт, в зависимости от данных:

- две карты диапазонов
- одна карта диапазонов и одна карта индивидуальных значений

Если у Вас нечисловые переменные, одна из Ваших карт должна быть картой индивидуальных значений. Вы не можете создавать бивариантную карту с двумя нечисловыми переменными.

Отображение атрибутов

Для отображения двух переменных в одном символе, важно выбрать различные атрибуты символа для каждой переменной. Например, Вы не можете выбрать цвет для обеих переменных, потому что один цвет перекроет другой. Выберите из следующих комбинаций:

- цвет и тип символа
- цвет и размер
- размер и тип символа

Тип символа должен использоваться для нечисловых данных, поскольку нет связи между типом символа и количеством чего-либо.

Работа с тематической картой

Все свойства тематических слоев могут быть скорректированы во время работы приложения. Есть два пути достичь этого:

- Метод `Theme.ThemeDlg`.
- Объект `ThemeProperties` тематического слоя.

Метод `ThemeDlg`

Тематический объект имеет метод, `ThemeDlg`, который открывает диалог, позволяющий пользователю изменять тематику. Следующая строка кода показывает как это просто сделать.

```
Map1.Datasets(1).Themes(1).ThemeDlg
```

Вы получите диалог с возможностью настройки параметров в нем для того типа тематического слоя, из которого он вызван.

Есть две возможности получить ошибку при использовании этого метода. Первое, Вы ошибаетесь с оформлением. Если эти диалоги не заполняют в соответствии с выбранными Вами заливками или Вас не устраивает, как получилась тематическая раскраска, поменять уже не удастся. Другая проблема в использовании этих методов в том, что они иногда слишком мощные. Возможно пользователю нужно только поменять цвет верхнего диапазона карты диапазонов или надо сохранить цвета но не позволить пользователю менять число диапазонов. Используя метод `ThemeDlg`, пользователь может менять любую часть или все в тематическом слое.

Объект ThemeProperties

У Вас также есть возможность поменять необходимые свойства тематического слоя при работе с объектом **ThemeProperties**. Это легко сделать и в результате увеличится контроль в руках пользователя над тем что ему позволить и чего не позволять в изменении настроек.

Объект **ThemeProperties** хранится в коллекции **Themes**. Свойства объекта **ThemeProperties** используются для определения атрибутов оформления тематического слоя (цвета, символы и др.).

Объект **ThemeProperties** действительно содержит информацию о том, как должен выглядеть тематический слой. Некоторые из свойств объектов **ThemeProperties** представлены другими объектами.

Свойство	Описание
AllowEmptyRanges	Контролирует, является ли допустимыми пустые диапазоны в тематическом слое диапазонов.
DataValue	Применяется к размерным символам, круговым и столбчатым диаграммам. Это величина, учитываемая при создании тематического слоя и она сопрягается с размером, задаваемым в свойстве Size.
DistMethod	Контролирует, как создаются диапазоны когда заново вычисляется объект Theme. Это величина DistribMethodConstants и по умолчанию она равна MiEqualCountPerRange. Другие константы метода это miEqualRangeSize, MiCustomRanges, miNaturalBreakRange, miStandardDeviation.
DotSize	Контролирует размер точки, используемый в тематических слоях плотности точек.
Graduated	Контролирует, является ли зависимым размер круговой диаграммы от суммарного значения тематических величин.
IndividualValueCategory	Коллекция IndividualValueCategory.

Independent	Контролирует являются ли независимыми значения для отдельных столбцов диаграммы (например, численность населения и сумма средних налогов - разные единицы измерений).
MultivarCategories	Есть один объект MultiVarcategory для картируемого поля или переменной в круговой или столбчатой диаграмме.
NumRanges	Контролирует число диапазонов для тематической карты диапазонов.
RangeCategories	Тематическая карта диапазонов имеет коллекцию объектов RangeCategory.
Size	Работает совместно со свойством DataValue для контроля размера тематических графиков в слоях круговых и столбчатых диаграмм.
SpreadBy	Контролирует как осуществлено автоматическое разбиение на интервалы.
SymbolStyle	Объект Style контролирующий использование символов для тематического слоя размерных символов.
ValuePerDot	Это свойство применяется к тематическому слою плотности точек. Оно указывает какому количеству данных соответствует одна точка.
Width	Указывает ширину всех столбцов в тематическом слое столбчатых диаграмм.

Несколько свойств из перечисленных выше действуют и для других объектов. Такие объекты включают объект **RangeCategory**, объект **IndividualValue**, объект **MultiVar** и объект **Style**. Взгляните на следующую таблицу чтобы увидеть свойства объекта **RangeCategory**. Другие объекты, относящиеся к тематике, ведут себя сходным образом.

Свойство	Описание	Пример кода
----------	----------	-------------

Max	Устанавливает максимальное значение для диапазона в слое тематических диапазонов.	<code>Map1.Datasets(1).Themes(1).Properties.RangeCategories(3).Max = 625</code>
Min	Устанавливает минимальное значение для диапазона в слое тематических диапазонов.	<code>Map1.DataSets(1).Themes(1).Properties.RangeCategories(3).Min = 595</code>
NumItems	Показывает число объектов в диапазоне.	<code>Print Map1.DataSets(1).Themes(1).Properties.RangeCategories(1).NumItems</code>
Style	Объект стиля представляющий стиль данного диапазона.	<code>Map1.Datasets(1).Themes.Item(1).Properties.RangeCategories.Item(1).Style.PickRegion</code>

Тематическая карта диапазонов имеет коллекцию объектов **RangeCategory**, один для каждого диапазона, хранящуюся в восходящем порядке. Свойство **ThemeProperties.RangeCategories** хранит коллекцию.

Если **ThemeProperties.DistMethod** это **MiCustomRanges**, MapX будет предполагать что у Вас есть набор собственных значений для диапазонов и они будут использоваться для вычислений. Если диапазоны перекрываются пре перевычислении, то будет генерироваться сообщение об ошибке.

Настройка тематической легенды

Когда Вы создаете тематическую карту, MapX автоматически создает легенду, объясняющую цвета, символы или размеры их на карте.

Объект Legend

Каждый тематический слой имеет объект **Legend (Theme.Legend)**.

Объект **Legend** содержит свойства для контроля изображения легенды.

Легенду легко изменить используя метод **LegendDlg** из объекта

Legend. Смотрите следующий пример и диалог, создаваемый в нем.

```
Map1.Datasets(1).Themes(1).Legend.LegendDlg
```

13

Использование систем координат

- ◆ Использование систем координат
- ◆ Концепции систем координат
- ◆ Получение объекта Coordinate System
- ◆ Запрос свойств объекта CoordSys
- ◆ Отображение Карты в разных системах координат
- ◆ Определение координат X-Y в различных системах координат
- ◆ Диалог Выбор Проекции
- ◆ Использование настроек из MAPINFOW.PRJ
- ◆ Применение аффинных преобразований
- ◆ Определение собственных регионов
- ◆ Преобразование региона
- ◆ Дополнительная информация.....
- ◆ Преобразование регионов
- ◆ Дополнительная информация...

В этой главе мы покажем как использовать системы координат (также называемыми "картографическими проекциями") для изменения вида карты или изменения единиц в которых координаты карты обрабатываются MapX.

Концепции систем координат

Каждая карта имеет систему координат, которая используется геоинформационными системами различными способами:

Система координат действует так, что преобразуются координаты X-Y некоторым образом. Например, местоположение объекта можно выразить в координатах широта-долгота а можно и в других единицах, например, в метрах.

Система координат действует на изображение карты. Изменение системы координат карты может привести к изменению - искажению, растягиванию и др., карты. Система координат представляет комплексную, продвинутую область компьютерной картографии. Эта дискуссия предполагает, что Вы уже знакомы с концепцией картографии, такими понятиями как восточное смещение (false eastings) и проекции карт.

Получение объекта **Coordinate System**

Модель объектов MapX представляет информацию о системе координат посредством объекта **CoordSys**. Свойства этого объекта сообщают все что Вам может понадобиться знать о системе координат.

Существует несколько путей получения объекта **CoordSys**. Какой путь Вы будете использовать, зависит от того, что Вы пытаетесь выполнить.

Свойство **Map.DisplayCoordSys** возвращает объект **CoordSys** доступный для чтения-записи. Вы можете использовать этот объект для контроля за системой координат и проекцией карты, чтобы отобразить ее на экране. Для более подробной информации, смотрите раздел *Отображение карты в различных системах координат* который находится ниже в этой главе.

Свойство **Map.NumericCoordSys** также возвращает объект **CoordSys** открытый для чтения-записи. Используйте этот объект для установки системы координат, которую MapX использует для преобразования координат X-Y. Более подробно написано в разделе *Определение координат X-Y в различных системах координат*.

Свойство **Layer.CoordSys** возвращает объект **CoordSys** открытый только для чтения, который сообщает Вам систему координат координат используемую для хранения объектов в указанном слое карты. Если Вы устанавливаете **Map.DisplayCoordSys** так, чтобы было совпадение с **Layer.CoordSys**, Вы максимизируете скорость прорисовки этого слоя на экране. (Когда **DisplayCoordSys** отличается от **CoordSys** в которой слой был сохранен, MapX конвертирует координаты, что приводит к задержке прорисовки по времени.)

Запрос свойств объекта **CoordSys**

Для каждого типа систем координат только некоторые свойства могут быть применены. Например, если свойство **CoordSys.Type** это **miRobinson** (12), то применимы только свойства **Datum**, **Units** и **OriginLongitude**. Проекция карты Robinson просто не использует другие свойства.

Чтобы узнать, какие свойства используются для каких проекций, смотрите раздел *Параметры используемые для Систем Координат*.

Вы можете запросить все возможные свойства для любого объекта **CoordSys**— даже такие свойства, которые не применяются для данной конкретной системы координат. Если свойство не применяется для данного типа **CoordSys**, MapX вернет значение по умолчанию. Если Вам надо сохранить объект **CoordSys**, Вы можете сделать это посредством запроса и сохранения всех его свойств, не беспокоясь о том, применимо ли здесь каждое из свойств.

Отображение Карты в разных системах координат

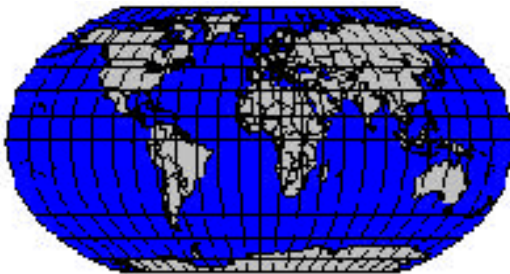
Есть два способа, которыми приложение может задать систему координат (или "проекцию") в которой отображается карта:

- Использование метода `Map.DisplayCoordSys.Set` для установки новой системы координат `CoordSys` в которой будет изображена карта.
- или-
- Включение нового объекта `CoordSys` в свойство `Map.DisplayCoordSys`.

Внимание: Если карта включает слой с растровым изображением, MapX автоматически отображает карту в той же проекции, в которой зарегистрирован растр. В этом случае, Вы не можете изменять систему координат.

Пример: Изображение карты в системе координат Робинсона

Следующий пример Visual Basic изменяет `DisplayCoordSys` таким образом, что карта изобразится в проекции Робинсона, и будет выглядеть так:



```
Dim iProjectionType As Integer
Dim iDatumNumber As Integer
Dim iUnits As Integer
Dim dOriginLongitude As Double
```

Инициализация переменных, которые будут использоваться `coordsys.set`.

```
' Значения были получены из ввода "Robinson"
' in the file MAPINFOW.PRJ, which looks like this:
```

```
' "Robinson", 12, 62, 7, 0
iProjectionType = miRobinson '(value: 12)
iDatumNumber = 62           'North American 1927
(NAD 27)
iUnits = miUnitMeter        '(value: 7)
dOriginLongitude = 0
Map1.DisplayCoordSys.Set iProjectionType,
iDatumNumber, _
iUnits, dOriginLongitude
```

Определение координат X-Y в разных системах координат

Объект Map имеет свойство **NumericCoordSys**, которое представляет систему координат используемую для преобразования числовых координат. Это свойство действует и на ввод и на вывод координат X-Y следующим образом:

- Когда Вы определяете координаты карты (как свойства объекта или параметры метода), MapX предполагает, что координаты уже в системе координат.
- Когда MapX возвращает координаты карты (как события или свойства объекта), координаты пересчитаны в какую-либо систему координат.

По умолчанию, система координат это Широта/Долгота WGS-84. Другими словами, если Вы не изменяете **NumericCoordSys**, MapX предполагает, что координаты карты это градусы Широта/Долгота.

Есть два способа которыми приложение может указать систему координат, используемую для преобразования исходных числовых координат:

- Использование метода **Map.NumericCoordSys.Set** для установки новых параметров системы координат **CoordSys**.
- или-
- Поместить новый объект **CoordSys** в свойство **Map.NumericCoordSys**.

Пример: Возвращение координат в разных системах координат

Следующий пример Visual Basic изменяет **NumericCoordSys** карты так, что координаты возвращаются в другой системе координат.

```
Dim iProjectionType As Integer
Dim iDatumNumber As Integer
```

```

Dim iUnits As Integer
Dim dOriginLongitude As Double

' Display map's center X-Y in original coordsys
Debug.Print "Original Center: " & Map1.CenterX &
", " & _
Map1.CenterY

' Initialize the variables to be used by
coordsys.set.
' Values were obtained from the "Robinson" entry
' in the file MAPINFOW.PRJ, which looks like this:
' "Robinson", 12, 62, 7, 0

iProjectionType = miRobinson '(value: 12)
iDatumNumber = 62 '(North American 1927
(NAD 27)
iUnits = miUnitMeter '(value: 7)
dOriginLongitude = 0

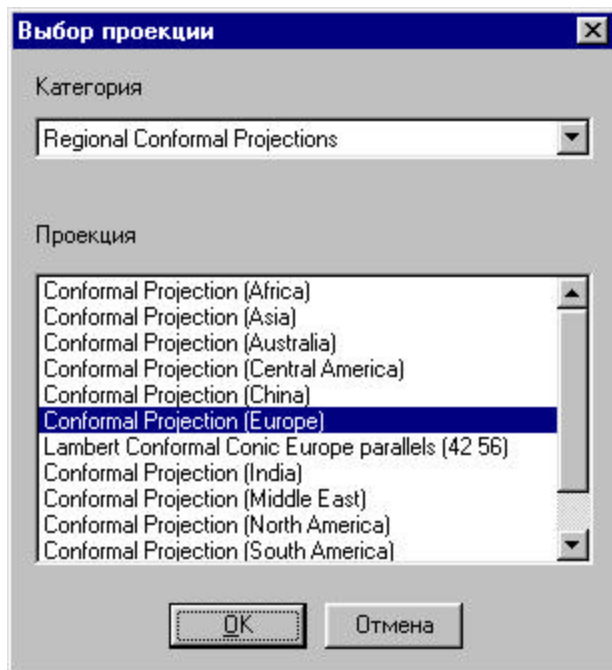
Map1.NumericCoordSys.Set iProjectionType,
iDatumNumber, _
iUnits, dOriginLongitude

' Display map's center X-Y in the new coordsys
Debug.Print "New Center: " & Map1.CenterX & ", " &
Map1.CenterY

```

Диалог Выбор Проекции

Что бы открыть диалог, позволяющий пользователю выбрать систему координат, используйте метод `CoordSys.PickCoordSys`.



Диалог инициализируется автоматически, так что текущая система координат уже выделена. Если пользователь нажмет ОК в диалоге, метод вернет **True**, и объект **CoordSys** обновляется в соответствии с той системой координат, которую пользователь выбрал в диалоге.

Что бы открыть диалог:

1. Щелкните правой кнопкой мышки на **Map**.
2. Выберите **Свойства** из быстрого меню.
3. В закладке **Главная**, нажмите кнопку **Проекция**.

MapX откроет такой же диалог **Выбор проекции**, какой вызывается методом **PickCoordSys**.

Пример

Следующий оператор открывает диалог **Выбор проекции** в процессе работы программы, позволяет пользователю менять систему координат в которой изображена карта.

```
Map1.DisplayCoordSys.PickCoordSys
```

Использование настроек из MAPINFOW.PRJ

Если Вы используете MapInfo Professional, Вы найдете файл с именем MAPINFOW.PRJ в той же директории, где и программы MapInfo. MAPINFOW.PRJ это текстовый файл, содержащий параметры, которые определяют систему координат MapInfo. Копия файла MAPINFOW.PRJ включена в поставку с MapX.

Предположим, Вы используете указанную проекцию в MapInfo— проекцию Робинсона, например— и Вы хотите определить эту же проекцию в приложении MapX. Файл MAPINFOW.PRJ дает Вам информацию, которая нужна для заполнения необходимых параметров в методах MapX, таких как **CoordSys.Set**.

Содержимое файла MAPINFOW.PRJ имеет следующие параметры. Параметры разделены запятыми.

Часть	Описание
Description	Строка, описывающая систему координат. Это описание появляется в диалоге.
Type	Число, точно совпадающее с одной из констант CoordSysTypeConstants. Пользователи могут настроить это число двумя способами: Указать аффинное преобразование, для этого надо добавить

	1000 к этому числу; указать прямоугольник, для этого надо добавить 2000 к этому числу. Подробнее смотрите ниже.
Datum	Число, представляющее один из датумов, поддерживаемых MapX, или специальный номер (999 или 9999) указывающий на пользовательский регион.
Units [Set of 4 Custom Datum Arguments]	Добавочная; применяется только если номер региона это 999.
[Set of 9 Custom Datum Arguments]	Добавочная; применяется только если номер региона это 9999.
Set of Projection Parameters	От 0 до 8 параметров, в зависимости от того, какой тип указан. В электронной Справке указано какие параметры в каких системах координат используются.
[Set of 7 Affine Transformation Parameters]	Добавочная; применяется только если номер Type это точно 1000 или 3000, больше чем константа CoordSysTypeConstant.
[Set of 4 Bounds Parameters]	Добавочная; применяется только если номер Type это точно 2000 или 3000, больше чем константа CoordSysTypeConstant.

Пример

Откройте MAPINFOW.PRJ в текстовом редакторе.

Внимание: MapX использует этот файл при интерпретации и установке системы координат карты. Сделайте резервную копию этого файла прежде чем вносить изменения.

Внутри MAPINFOW.PRJ, найдите имя проекции MapInfo а именно "Robinson". Строка, в которой Вы это нашли это имя содержит параметры проекции Робинсона:

`"Robinson", 12, 62, 7, 0`

Эта строка сообщает кое-что о проекции Робинсона:

- В начале строки помещено название. Первое число после описания, 12 в этом примере, сообщает, какой тип системы координат применяется. Проверьте по таблице номер 12 среди констант `MapX CoordSysType` и увидите, что тип 12 этих констант соответствует `miRobinson`. (В этом случае, описание проекции делает ее очевидной, то есть что Вы используете проекцию Робинсона; но в других случаях это может быть и не так.)
- Проекция Робинсона использует только три параметра: регион, единицы измерения и опорная долгота. Теперь Вы знаете как интерпретировать остальные числа в строке: 62 это номер региона, 7 это номер единиц измерения и 0 это опорная долгота. Из списка регионов, поддерживаемых `MapX`, Вы можете определить, что регион 62 представляет NAD 27 для континентальной части США.
- Из таблицы констант `MapUnit`, Вы можете определить, что тип номер 7 соответствует метрам, `miUnitMeter`.

Вы можете использовать настройки этих значений (12, 62, 7, 0) как параметры, когда вызываете метод `CoordSys.Set`. Например:

```
objCoordSys.Set 12, 62, 7, 0
```

Как интерпретировать специальные номера регионов

Пользователи `MapInfo` могут изменять файл `MAPINFOW.PRJ`, определяя собственные регионы. Есть два типа собственных определений регионов:

- Полностью определенный пользователем свой регион состоит из числа 9999, следующим за девятью параметрами системы координат, разделенных запятыми.
- Упрощенно определенный регион состоит из номера 999, следующим за четырьмя параметрами региона, разделенными запятыми.

Если строка в файле MAPINFOW.PRJ указывает номер региона 999 или 9999, то параметры пользовательского региона появляются перед остатком параметров системы координат (единицы, опорная долгота и др.). Более подробная информация об этом в конце главы.

Как интерпретировать номера специальных типов систем координат

Номер типа системы координат, появляющийся в MAPINFOW.PRJ (такой как 12, в примере выше) может не совпадать ни с одним из типов констант MapX **CoordSysType**. Вам может понадобиться вычесть 1000, 2000 или 3000 из числа для получения числа которое сможет представить полноценное значение для MapX **CoordSysType**.

Если строки из MAPINFOW.PRJ содержит аффинное преобразование, то к номер системы координат будет добавлена 1000, и будут добавлены 7 параметров аффинных преобразований к концу строки.

Если строка включает определение прямоугольника, в котором определена данная система координат, то к номеру типа проекции добавится 2000 и четыре параметра, описывающих этот прямоугольник будут добавлены в конец строки.

Например, следующая строка из гипотетического файла MAPINFOW.PRJ представляет систему координат и с аффинными преобразованиями и с определением прямоугольника, в котором эта система координат действует. Эта система координат использует UTM Zone 10 (тип системы координат: **miTransverseMercator**, имеющей номер 8). Таким образом, поскольку система координат включает и аффинные преобразования и описание прямоугольника в котором проекция действует, то число описания будет 3008 (8 + 1000 + 2000).

```
"DCS", 3008, 74, 7, -123, 0, 0.9996, 500000, 0, 3,  
1.57, -0.21, _  
84120.5, 0.19, 2.81, -20318.0, 70000, 0, 80000, 50000
```

Хотя число 3008 подходит и для файла MAPINFOW.PRJ, Вы должны использовать неизмененное число, 8, в MapX.

Приведенный ваш пример "DCS" может быть интерпретирован следующим образом:

```
name,  
CoordSysType + 3000,  
the projection parameters (Transverse Mercator uses 7  
—  
parameters),  
the 7 affine transformation parameters,  
The 4 bounds parameters (in the order: xmin, ymin,  
xmax, ymax)
```

Применение аффинных преобразований

Если Вам надо определить вращение или сдвиг систем координат, то надо задать аффинное преобразование. В MapX, можно определить эти преобразования при создании объекта **AffineTransform**, затем передать объект в метод **CoordSys.Set**.

Внимание: Если Вы просто хотите повернуть карту, не надо делать какие-либо аффинные преобразования. Вместо этого, установите в объекте Map свойство **Rotation**.

Аффинное преобразование имеет следующую форму:

$$x' = Ax + By + C$$

$$y' = Dx + Ey + F$$

В этих уравнениях, базовые координаты (x, y) трансформируются так, что получаются производные координаты (x', y'). Шесть констант от A до F определяют всю трансформацию:

A	Масштабирование или растяжение по оси X.
B	Вращение или перемещение вдоль оси X.
C	Сдвиг вдоль оси X.
D	Масштабирование или растяжение по оси Y.
E	Вращение или перемещение вдоль оси Y.
F	Сдвиг вдоль оси Y.

Пример

Предположим, что надо определить систему координат (назовем ее DCS) которая получена из другой системы координат UTM Зона 10 :

$$x' = 1.57x - 0.21y + 84120.5$$

$$y' = 0.19x + 2.81y - 20318.0$$

В этой трансформации, (x', y') представляют координаты в DCS (десятичные градусы), и (x, y) представляют координаты, базирующиеся на проекции UTM-10. Следующий пример программного кода на Visual Basic показывает как происходят аффинные преобразования в MapX:

```
Dim objAffine As New MapXLib.AffineTransform

' Declare variables to hold coordinate system
information
Dim iProjectionType As Integer
Dim iDatumNumber As Integer
Dim iUnits As Integer
Dim dOriginLongitude As Double
Dim dOriginLatitude As Double
Dim dScaleFactor As Double

Dim lFalseEasting As Long
Dim lFalseNorthing As Long

' initialize the AffineTransform object, using
' the numbers from the equation above
objAffine.Set 7, 1.57, -0.21, 84120.5, _
              0.19, 2.81, -20318.0

' Initialize the variables to be used by
coordsys.set.
' Values were obtained from the "UTM Zone 10 (NAD
83)"
' entry in the file MAPINFOW.PRJ, which looks like
this:
```

```

        ' "UTM Zone 10 (NAD 83)\p26910", 8, 74, 7, -123,
0, 0.9996,
        500000, 0

        iProjectionType = miTransverseMercator '(value:
8)
        iDatumNumber = 74          'North American 1983
(NAD 83)
        iUnits = miUnitMeter      '(value: 7)
        dOriginLongitude = -123
        dOriginLatitude = 0
        dScaleFactor = 0.9996
        lFalseEasting = 500000
        lFalseNorthing = 0

        Map1.DisplayCoordSys.Set iProjectionType,
iDatumNumber, iUnits, _
        dOriginLongitude, dOriginLatitude, ,
        , , _
        dScaleFactor, lFalseEasting,
lFalseNorthing, _
        , , objAffine

```

Определение собственных картографических регионов

Большинство систем координат используют один из стандартных регионов, поддерживаемых MapX. Если нужен регион, которого нет в этом списке, и Вы знаете его математические параметры, то можно определить систему координат, используя собственный картографический регион.

Это обсуждение сопровождается примерами и для MapX и для MapInfo Professional, для тех пользователей, которые используют оба продукта. Если Вы уже создавали свой картографический регион в MapInfo, то Вам будет легко адаптировать регион к MapX. Если Вы не используете MapInfo Professional, можно просто игнорировать пример для MapInfo.

Что такое картографический регион?

Регион это математическое описание формы Земли и ориентации в определенных местах. Поскольку форма Земли не везде одинакова, то в различных частях Земли используются разные коэффициенты для учета этих особенностей регионов. Эти локальные математические описания (назовем их регионами) обеспечивают наиболее точное приближение к форме Земли в конкретных областях Земли.

Каждая система координат использует использует специфическое математическое описание формы Земли (регион) для аппроксимации земной поверхности. Если две системы координат используют разные регионы, то геоинформационная система осуществляет трансформацию регионов когда она трансформирует координаты из одной системы в другую. MapX и MapInfo используют метод трансформации регионов Bursa-Wolfe, который имеет среднюю точность 10 метров. (Когда MapInfo осуществляет конвертацию между двумя системами координат, использующими одинаковый регион, то трансформации регионов не происходит и точность будет в пределах 0.1 метра.)

Параметры, определяющие картографический регион

MapX и MapInfo Professional используют следующую информацию для определения региона:

- Эллипсоид, называемый сфероидом. Это эллипс, вращаемый вокруг короткой оси, и образующий при этом трехмерную поверхность. Эллипсоид описывается двумя математическими параметрами: длиной, в метрах, его главной оси (обозначаемой буквой *a*) и ее степенью уплощения (обозначаемой буквой *f*). Поддерживается около 40 предопределенных эллипсоидов; смотрите раздел Эллипсоиды, поддерживаемые MapX.
- Три параметра сдвига определяют расстояние, в метрах, сдвига эллипсоида вдоль каждой из его осей. Эти параметры обычно обозначаются как *dX*, *dY* и *dZ*. Вы можете также встретить обозначения *u*, *v* и *w*. Метод MapX **Datum.Set** определяет эти параметра так: *ShiftX*, *ShiftY* и *ShiftZ*.

Три параметра вращения указывают угол, в единицах дуги, на который осуществляется поворот каждой из трех осей. Эти параметры обычно обозначаются как *EX*, *EY* и *EZ*. Метод **Datum.Set** определяет эти параметры как *RotateX*, *RotateY* и *RotateZ*.

- Фактор коррекции масштаба указывает величину, в долях миллиона, которая выправляет размер эллипсоида. Этот параметр обозначается буквой *m*.
- Долгота начального меридиана, в градусах к востоку от Гринвича. Начальный меридиан, указывает, какому месту присваивается значение долготы, равное 0°. Большинство регионов используют Гринвичский меридиан в качестве начального, так что этот параметр обычно равен нулю. Таким образом, некоторые регионы используют разные начальные меридианы. Например, регион NTF использует Париж как начальный меридиан, имеющий значение 2.33722917 градусов к востоку от Гринвича. Если Вы используете регион NTF в Вашей системе координат, все значения долготы отсчитываются от Парижа а не от Гринвича.

Определение картографического региона в MapInfo Professional

В MapInfo Вы определяете свой регион в системе координат используя номер 9999 и следующие за ним параметры, в таком порядке:

`9999, EllipsoidNumber, dX, dY, dZ, EX, EY, EZ, m, PrimeMeridian`

Некоторые регионы определяются только параметрами эллипсоида и сдвига (dX, dY, dZ), без параметров вращения, корректировки масштаба или начального меридиана. В этих случаях, Вы можете использовать номер региона 999 вместо 9999, и упрощенное определение:

`999, EllipsoidNumber, dX, dY, dZ`

Эти настройки параметров копируются в файл MAPINFOW.PRJ; смотрите пример ниже.

Определение картографического региона в MapX

В MapX, Вы определяете собственный регион, создавая объект **Datum**, и используя его метод **Set** для установки параметров. Вы затем используете этот объект **Datum** как один из параметров метода **CoorsSys.Set**.

Общие ошибки и как их избежать

Параметры сдвига и вращения описывают ориентацию эллипсоида в пространстве. Важно убедиться в том, что эти параметры имеют правильный знак (положительный или отрицательный). Обычно документация описывающая локальный регион приводит список параметров, требуемых для конвертации из локального региона в регион WGS 84. (Это тоже самое, что сказать что эти параметры были получены при вычитании локального региона из WGS 84.) В этом случае, Вы можете использовать параметры точно такими какими они описаны в документации. Таким образом, если Вы имеете документ со списком параметров для конвертации координат в противоположном направлении — из WGS 84 в локальный регион — то Вы можете поменять знак для параметров сдвига, вращения и коррекции масштаба.

Так же очень важно помещать параметры в строго заданной очередности. В некоторых списках параметры вращения пишутся в таком порядке - EZ первым, вот так: EZ, EY, EX. В этих случаях, исправьте порядок следования параметров так как надо для MapX.

Преобразование картографического региона

При конвертации координат из одного региона в другой, MapX использует методы Молоденского (3-параметра) и Бурса-Вольфа (7-параметров). Это основные методы для конвертации координат из одного региона в другой.

После введения региона NAD 83, NOAA разработала программу NADCON, расшифровывающуюся как North American Datum CONversion. Это весьма специализированная программа, преобразующая координаты только из NAD 27 в NAD 83 и обратно. Для этой специализированной задачи есть метод, более точный чем метод универсальный Молоденского; NADCON дает точность до 0.1 метра, и метод Молоденского может дать точность только 10-30 метров. Большинство правительственных организаций США стандартизировали программу NADCON для конвертации между NAD 27 и NAD 83.

Начиная с MapX 3.5, алгоритм NADCON используется для конвертации координат между NAD 27 и NAD 83 если такие координаты лежат в тех местах, которые покрываются NADCON (США, Пуэрто Рико и Виргинские острова). Если координаты лежат вне этих областей или используются регионы отличные от NAD 27 или NAD 83, MapX использует методы конвертации Молоденского или Бурса-Вольфа.

Поскольку требуется доступ к файлу, метод NADCON работает значительно медленнее чем метод Молоденского. Если Вы хотите отключить метод NADCON, добавьте “NADCON” в registry. Registry должен иметь такой путь:

- HKEY_LOCAL_MACHINE\Software\MapInfo\MapX\3.0\NadCon

Если этот ключ имеет значение ноль, то будет использоваться метод Молоденского вместо NADCON. Чтобы пользоваться методом NADCON, установите значение ключа на 1 (по умолчанию). MapX читает этот ключ при загрузке, и все программы используют эти же настройки. Вы не можете отключить NADCON или включить его для отдельной карты.

Дополнительная информация...

Дополнительную информацию о системах координат и проекциях карт можно найти в книгах:

- John P. Snyder. Map Projections—A Working Manual. Washington: U.S. Geological Survey Professional Paper 1395. 1987.
- John P. Snyder and Philip M. Voxland. An Album of Map Projections. Washington: U.S. Geological Survey Professional Paper 1453. 1989.

Заказать эти книги можно по адресу:

Earth Science Information Center
U.S. Geological Survey
507 National Center
Reston, VA 22092
703-860-6045 or 1-800-USA-MAPS

14

Использование сборочных слоев

- ♦ Использование сборочных слоев
- ♦ Что такое Сборочный слой?
- ♦ Термины и Концепции
- ♦ Как делать приложения со сборочными слоями
- ♦ Подготовка сборочного слоя
- ♦ Создание инструмента сборки
- ♦ Переустановка сборочного слоя
- ♦ Ограничения и требования для сборочного слоя
- ♦ Дополнительная информация...

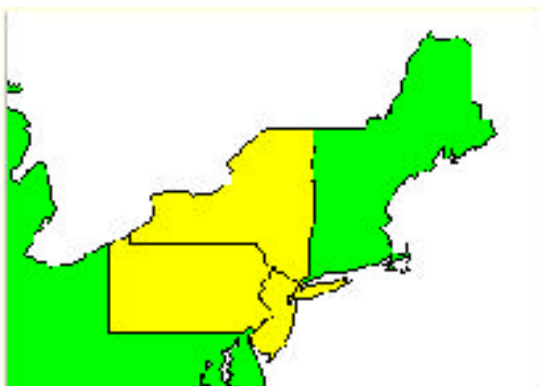
В этой главе, Вы познакомитесь со специальным типом слоев карты, известным как Сборочный слой, который позволяет пользователю проводить “сквозной” анализ. Если Ваша карта включает сборочный слой, может подвести курсор к нужному месту, нажать кнопку мышки и увидеть этот регион более детально. Сборочные слои обеспечивают интуитивный, легкий в использовании интерфейс, позволяющий пользователю выявлять данные при нажатии кнопки мышки.

Что такое сборочный слой?

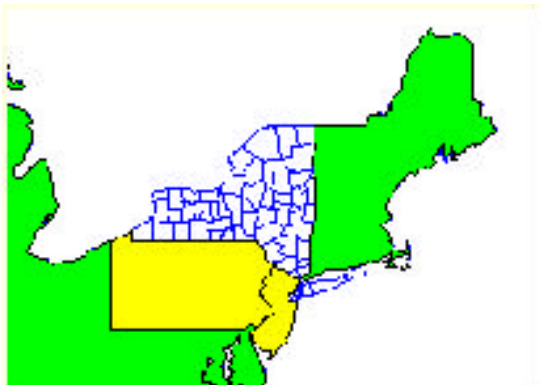


Сборочный слой это слой карты со специальным поведением. Сборочные слои позволяют пользователю исследовать карту по иерархическому принципу, указывая курсором на нужный регион и нажимая кнопку мышки, если надо познакомиться с регионом поподробнее.

Когда карта появляется на экране, сборочный слой выглядит как обычный слой карты. Например, данный сборочный слой показывает территории продаж.



Если пользователь выбрал инструмент сборки, а затем щелкнул мышкой на объекте карты, то этот объект сразу изобразится более подробно. Например, если Вы щелкаете на территории продаж, на одном конкретном объекте с границами, то это воспринимается сборочным слоем как команда “Показать более детально эту часть территории”.



В зависимости от того как был создан сборочный слой и как работает приложение, пользователь может увеличивать подробность объекта карты неоднократно. С каждым успешным щелчком мышки карта может показывать более детальную обстановку вокруг выбранного места. Если щелкнуть на границе штата, то появятся границы графств, которые этот штат образуют.

Чтобы уменьшить образовавшуюся детальность, пользователь может выбрать другой инструмент "возврат", и щелкните на выбранной области. Детальное изображение исчезнет.

Сборный слой строится из ряда таблиц. Вам надо обеспечить одну таблицу для каждого “уровня” в иерерхии сборочного слоя, плюс одну специальную таблицу, которая будет определять как различные уровни соотносятся друг с другом иерархически. Таким образом, сложный состав сборочного слоя скрыт от пользователя. Сборочный слой появляется в карте MapX как отдельный слой. Если пользователь открывает диалог Управление слоями, сборочный слой появляется в нем как отдельный слой среди прочих в списке.

Термины и Концепции

Что бы усвоить все содержание этой главы, Вам надо понять некоторые термины и концепции сборной картографии.

- Что бы поддерживать сборочную картографию, карта должна включать **Сборочный слой**. Это специальный слой, составленный из двух и более таблиц. Сборочный слой отображает объекты из различных таблиц как единый слой карты.
- Для сборочного слоя важно, чтобы он содержал хотябы два уровня, которые формируют иерерхию. Например, предположим, что сборочный слой представляет границы штатов и пользователь может щелкнуть мышкой на штате и увидеть все границы графств, входящих в штат. В предыдущем примере границы штатов представляли собой один уровень а границы графств представляют другой уровень. Уровни формируют иерархию. Каждый штат содержит некоторое число графств. В этом примере, мы говорим, что граница штата “материнский” объект, а каждая граница графства это “дочерний” объект.
- Обычно пользователь использует сборочный слой только в части карты, нажимая специальный инструмент Сборочного слоя, там где надо повысить детальность. Для создания инструмента просмотра сборочного слоя используйте метод **CreateCustomTool**.

- Использование инструмента просмотра сборочного слоя приводит в действие событие **ToolUsed**. Внутри этого события, Вы обрабатываете действие по активизированию сборочного слоя при щелкание мышкой на объекте карты в режиме просмотра сборочного слоя. Преобразование объекта в более детальный объект происходит в два этапа: во-первых, Вы получаете список “дочерних” объектов, которые будут помещены в тот объект на котором Вы щелкните и будут добавлены к слою (при вызове метода **Layer.DrilldownAddFeatures**). Затем происходит удаление исходного объекта на котором пользователь щелкнул мышкой. Другими словами, объект замещается “дочерними” объектами.
- Любое приложение со сборочным слоем требует особого менеджера иерерхии “уровней”, являющегося программным компонентом, понимающим иерархические соотношения между различными “уровнями” в сборочном слое. Когда пользователь щелкает мышкой на объекте, то менеджер иерархии определяет, какие “дочерние” объекты будут добавлены к карте вместо “материнского”.
- Процедура события **ToolUsed** это простой пример менеджера иерархии. Когда пользователь работает с инструментом показа сборочного слоя и щелкает на объекте карты, событие щелчка обрабатывается процедурой события **ToolUsed**; код выполняется внутри этой процедуры, определяя какие объекты будут добавлены для расширения детальности карты.

- Вы можете так составить сборочный слой, что в нем будет две или больше альтернативных таблиц, которые могут быть показаны на данном уровне иерархии. Например, предположим, что для каждой границы штата у вас есть две альтернативы “дочерних” объектов: Штат может быть замещен границами или штат может быть замещен телефонными номерами. Такой сборочный слой считается двухуровневым—один уровень “материнский” и один “дочерний”. При попадании на “дочерний” уровень, существует выбор между двумя типами “дочерних” объектов (графство или телефоны). При такой организации сборочного слоя, приложение должно определить какой из “дочерних” уровней надо показывать, возможно появится подсказка для пользователя.

Как разрабатывать приложение со сборочным слоем

Приложения со сборочным слоем требуют внимательного подхода к настройке и подготовке такого слоя. Важнейшие шаги могут быть сформулированы следующим образом (более детальное описание каждого шага будет в конце главы).

1. Получение различных таблиц которые Вы будете использовать для построения многоуровневого сборочного слоя. Таблицы могут быть созданы с использованием MapInfo Professional или получены от MapInfo Corporation или от третьих лиц.
2. Создание новой, пустой сборочной таблицы (.tab файл) со специальными колонками и специальными метаданными. Метаданные присваивают имя уровня каждой таблице компонентов и также идентифицируют важнейшие колонки в таблицах компонентов — колонку ID и колонку заголовков.
3. Включение Вашей сборочной таблицы в карту (то есть, включение сборочной таблицы в рабочий набор, который Вы используете или добавление сборочной таблицы к Вашей карте такими методами как [Layers.Add](#)).
4. Добавление элементов пользовательского интерфейса (кнопки панелей) к приложению, так что пользователь сможет выбирать инструмент просмотра сборочного слоя и после этого щелкать на объекте, чтобы увидеть вместо него “дочерние” объекты.

5. Добавление кода к приложению, который поможет использовать инструментарий показа сборочного слоя (другими словами, это написание иерархического менеджера). Этот код нуждается в указании какие объекты пользователь выбрал; определить какие “дочерние” объекты заменят “материнский” объект; и вставить различные методы (**DrilldownRemoveFeatures**, **DrilldownAddFeatures**) которые повышают детальность или наоборот, возвращают карту в исходное состояние.

Подготовка сборочного слоя

Для создания сборочного слоя, надо иметь две или больше таблиц MapInfo. Точнее, Вам надо:

- Одну таблицу MapInfo для каждого уровня детальности в сборочном слое. Они также называются таблицами компонент.
- Кроме того, нужна пустая таблица, содержащая специальные метаданные, описывающие таблицы компонент. Ее называют **Сборочная таблица**.

Например, предположим, что надо показать на карте границы штатов, но позволить пользователю щелкнуть на штате и увидеть границы графств—двухуровневый сборочный слой. Вам нужны три таблицы: А таблица (компонентов) с границами штатов, таблица (компонентов) с границами графств и сборочная таблица.

Внимание: Сборочная таблица является "пустой" в которой не хранятся собственные данные (кроме метаданных). Когда Вы показываете сборочный слой на карте, MapX создает временную таблицу и затем копирует объекты из таблиц(ы) компонент во временную таблицу. Объекты, появляющиеся в сборочном слое в действительности являются копией объектов в таблице(ах) компонент. Когда приложение MapX завершает работу, временная таблица стирается все что остается от сборочной таблицы это .tab.

Требования для таблиц компонент

Каждый объект в сборочном слое должен иметь идентификационный ключ (желательно строинг типа "New York"). Все ключи внутри одного уровня должны быть уникальными; например, уровень с границами штатов может содержать только один штат с названием "Washington". Таким образом, идентификационные ключи объектов не обязаны быть уникальными если речь идет о разных уровнях сборочного слоя. Например, если сборочный слой включает границы штатов и границы графств, Вы можете использовать название "Washington" для штата и "Washington" для графства.

Требования для сборочной таблицы

Типичная таблица MapInfo состоит из нескольких файлов; например, таблица World состоит из файлов World.tab, World.map, World.id, World.ind и World.dat. Сборочная таблица отличается от других таблиц, она состоит из одного файла: filename.tab.

Файл .tab это текстовый файл, который Вы можете просматривать и редактировать в любом текстовом редакторе.

Файл .tab сборочной таблицы должен определять специфические колонки и ключи метаданных, как будет описано ниже.

Сборочная таблица должна определять три стандартных колонки: Key, Level и Label. Все три колонки являются символьными, размером 32 символа.

Сборочная таблица должна содержать настройки ключей метаданных. Синтаксис ключей метаданных такой:

- Ключевое слова begin_metadata маркирует начало части метаданных в .tab файле.
- Каждая строка метаданных имеет два элемента: ключ и значение. Например, ключ "\IsDrilldown" имеет значение "True". Все ключи и значения заключены к кавычки.

- Сборочная таблица должна включать ключ `\IsDrilldown` и этот ключ должен иметь значение `True`.
- Каждый ключ начинается с наклонной черты `"\"` (backslash).
- Ключи метаданных могут располагаться в иерархии. Каждый уровень в иерархии помечается наклонной чертой (`\`). Значения ключа ограничено 239 символами.
- Сборочная таблица включает иеранхию ключей `\DDMap\ComponentMaps\` . Внутри этой иерархии Вы указываете четыре ключа метаданных для каждого компонента таблицы:

Ключи метаданных в компонентных таблицах

Ключ	Описание
File	Ключ, который идентифицирует путь и имя компонентной таблицы.
LevelID	Ключ, который определяет идентификатор для этой компонентной таблицы. Пример: Вы можете использовать "States" как значение ключа, если эта таблица содержит границы штата. Когда вызывается метод, такой как DrilldownReset укажите "States" как аргумент уровня.
FeatureIDCol	Идентифицирует число колонок в компонентной таблице, которая содержит уникальные сборочные ключи. Дополнительный ключ; если он пропущен, используется номер колонки 1.
FeatureCaptionCol	Идентифицирует число колонок в компонентной таблице, которые должны использоваться для подписывания. Дополнительный ключ; если он пропущен, используется номер колонки 1.

Например, “Sample Drilldown Table” включает в себя иерархии ключей такие как “\DDMap\ComponentMaps\One\LevelID” и “\DDMap\ComponentMaps\Twox\LevelID”. Обратите внимание, что DDMap, ComponentMaps и LevelID это стандартные, необходимые компоненты иерархии ключей, в то время как One и Twox это уронви, созданные пользователем в данном случае. Вы можете использовать любые имена ключей вместо One, Twox, Twoxx и др.; эти имена ключей не имеют другого смысла, как отделять различные имена ключей в компонентной таблице.

Внутри иерархии ключей \DDMap\HierarchyManager\ существуют три дополнительных ключа:

Ключ	Описание
IsDLL	Не используется в версии 4; зарезервирован для следующих версий. Логический индикатор; "TRUE" означает, что этот сборочный слой использует DLL в качестве менеджера иерерхий.
ID	Не используется в версии 4; ; зарезервирован для следующих версий. Это имя DLL или GUID к CoCreate.
InitialLevel	Начальная компонентная таблица для отображения при первой загрузке/отображения сборочного слоя. Дополнительный ключ; он применяется если надо завести пустой сборочный слой.

Пример сборочной таблицы

Ниже приводится содержание сборочной таблицы - примера. Если Вы хотите создать собственную сборочную таблицу, Вы можете копировать этот пример в текстовый файл и изменять его, в соответствии с вашими компонентными таблицами.

```
!table
!version 300
!charset WindowsLatin1
```


Definition Table

Type Native Charset "WindowsLatin1"

Fields 3

Key Char (32);

Level Char (32);

Label Char (32);

begin_metadata

"\IsDrilldown" = "TRUE"

"\DDMap\ComponentMaps\One\File" = "C:\Program
Files\MapInfo\Data\USA\usaXXX.TAB"

"\DDMap\ComponentMaps\One\LevelID" = "USA"

"\DDMap\ComponentMaps\One\FeatureIDCol" = "3"

"\DDMap\ComponentMaps\One\FeatureCaptionCol" = "1"

"\DDMap\ComponentMaps\Twox\File" = "C:\Program
Files\MapInfo\Data\USA\2Region.TAB"

"\DDMap\ComponentMaps\Twox\LevelID" = "2Region"

"\DDMap\ComponentMaps\Twox\FeatureIDCol" = "1"

"\DDMap\ComponentMaps\Twox\FeatureCaptionCol" = "3"

"\DDMap\ComponentMaps\Twoxx\File" = "C:\Program
Files\MapInfo\Data\USA\MultiRegionSales.TAB"

"\DDMap\ComponentMaps\Twoxx\LevelID" = "MultiRegion"

"\DDMap\ComponentMaps\Twoxx\FeatureIDCol" = "1"

"\DDMap\ComponentMaps\Twoxx\FeatureCaptionCol" = "3"

"\DDMap\ComponentMaps\Two\File" = "C:\Program
Files\MapInfo\Data\USA\states.TAB"

"\DDMap\ComponentMaps\Two\LevelID" = "States"

"\DDMap\ComponentMaps\Two\FeatureIDCol" = "3"

```

"\DDMap\ComponentMaps\Two\FeatureCaptionCol" = "1"

"\DDMap\HierarchyManager\IsDLL" = "TRUE"
"\DDMap\HierarchyManager\ID" = "SomeDLL.dll"
"\DDMap\HierarchyManager\InitialLevel" = "USA"
end_metadata

```

Создание инструмента Сборки (Drilldown Tool)

Создаваемый Вами пользовательский интерфейс должен позволять пользователю просматривать уровни сборочного слоя для отдельных объектов карты. Для этого чаще всего используется инструментарий Сборки (например, кнопки), после активации которых, надо щелкнуть мышкой на нужном объекте карты.

Можно создать инструментарий **Сборки** используя метод **CreateCustomTool**. Установка констант **CursorConstants** включает два курсора существующих специально для приложений со сборочным слоем: **miDrilldownExpandCursor** и **miDrilldownContractCursor**.

Например:

```

' Drilldown Expand Tool
Map1.CreateCustomTool 1, miToolTypePoint,
miDrilldownExpandCursor, miDrilldownContractCursor,
miDrilldownContractCursor

```

Этот пример создает одиночный инструмент, который может быть использован в двух действиях **Показать сборочный слой (Drilldown)** и **Скрыть сборочный слой (RollUp)**. По умолчанию, инструмент действует как инструмент **Показать сборочный слой**, и отображает "расширенный" курсор (курсор со знаком +). Если пользователь держит нажатыми клавиши **SHIFT** или **CTRL** пока нажимает клавишу мышки, то курсор меняет знак на противоположный (со знаком "-"). Такой подход позволяет пользователю переключать режимы работы со сборочным слоем не убирая курсор с объекта; встроенные инструменты **ZoomIn** и **ZoomOut** имеют такое же поведение.

После того как Вы создали инструментарий, Вы можете активизировать инструмент установкой свойства **CurrentTool**.

```
Map1.CurrentTool = 1
```

Каждое использование пользовательского инструмента **Сборки** приводит к действию события **ToolUsed**. Внутри процедуры события **ToolUsed**, Вам надо выполнить код, который приводит к действию показ сборочных слоев. Этот процесс состоит из четырех стадий:

1. Определите на каком объекте карты пользователь щелкнул мышкой, используя такие методы как **SelectByPoint** или **SearchAtPoint**.
2. Определите какие "дочерние" объекты будут помещены вместо "материнского" когда пользователь щелкнет Мышкой. Например, Вы можете использовать одним или несколькими операторами **Case** для определения какие "дочерние" объекты будут замещать выбранный "материнский" объект.
3. Вызовите метод **DrilldownAddFeatures** для добавления "дочерних" объектов к карте.
4. Вызовите метод **DrilldownRemoveFeatures** для удаления "материнского" объекта (на котором пользователь щелкнул мышкой).

Внимание: Эти действия добавить/удалить не изменяют компоненты таблицы; Вы не можете "редактировать" таблицы. Когда Вы используете метод **DrilldownAddFeatures** для добавления объектов, проявляется только эффект копирования объектов в тот ряд объектов, которые являются видимыми.

Создание инструмента отмены сборки

После того, как Вы создали инструмент показа сборочного слоя, Может понадобится сделать инструмент скрытия сборочного слоя, имеющий действие обратное инструменту показа сборочного слоя.

Вы можете создать Инструмент скрытия сборки таким же образом, как Вы создали показа сборки. Вам надо использовать те же методы (**DrilldownAddFeatures** и **DrilldownRemoveFeatures**). Разница в том, что вместо добавления "дочерних" и удаления "материнского", Вы делаете наоборот — добавляете "материнский" объект и удаляете "дочерние".

Пример: Инструмент Показа/Скрытия Сборки

Следующий пример процедуры события **ToolUsed** демонстрирует как обрабатывать инструмент **Показа** и **Скрытия сборки**.

Эта процедура написана что бы обработать сборочный слой со следующей иерархией слоев (уровней):

USA	Слой содержащий отдельный регион, представляющий целиком США.
2Region	Слой, содержащий два больших региона (Восток и Запад). Каждый регион покрывает примерно половину США.
MultiRegion	Слой содержащий восемь небольших регионов. Каждый регион покрывает примерно восьмую часть США.
States	Слой, содержащий 50 штатов Америки.

```

Private Sub Map1_ToolUsed(ByVal ToolNum As Integer,
    ByVal X1 As Double, ByVal Y1 As Double, ByVal X2 As
    Double, ByVal Y2 As Double, ByVal Distance As
    Double, ByVal Shift As Boolean, ByVal Ctrl As
    Boolean, EnableDefault As Boolean)

    Dim AddKeys() As String
    Dim NewLevel As String
    Dim fs As Features
    Dim strLevel As String
    Dim DelKeys() As String
    Dim pnt As New Point

    ' -----
    ' Expand
    ' -----

    If ToolNum = 1 Then
        ' Expand the feature the user clicked on.
        ' First, figure out what object we have
        pnt.Set X1, Y1
        Set fs =
Map1.Layers(strTbl).SearchAtPoint(pnt)

        If fs.Count = 1 Then
            ReDim DelKeys(0)
            Map1.Layers(strTbl).KeyField = "Level"
            strLevel = fs(1).KeyValue

            Map1.Layers(strTbl).KeyField = "Key"
            DelKeys(0) = fs(1).KeyValue

            If strLevel = "USA" Then

```

```

NewLevel = "2Region"
ReDim AddKeys(1)
AddKeys(0) = "West"
AddKeys(1) = "East"

ElseIf strLevel = "2Region" Then
    NewLevel = "MultiRegion"
    ReDim AddKeys(3)
    Select Case DelKeys(0)
        Case "West"
            AddKeys(0) = "mrRgn1"
            AddKeys(1) = "mrRgn2"
            AddKeys(2) = "mrRgn3"
            AddKeys(3) = "mrRgn4"
        Case "East"
            AddKeys(0) = "mrRgn5"
            AddKeys(1) = "mrRgn6"
            AddKeys(2) = "mrRgn7"
            AddKeys(3) = "mrRgn8"
    End Select

ElseIf strLevel = "MultiRegion" Then
    NewLevel = "States"
    Select Case DelKeys(0)
        Case "mrRgn1"
            ReDim AddKeys(4)
            AddKeys(0) = "16"
            AddKeys(1) = "30"
            AddKeys(2) = "41"
            AddKeys(3) = "53"
    End Select

```

```

        AddKeys(4) = "56"
Case "mrRgn2"
    ReDim AddKeys(5)
    AddKeys(0) = "04"
    AddKeys(1) = "06"
    AddKeys(2) = "08"
    AddKeys(3) = "32"
    AddKeys(4) = "35"
    AddKeys(5) = "49"

Case "mrRgn3"
    ReDim AddKeys(4)
    AddKeys(0) = "19"
    AddKeys(1) = "27"
    AddKeys(2) = "31"
    AddKeys(3) = "38"
    AddKeys(4) = "46"
Case "mrRgn4"
    ReDim AddKeys(5)
    AddKeys(0) = "48"
    AddKeys(1) = "22"
    AddKeys(2) = "05"
    AddKeys(3) = "40"
    AddKeys(4) = "20"
    AddKeys(5) = "29"
Case "mrRgn5"
    ReDim AddKeys(4)
    AddKeys(0) = "17"
    AddKeys(1) = "55"
    AddKeys(2) = "18"

```

```

        AddKeys(3) = "26"
        AddKeys(4) = "39"
    Case "mrRgn6"
        ReDim AddKeys(8)
        AddKeys(0) = "33"
        AddKeys(1) = "09"
        AddKeys(2) = "23"
        AddKeys(3) = "25"
        AddKeys(4) = "34"
        AddKeys(5) = "36"
        AddKeys(6) = "42"
        AddKeys(7) = "44"
        AddKeys(8) = "50"
    Case "mrRgn7"
        ReDim AddKeys(7)
        AddKeys(0) = "21"
        AddKeys(1) = "37"
        AddKeys(2) = "47"
        AddKeys(3) = "51"
        AddKeys(4) = "54"
        AddKeys(5) = "10"
        AddKeys(6) = "11"
        AddKeys(7) = "24"
    Case "mrRgn8"
        ReDim AddKeys(4)
        AddKeys(0) = "28"
        AddKeys(1) = "01"
        AddKeys(2) = "12"
        AddKeys(3) = "13"
        AddKeys(4) = "45"

```



```

        End Select
    Else
        MsgBox "Can't drill down any
further; Level: " _
        + strLevel + " Key: " + DelKeys(0)
        Exit Sub
    End If
    ElseIf fs.Count = 0 Then
        MsgBox "No features selected"
        Exit Sub
    Else
        MsgBox "More than one feature selected;
can only _
        drilldown on exactly one item"
        Exit Sub
    End If
    Map1.Layers(strTbl).DrillDownRemoveFeatures
strLevel, _
    DelKeys
    Map1.Layers(strTbl).DrillDownAddFeatures
NewLevel, AddKeys
    ' -----
    ' Contract
    ' -----
    ElseIf ToolNum = 2 Then
        ' do the drilldown contract thang
        pnt.Set X1, Y1
        Set fs =
Map1.Layers(strTbl).SearchAtPoint(pnt)

        If fs.Count = 1 Then

```

```

ReDim DelKeys(0)
Map1.Layers(strTbl).KeyField = "Level"
strLevel = fs(1).KeyValue

Map1.Layers(strTbl).KeyField = "Key"
DelKeys(0) = fs(1).KeyValue

If strLevel = "States" Then
    NewLevel = "MultiRegion"
    Select Case DelKeys(0)
        Case "16", "30", "41", "53",
"56"

            ReDim DelKeys(4)
            ReDim AddKeys(0)
            AddKeys(0) = "mrRgn1"
            DelKeys(0) = "16"
            DelKeys(1) = "30"
            DelKeys(2) = "41"
            DelKeys(3) = "53"
            DelKeys(4) = "56"

        Case "04", "06", "08", "32",
"35", "49"

            ReDim DelKeys(5)
            ReDim AddKeys(0)
            AddKeys(0) = "mrRgn2"

            DelKeys(0) = "04"
            DelKeys(1) = "06"
            DelKeys(2) = "08"

```

```

DelKeys(3) = "32"
DelKeys(4) = "35"
DelKeys(5) = "49"

Case "19", "27", "31", "38",
"46"

ReDim DelKeys(4)
ReDim AddKeys(0)
AddKeys(0) = "mrRgn3"

DelKeys(0) = "19"
DelKeys(1) = "27"
DelKeys(2) = "31"
DelKeys(3) = "38"
DelKeys(4) = "46"

Case "48", "22", "05", "40",
"20", "29"

ReDim DelKeys(5)
ReDim AddKeys(0)
AddKeys(0) = "mrRgn4"

DelKeys(0) = "48"
DelKeys(1) = "22"
DelKeys(2) = "05"
DelKeys(3) = "40"
DelKeys(4) = "20"
DelKeys(5) = "29"

```

```

"39"
Case "17", "55", "18", "26",

ReDim DelKeys(4)
ReDim AddKeys(0)
AddKeys(0) = "mrRgn5"

DelKeys(0) = "17"
DelKeys(1) = "55"
DelKeys(2) = "18"
DelKeys(3) = "26"
DelKeys(4) = "39"

Case "33", "09", "23", "25",
"25", "34", _
"36", "42", "44", "50"
ReDim DelKeys(8)
ReDim AddKeys(0)
AddKeys(0) = "mrRgn6"
DelKeys(0) = "33"

DelKeys(1) = "09"
DelKeys(2) = "23"
DelKeys(3) = "25"
DelKeys(4) = "34"
DelKeys(5) = "36"
DelKeys(6) = "42"
DelKeys(7) = "44"
DelKeys(8) = "50"

Case "21", "37", "47", "51",
"54", "10", _

```

```

"11", "24"

    ReDim DelKeys(7)
    ReDim AddKeys(0)
    AddKeys(0) = "mrRgn7"

    DelKeys(0) = "21"
    DelKeys(1) = "37"
    DelKeys(2) = "47"
    DelKeys(3) = "51"
    DelKeys(4) = "54"
    DelKeys(5) = "10"
    DelKeys(6) = "11"
    DelKeys(7) = "24"

Case "28", "01", "12", "13",
"45"

    ReDim DelKeys(4)
    ReDim AddKeys(0)
    AddKeys(0) = "mrRgn8"

    DelKeys(0) = "28"
    DelKeys(1) = "01"
    DelKeys(2) = "12"
    DelKeys(3) = "13"
    DelKeys(4) = "45"

End Select

ElseIf strLevel = "MultiRegion" Then
    NewLevel = "2region"

```

```

        Select Case DelKeys(0)
            Case "mrRgn1", "mrRgn2",
"mrRgn3", "mrRgn4"

                ReDim DelKeys(3)
                ReDim AddKeys(0)
                AddKeys(0) = "West"
                DelKeys(0) = "mrRgn1"
                DelKeys(1) = "mrRgn2"
                DelKeys(2) = "mrRgn3"
                DelKeys(3) = "mrRgn4"

            Case "mrRgn5", "mrRgn6",
"mrRgn7", "mrRgn8"

                ReDim DelKeys(3)
                ReDim AddKeys(0)
                AddKeys(0) = "East"
                DelKeys(0) = "mrRgn5"
                DelKeys(1) = "mrRgn6"
                DelKeys(2) = "mrRgn7"
                DelKeys(3) = "mrRgn8"

        End Select

    ElseIf strLevel = "2Region" Then
        NewLevel = "USA"
        ReDim DelKeys(1)
        ReDim AddKeys(0)
        AddKeys(0) = "1"
        DelKeys(0) = "West"
        DelKeys(1) = "East"

```

```

        Else
            MsgBox "Can't contract any more;
Level: " + strLevel + "   Key: " + DelKeys(0)
            Exit Sub
        End If

    Else
        MsgBox "No items or Multiple items under
cursor; Can only Drilldown on one item"
        Exit Sub
    End If

    Map1.Layers(strTbl).DrillDownRemoveFeatures
strLevel, DelKeys
    Map1.Layers(strTbl).DrillDownAddFeatures
NewLevel, AddKeys
    ElseIf ToolNum = 3 Then
        Dim f As Feature
        Dim fss As Features
        Dim s As Style
        Dim key() As String

        ' Получение объекта
        pnt.Set X1, Y1
        Set fss =
Map1.Layers(strTbl).SearchAtPoint(pnt)

        Map1.Layers(strTbl).KeyField = "Level"
        strLevel = fss(1).KeyValue

        Map1.Layers(strTbl).KeyField = "Key"

```

```

        ReDim key(0)
        key(0) = fss(1).KeyValue

        If fss.Count = 1 Then

            Set fss = _

Map1.Layers(strTbl).GetDrilldownFeaturesByID(strLevel, key)

            Set f = fss(1)
            Set s = f.Style

            s.RegionColor = (RGB(0, 0, 255))
            f.Style = s

            f.Update
        Else
            MsgBox "Can only change one feature at a
time"

            Exit Sub
        End If
    End If
End Sub

```


Переустановка сборочного слоя

Вы можете “переустановить” сборочный слой вызовом метода **DrilldownReset**. Переустановка **Сборочного слоя** очищает целый слой, затем переинициализирует слой, используя объекты из одной компонентных таблиц (component tables).

Например, Вы можете включить "Reset button" в создаваемый пользовательский интерфейс. Когда пользователь нажимает на эту кнопку, происходит вызов метода **DrilldownReset** для пересохранения первоначального состояния **Сборочного слоя**.

Включение кнопки переустановки это хорошая идея, потому что дает пользователю быстрый и легкий путь для пересохранения сборочного слоя в гомогенном состоянии.

Пример

```
Private Sub Command2_Click()  
    Map1.Layers("DDTestUSA2").DrilldownReset  
    "States"  
End Sub
```

Ограничения и требования к Сборочному слою

Некоторые ограничения применяются к сборочному слою, ниже приводится их список.

- Каждый объект в **Сборочном слое** должен содержать индекс ID который является уникальным в данной компонентной таблице (хотя ID не должен быть уникальным среди всех компонентных таблиц, которые входят в сборочный слой).
- Вы не можете использовать растровое изображение как подложку в сборочном слое.
- Когда создается тематический слой с **Themes.Add**, вычисляются диапазоны для слоев с большим числом строк, для таких слоев как сборочные или серверные, это может занять много времени. Параметр **ComputeTheme** из метода **Add** позволяет Вам создавать невычисляемые тематические слои для любого типа тематики. Невычисляемые тематические слои дают Вам возможность создавать тематику без автоматического вычисления диапазонов. Вы можете затем создать диапазоны самостоятельно. Это осуществляется быстрее для сборочных или серверных слоев.
- Сборочный слой не “помнит” статус различных сборочных уровней (какие объекты были заменены более подробными). Если Вы хотите, что бы приложение хранило точный статус карт использованных ранее, надо написать код для хранения сборочных настроек карты, когда Вы просматриваете сборочный слой и обращаться к этим настройкам при старте сеанса.
- Хотя Вы можете редактировать объекты в сборочном слое, результаты редактирования не сохраняется и компонентные таблицы не подвергаются редактированию. Когда Вы редактируете сборочный слой, Вы не изменяете компонентной таблицы, Вы изменяете временную копию объекта из компонентной таблицы.

Дополнительная информация...

Для более подробной информации о сборочных слоях, смотрите следующие разделы.

Методы и свойства, относящиеся к сборочным слоям

Метод `Layer.DrilldownAddFeatures` добавляет объекты к сборочному слою.

Метод `Layer.DrilldownRemoveFeatures` удаляет объекты из сборочного слоя.

Метод `Layer.DrilldownReset` очищает целый сборочный слой, когда показывается только один уровень детальности из слоя.

Метод `Layer.GetDrilldownFeaturesByID` отыскивает объекты Features по заданным ID.

Что бы определить, является ли слой сборочным, проверьте, возвращает ли свойство `Layer.Type` значение `miLayerTypeDrilldown` (значение: 7).

Константы, относящиеся к сборочным слоям

Установка `CursorConstants` включает две константы (`miDrilldownExpandCursor` и `miDrilldownContractCursor`) специально приспособленных для инструментов [Показать/Скрыть сборочный слой](#). Используйте эти константы с методом `Map.CreateCustomTool`.

15

Экспорт карт

- ◆ Экспорт карт
- ◆ Свойство ExportSelection
- ◆ Печать карт

Часто пользователю может понадобиться распечатать карту или другой картографический материал из другого приложения. MapX имеет методы, позволяющие посылать содержимое карты в буфер, на принтер или в графический файл.

Экспорт карт

Что бы экспортировать карту в графический файл или скопировать содержимое карт в буфер, используйте метод **ExportMap**.

ExportMap	Экспортирует карту в графический файл.	Map1.ExportMap _ “C:\Map.TIF”,_ miFormatTIF
ExportMap	Экспортирует карту в буфер.	Map1.ExportMap “clipboard”

Внимание: Дополнительные параметры в [квадратных скобках]

Map.ExportMap (Location, Format, [W], [H])

Location	Указывает куда помещать файл. Если ключевое слово ‘CLIPBOARD’ используется, изображение помещается в буфер.
Format	Выходной формат. Берется значение ExportFormatConstants.
W	Ширина вывода. . Это величина двойной точности, указывающая высоту в “бумажных” единицах” (Map.PaperUnit). Это дополнительный параметр и если он не указан, используется Map.MapPaperHeight.
H	Высота вывода. Это величина двойной точности, указывающая высоту в “бумажных” единицах” (Map.PaperUnit). Это дополнительный параметр и если он не указан, используется Map.MapPaperHeight.

Константы формата

Формат	Описание
miFormatWMF	Metafile
miFormatBMP	Bitmap

miFormatJPEG	JPEG image
miFormatTIF	TIF
miFormatGIF	GIF Image
miFormatPNG	Portable Network Graphics
miFormatPSD	PhotoShop

```

'This sample demonstrates the Map.ExportMap method.
It uses
'the method to place a map in the clipboard as a
BMP.
Map1.PaperUnit = miUnitCentimeter

'Export a 12 cm by 9 cm map to the clipboard in BMP
Format
Map1.ExportMap "clipboard", miFormatBMP, 12, 9

```

Свойство **ExportSelection**

Использование свойства **ExportSelection** карты позволит Вам контролировать, будет ли выделенный фрагмент карты экспортироваться вместе с картой. Посмотрите на пример кода, использующего это свойство:

```
'Export a jpg image for display and include the  
selection highlighting in the image  
Map1.ExportSelection=True  
Map1.ExportMap "c:\temp\map.jpg", miFormatJPEG
```

Печать карт

Для печати карт используйте метод **PrintMap**.

Внимание: Текущая карта прорисована так, что заполняет прямоугольник. Наилучшие результаты получаются когда выдерживается пропорция соотношения высоты к ширине. Этот метод просто печатает содержимого карты MapX; при этом не показываются иконы, панели инструментов или меню. Вы можете захотеть распечатать форму, содержащую контроль MapX если Вам надо увидеть дополнительные настройки вашего вывода.

Метод	Описание	Пример программного кода
PrintMap	Печатает карту.	Map1.PrintMap Printer.hDC, 0, 0, Map1.Width * 100, Map1.Height * 100

Map.PrintMap (hDC, x, y, w, h)

Часть	Описание
hDC	Описание устройства принтера. Может быть любым устройством.
x	Верхний левый угол X в единицах HIMETRIC.
y	Верхний левый угол Y в единицах HIMETRIC.
w	Ширина в единицах HIMETRIC.
h	Высота в единицах HIMETRIC.

16

Работа с Visual C++

- ◆ Знакомство с примерами приложений
- ◆ Обновление приложений
- ◆ Доступ к свойствам и
- ◆ Включение файла
- ◆ Создание управления
- ◆ Создание разделов меню с
- ◆ Обработка событий MapX
- ◆ Применение
пользовательских
- ◆ Связывание данных с
- ◆ Добавление быстрого
- ◆ Использование
- ◆ Обработка
- ◆ Создание карты в диалоге
C++

Один из путей изучения MapX состоит в знакомстве с примерами приложений. Примеры приложений находятся в следующей папке:
[MapInfo MapX > Samples40](#).

Знакомство с примерами приложений

Когда Вы начнете изучать приложение на C++ `mapxsamp.cpp`, следующие темы помогут Вам понять MapX.

- Обновление C++ приложений из ранних версий MapX
- Доступ к методам и свойствам MapX в C++
- Включение `Mapx.cpp` в Ваш проект
- Создание MapX контроля
- Создание разделов меню
- Обработка событий MapX
- Использование пользовательских инструментов
- Связывание данных
- Добавление “быстрых” меню
- Использование встроенных диалогов Электронной справки
- Обработка исключительных состояний MapX в C++
- Создание карты в диалоге C++

Внимание: Эти разделы предполагают, что Вы используете модель Microsoft's Document/View (стандартное приложение MFC AppWizard). Приложение-пример может быть встроено в Developer Studio с помощью `mapxsamp.mdp`.

Советы для разработчиков на C++

Если Вы используете Visual C++ 5 или ранние версии, добавьте пример проекта `mapxsamp.cpp` к Вашему рабочему набору. Это даст Вам быстрый доступ к файлам-примерам и к Class Wizard, и к файлам Вашего собственного проекта. Вы можете удалить пример проекта из рабочего набора в любой момент.

Если Вы создаете приложение MapX используя ранние версии MapX, смотрите раздел *“Обновление приложений на C++ из ранних версий MapX.”*

Обновление приложений на С++ из ранних версий MapX

Если Вы написали проложение С++ используя ранние версии MapX, Вам надо использовать новые упаковочные классы MapX (wrapper classes) - (mapx.h и mapx.cpp) для обновления Вашего приложения в текущую версию MapX.

Вам может понадобиться изменить строки, которые Вы передаете в CreateDispatch. С ранними версиями MapX, Вы могли использовать строку для указания имени объекта; например:

```
Flds.CreateDispatch("MapX.Fields")
```

С текущей версией MapX, Вам надо указать другую строку:

```
Flds.CreateDispatch("MapX.Fields.4")
```

Что бы сделать Ваш код более совместимым с будущими версиями MapX, можно использовать GetClsid вместо строки. Результат, возвращаемый GetClsid будет работать и в текущей и в будущих версиях MapX. Например:

```
Flds.CreateDispatch(Flds.GetClsid())
```

Это изменение применяется ко всем объектам, которые Вы создаете.

В объектной модели MapX, можно создавать отдельные (не прикрепленные) объекты с этими классами объектов:.

AffineTransform	BindLayer	BitmapSymols
CoordSys	Datum	Feature
Fields	LayerInfo	Map
ODBCQueryInfo	Parts	Point
Points	Rectangle	RowValue
RowValues	Style	

Доступ к свойствам и методам MapX в C++

Объекты

Каждый объект MapX выполняется в виде C++ класса в файлах MapX.h и MapX.cpp. Имя класса C++ тоже что и объекта MapX, с приставкой 'CMapX' в начале. Например, объект DataSet имеет класс, называемый CMapXDataset.

Свойства

Свойства объекта MapX выполняются функциями - элементами в (member functions) классах C++. Свойство Read/write (Чтение/запись) похоже на свойство 'Name' объекта DataSet, и будет иметь две функции-элемента в CMapXDataSet—одна устанавливает значение свойства, а вторая извлекает значение свойства. Имена функций-элементов будут иметь приставки 'Get' или 'Set' перед именем свойства.

```
CString GetName();  
void SetName(LPCTSTR);
```

Свойство, доступное только для чтения, будет иметь только одну функцию-элемент 'Get' и не будет иметь функцию 'Set'.

Методы

Методы объекта MapX выполняются функцией-элементом с тем же именем, что и имя метода объекта MapX. Заметьте, что типы многих параметров будут 'const VARIANT &'. Это означает, что функция-элемент ожидает переменную типа VARIANT для передачи.

Дополнительные параметры

Многие методы имеют параметры, рассматриваемые как дополнительные в Visual Basic или других скриптовых языках. В C++ все параметры должны быть указаны, когда вызываются эти методы (смотрите ниже “совмещенные” методы). Все дополнительные параметры рассматриваются как VARIANT. Для вызова метода MapX с дополнительным параметром, надо установить переменную следующим образом:

```
VARIANT vtOptional;  
vtOptional.vt = VT_ERROR;  
vtOptional.scode = DISP_E_PARAMNOTFOUND;
```

Файл MapX.h включает вспомогательный (helper) класс COptionalVariant который делает это в его конструкторе. Вы можете использовать это когда надо вызвать метод с дополнительным параметром.

Перезагруженные функции-элементы (Overload Member Functions)

Что бы упростить вызов методов используются нормальные типы C++ вместо VARIANTS, большинство методов MapX перегружаются с простыми параметрами со значениями по умолчанию, которые конструируют типы VARIANT и вызывают ‘реальные’ функции=элементы. Поскольку не все возможные комбинации обработаны, остается возможность получить Ваши собственные классы из классов CMapX и добавить Ваши собственные перезагруженные функции-элементы.

OLE Dispatch Driver

C++ классы в MapX.h получаются из класса MFC - COleDispatchDriver. Все свойства и методы в конце концов завершаются вызовом `IDispatch->Invoke()` что бы сообщить MapX ох что надо делать. COleDispatchDriver всегда корректно обрабатывает вызванную `Release()` с указчиками IDispatch возвращенными из свойств или методов, но есть несколько мест, где надо быть внимательными. Когда объект MapX передается событию через интерфейс IDispatch, Вы должны быть уверены, что не освобождаете указчик, потому что объекты передаются событиям, не бывшим в `AddRef()` MapX. Пример:

```
void
CMapxSampleView::OnThemeModifyRequested(LPDISPATCH
Theme)
{
    try {
        CMapXTheme theme;
        COptionVariant vHelpFile, vHelpID;                                // mark
                                                                    // don't have a helpfile

        theme.AttachDispatch(Theme, FALSE); // don't
auto release
        theme.ThemeDlg(vHelpFile, vHelpID);
    }
    catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    }
    catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}
```

```

    }
}

```

Свойства и методы, которые возвращают другие объекты

Вы можете присвоить значение методов или свойств, которые возвращают объекты прямо в переменные того же типа.

Пример

```

CMapXThemes objThemes =
    m_ctrlMapX.GetDatasets()[1L].GetThemes();

```

Пример

```

class CMapXDataset : public COleDispatchDriver
{
public:
    CMapXDataset() {}           // Calls
    COleDispatchDriver default
        constructor
    CMapXDataset(LPDISPATCH pDispatch) :
        COleDispatchDriver(pDispatch) {}
    CMapXDataset(const CMapXDataset& dispatchSrc) :
        COleDispatchDriver(dispatchSrc) {}

    // Attributes
public:
    CString GetName();
    void SetName(LPCTSTR);
    long GetRowCount();
    CMapXFields GetFields();
    CMapXThemes GetThemes();
    long GetGeoField();
    long GetSecondaryGeoField();
    CMapXLayer GetLayer();

```

```

    short GetType();

// Operations
public:
    void Refresh();
    VARIANT GetValue(const VARIANT& Row, const
VARIANT& Column);
    VARIANT GetValue(long Row, long Column)
        { return GetValue(COleVariant(Row),
COleVariant(Column)); }
    VARIANT GetValue(long Row, LPCTSTR Column)
        { return GetValue(COleVariant(Row),
COleVariant(Column)); }

    CMapXSourceRows GetSourceRows(const VARIANT& Row);
    CMapXSourceRows GetSourceRows(long Row)
        { return GetSourceRows(COleVariant(Row)); };

```


Включение файла MapX.cpp в Ваш проект

Включите MapX.cpp и .h файлы в Ваш проект. Они содержат определения класса и внедрения метода для доступа к контролю MapX. Файлы MapX.h и MapX.cpp могут быть найдены в поддиректории [Samples40\CPP](#) той директории, в которой установлен MapX.

Использование Visual C++ Version 4

В меню [Insert](#), выберите [Files Into Project](#). Выберите MapX.cpp в качестве файла который надо вставить.

Внимание: Не выбирайте команду [Insert > Component](#). Выбирая его, Вы создаете .cpp файл, но он будет неполным.

Использование Visual C++ Version 5 или выше

Из меню [Project](#), выберите [Add To Project > Files](#). Выберите MapX.cpp как файл который надо добавить.

Предупреждение! Не выбирайте команду [Project > Components And Controls](#). Делая так, Вы создаете .cpp файл, но он будет неполным.

Создание управления MapX, с помощью C++

Включите управление в представление (view), которое будет содержать его:

```
class CMapxSampleView : public CView
{
    :
    :
    :
protected:
    CMapX  m_ctrlMapX;
    :
    :
}
```

Что бы объявить константу, представляющую управление индексом ID для MapX:

1. Выполните команду [View > Resource Symbols](#).
2. Нажмите NEW.
3. Введите "IDC_MAP" в качестве имени.

Что бы создать обработчики для сообщений WM_SIZE и WM_CREATE в классе Wizard:

1. Выполните команду [View > ClassWizard](#)
2. Выберите Ваш [View](#) класс из списка класса Name.
3. В окне сообщений, щелкните на "WM_CREATE", и нажмите [Add Function](#).
4. Затем, в окне сообщений, выберите "WM_SIZE" и нажмите [Add Function](#).
5. Затем, нажмите [Edit Code](#).

Создайте механизм управления, когда представление (view) создано.

В **CMapXSampView::OnCreate**:

```
// create map with default size
// resize message will cause it to be
// size to the client area of the view
m_ctrlMapX.Create(NULL, WS_VISIBLE,
CRect(0,0,100,100), this,
IDC_MAP);
```

Держите контроль размера в синхронизации с
содержащим окном:

```
// resize the map to be the same size as our
client area
void CMapxSampleView::OnSize(UINT nType, int cx,
int cy)
{
    CView::OnSize(nType, cx, cy);

    if (cx != 0 && cy != 0)
        m_ctrlMapX.MoveWindow(0,0,cx,cy,TRUE);
}
```

Создайте новый обработчик сообщения для сообщения WM_SETFOCUS
как Вы это делали для сообщения WM_CREATE.

В нашем примере, надо убедиться что MapX дает фокусировку
активному окну:

```
void CMapxSampleView::OnSetFocus(CWnd* pOldWnd)
{
    CView::OnSetFocus(pOldWnd);

    m_ctrlMapX.SetFocus();
}
```

Создание разделов меню с помощью C++

Вы можете применять возможности MapX при создании меню для Doc/View который содержит управление MapX. Сошлитесь в меню на то место, где создан шаблон doc (mapxsamp.cpp):

```
CMultiDocTemplate* pDocTemplate;
pDocTemplate = new CMultiDocTemplate(
    IDR_MAPXSATYPE,                //Menu items
    to control MapX
    RUNTIME_CLASS(CMapxSampleDoc),
    RUNTIME_CLASS(CChildFrame),    //custom MDI
    child frame
    RUNTIME_CLASS(CMapxSampleView));

AddDocTemplate(pDocTemplate);
```

Добавьте обработчики для разделов меню MapX. В примере, эти разделы меню определены в секции меню файла mapxsamp.rc. Обратите внимание, что в этом примере мы используем команды меню для быстрой демонстрации интерактивных инструментов работы с картой (**Zoom In**, **Radius Select**, и др.). В более безупречных приложениях, лучше использовать инструменты в виде кнопок пользовательского интерфейса, чем команды меню.

```
"Previous &View",                ID_CONTEXT_PREVIOUSVIEW
"View &Entire Map",              ID_VIEW_VIEWENTIREMAP
SEPARATOR
"&Properties...",                ID_VIEW_PROPERTIES
"&Layer Control...",             ID_VIEW_LAYERCONTROL
```

```

SEPARATOR
"&Toolbar",           ID_VIEW_TOOLBAR
"&Status Bar",        ID_VIEW_STATUS_BAR

"&Arrow",             ID_MAP_TOOL_ARROW
SEPARATOR
"Zoom &In",           ID_MAP_TOOL_ZOOMIN
"Zoom &Out",          ID_MAP_TOOL_ZOOMOUT
"&Pan",               ID_MAP_TOOL_PAN
"&Center",            ID_MAP_TOOL_CENTER
SEPARATOR
"&Select",            ID_MAP_TOOL_SELECT
"&Radius Select",
ID_MAP_TOOL_RADIUSSELECT
"R&ectangle Select",
ID_MAP_TOOL_RECTANGLESELECT

```

После того, как определены эти индексы ID разделов меню, обработчики могут быть добавлены через [Class Wizard](#). В Вашем классе, содержащем управление MapX, создайте обработчики для ID_MAP_TOOL_ARROW, ID_MAP_TOOL_ZOOMIN, и т.д.

```

// tell MapX what the current tool is
void CMapxSampleView::OnMapToolArrow()
{
    m_ctrlMapX.SetCurrentTool(miArrowTool);
}

void CMapxSampleView::OnMapToolZoomin()
{
    m_ctrlMapX.SetCurrentTool(miZoomInTool);
}

// switch to the previous view

```

```
void CMapxSampleView::OnContextPreviousview()  
{  
    m_ctrlMapX.ZoomTo(m_dPrevZoom, m_dPrevX,  
m_dPrevY);  
}
```

После того, как инструмент выбран, действуют встроенные в MapX функциональные обработчики для масштабирования, выбора и др.

Обработка событий MapX с помощью C++

Для обработки событий MapX, Вам надо во-первых, построить карту приема событий (eventsink map) для тех событий, которые Вас интересуют. Константы для идентификации события DISPATCH определены в MapX.h для пользовательских событий MapX, и в <olectl.h> для ряда событий OLE.

```
// From MapX.h
#define MAPX_DISPID_SELECTION_CHANGED 0x1
#define MAPX_DISPID_RESOLVEDATABIND 0x2
#define MAPX_DISPID_TOOLUSED 0x3
#define MAPX_DISPID_REQUESTDATA 0x4
#define MAPX_DISPID_DATAMISMATCH 0x5
#define MAPX_DISPID_MAPVIEWCHANGED 0x6
#define MAPX_DISPID_ANNOTATIONADDED 0x7
#define MAPX_DISPID_ANNOTATIONCHANGED 0x8
#define MAPX_DISPID_THEMEMODIFYREQUESTED 0x9
#define MAPX_DISPID_DRAWUSERLAYER 0x0a
#define MAPX_DISPID_POLYTOOLUSED 0x0b
// From <olectl.h>
#define DISPID_CLICK (-600)
#define DISPID_DBLCLICK (-601)
#define DISPID_KEYDOWN (-602)
#define DISPID_KEYPRESS (-603)
#define DISPID_KEYUP (-604)
#define DISPID_MOUSEDOWN (-605)
#define DISPID_MOUSEMOVE (-606)
#define DISPID_MOUSEUP (-607)
#define DISPID_ERROREVENT (-608)
```

Макрос `ON_EVENT` в `EVENT_SINK` также определяет ID для управления MapX (`IDC_MAP` в примере), параметры для события, и имя метода обработки события.

В заголовке файла View (`MapXSampView.h`), поместите строку `"DECLARE_EVENTSINK_MAP ()"` ниже строки `"DECLARE_MESSAGE_MAP"`.

Из `mapxsampview.cpp`:

```
BEGIN_EVENTSINK_MAP(CMapxSampleView, CView)
    ON_EVENT(CMapxSampleView, IDC_MAP,
DISPID_MOUSEMOVE,
        OnMouseMoveInMap, VTS_I2 VTS_I2 VTS_XPOS_PIXELS
VTS_YPOS_PIXELS)
    ON_EVENT(CMapxSampleView, IDC_MAP,
MAPX_DISPID_MAPVIEWCHANGED,
        OnMapViewChanged, VTS_NONE)
    ON_EVENT(CMapxSampleView, IDC_MAP, DISPID_MOUSEUP,
        OnMouseUpInMap, VTS_I2 VTS_I2 VTS_XPOS_PIXELS
VTS_YPOS_PIXELS)
    ON_EVENT(CMapxSampleView, IDC_MAP,
MAPX_DISPID_TOOLUSED,
        OnToolUsed, VTS_I2 VTS_R8 VTS_R8 VTS_R8 VTS_R8
VTS_R8 VTS_BOOL
        VTS_BOOL VTS_PBOOL)
    ON_EVENT(CMapxSampleView, IDC_MAP,
MAPX_DISPID_THEMEMODIFYREQUESTED,
        OnThemeModifyRequested,
        VTS_DISPATCH)
END_EVENTSINK_MAP()
```

Код обработчика события для события **OnToolUsed** следующий.
Объявление из `mapxsampview.h` такое:


```

    void OnToolUsed(short ToolNum, double X1, double
Y1, double X2,
    double Y2, double Distance, BOOL Shift, BOOL
Ctrl, BOOL*
    EnableDefault);

```

...и внедрение из `mapxsampview.cpp` (мы только выводим параметры в окно отладки используя макрос `TRACE`):

```

void CMapxSampleView::OnToolUsed(short ToolNum,
double X1, double Y1, double X2, double Y2, double
Distance, BOOL Shift, BOOL Ctrl, BOOL*
EnableDefault)
{
    CString str;
    str.Format("Tool=%d, [%f,%f] [%f, %f], dist=%f, %s
%s\n",
        ToolNum, X1,Y1,X2,Y2,Distance,
        (Shift)?"Shift":"", (Ctrl)?"Ctrl":"" );
    TRACE(str);
}

```

Вы нормально обработаете пользовательские инструменты в этой функции, или переопределите поведение MapX по умолчанию для встраивания в инструменты. Пример смотрите в разделе *“Использование пользовательских инструментов”*.

Внимание: Для событий в которых объекты передаются как параметры, обработчик событий не меняет контрольный счет объекта. Пример смотрите в разделе *“Обработка исключительных ситуаций MapX”*.

Применение пользовательских инструментов (C++ пример)

После того как Вы применили метод `CreateCustomTool` для создания пользовательских инструментов, используйте событие `ToolUsed` для выполнения действия когда пользователь применяет собственный инструмент.

Этот пример тестов показывает какой инструмент используется. Затем, в зависимости от того, какой инструмент используется, этот пример также:

- Изменяет стиль объекта карты под курсором, или
- Помещает новый символ там, где пользователь щелкнул мышкой.

```
void CMapxSampleView::OnToolUsed(short ToolNum,
double X1,
    double Y1, double X2, double Y2, double
Distance,
    BOOL Shift, BOOL Ctrl, BOOL* EnableDefault)
{
    CString str;
    CMapXPoint pnt;

    str.Format("Tool=%d, [%f,%f] [%f, %f], dist=%f, %s
%s\n",
        ToolNum, X1,Y1,X2,Y2,Distance,
        (Shift)?"Shift":"", (Ctrl)?"Ctrl":"");
    TRACE(str);
    // change the style of the feature under the
    cursor
    if (ToolNum == MAP_TOOL_CHANGESTYLE) {
        try {
```

```

        // Need the dispatch to use the point
        if (pnt.CreateDispatch(pnt.GetClsid())) {
            pnt.Set(X1, Y1);
        }
        else {
            // something went wrong, can't use the
point...

            AfxThrowOleException(CO_E_CLASS_CREATE_FAILED);
        }

        CMapXLayers        layers =
m_ctrlMapX.GetLayers();
        // Get the USA feature under the cursor
        CMapXFeatures      ftrs =

layers.Item("USA").SearchAtPoint(LPDISPATCH(pnt));
        // work on only the first feature
        CMapXFeature       ftr = ftrs.Item(1);
        // get the style object from the feature
        CMapXStyle         style = ftr.GetStyle();

        style.SetRegionBackColor(255);
        // update the feature in the layer
        ftr.Update();
    }
    catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    }
    catch (COleException *e) {

```

```

        e->ReportError();
        e->Delete();
    }
}

// place a new symbol at the point clicked on
else if (ToolNum == MAP_TOOL_NEWPOINT) {
    try {
        CMapXLayers          layers =
m_ctrlMapX.GetLayers();
        CMapXFeature         ftr;
        // Need the dispatch id to use the feature
        if (ftr.CreateDispatch(ftr.GetClsid())) {
            // Symbol feature
            ftr.SetType(miFeatureTypeSymbol);
            // Get the point object from the feature
            // and call the Set method
            ftr.GetPoint().Set(X1, Y1);
            // Add it to the layer
            layers.Item("USA").AddFeature(ftr);
        }
    }
    else {

AfxThrowOleException(CO_E_CLASS_CREATE_FAILED);
    }
}

catch (COleDispatchException *e) {
    e->ReportError();
    e->Delete();
}

catch (COleException *e) {
    e->ReportError();
}

```

```

        e->Delete();
    }
}
}

```

Связывание данных с использованием C++

Приложение CMapXSample имеет примеры связывания данных для следующих типов наборов данных dataset: **miDataSetDAO**, **miDataSetODBC**, **miDataSetUnbound**, **miDataSetGlobalHandle**.

Примеры так же показывают как использовать динамическое связывание данных, добавляет слой точек XY, сопоставление почтовых кодов ZIP, и автоматическое сравнение.

Смотрите файл **mapxsampview.cpp** для проверки примеров. Ниже имеется один из обработчиков команд меню для добавленных данных из файла использующий тип **miDataSetGlobalHandle**. Формат данных должен быть такой: кавычки вокруг стрингов, табуляторы между полей, и возврат каретки/конец строки в конце записи. Например:

```

"\"NY\" \"t105.34\"t100\"t1\"r\n"
"\"MA\" \"t245.19\"t200\"t2\"r\n"
"\"NY\" \"t195.0\"t300\"t3\"r\n"
"\"AK\" \"t195.0\"t125\"t4\"r\n"
"\"CA\" \"t56.453\"t200\"t5\"r\n";

```

Обратите Внимание, что этот тип связывания данных (**miDataSetGlobalHandle**) является только одним из нескольких типов наборов данных, которые поддерживаются.

Функция `CMapxSampleView::OnMapAdddata()` обрабатывает разделы меню в меню Карта. Она подсказывает имя файла, содержащего данные в форме, указанной выше, читает файл, и добавляет его к коллекции карт набора данных dataset.

```
void CMapxSampleView::OnMapAdddata()
{
    :
    :
    CFileDialog dlgFile(TRUE, "*.txt", NULL, 0,
szDataFilter, this);

    if (dlgFile.DoModal() == IDCANCEL) // User
cancelled the dialog
        return;

    // Read file into a string, and copy it to a
global memory buffer
    :
    :
    // Allocate the memory buffer, copy the string
into it
    :
    :
    // Declare the variables that will be parameters
to
    // DataSets.Add()
    short Type;
    VARIANT SourceData, Name, GeoField,
                                SecondaryGeoField,
    BindLayerName, Fields;
    CString strName= "TestData";
```

```

// set up optional parameters; most will not be used
// Note: you could also use the line
//COptionalVariant SecondaryGeoField;
// instead
SecondaryGeoField.vt = VT_ERROR;
SecondaryGeoField.scode = DISP_E_PARAMNOTFOUND;

// let mapx auto detect geofield
GeoField.vt = VT_ERROR;
GeoField.scode = DISP_E_PARAMNOTFOUND;

// let mapx find which layer to bind to
BindLayerName.vt = VT_ERROR;
BindLayerName.scode = DISP_E_PARAMNOTFOUND;

// use all fields with defaults
Fields.vt = VT_ERROR;
Fields.scode = DISP_E_PARAMNOTFOUND;

// set the name of our dataset
Name.vt = VT_BSTR;
// Remember to SysAlloc() the string; it's going
into a BSTR
Name.bstrVal = strName.AllocSysString();

// set up source data - no error checking on alloc
Type = miDataSetGlobalHandle;
SourceData.vt = VT_I4;
SourceData.lVal = (long)hGlobalData;

```

```

    try {
        // now add the dataset to the datasets
collection
        CMapXDataset ds =
m_ctrlMapX.GetDatasets().Add(Type,
        SourceData, Name, GeoField, SecondaryGeoField,
BindLayerName,
        Fields);
    }
    catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    }
    SysFreeString(Name.bstrVal)
}

```


Добавление быстрого меню с использованием C++

Событие `DISPID_MOUSEUP` позволяет нам добавлять быстрые (открывающиеся при нажатии правой кнопки мышки) меню.

Создание ресурса меню для Вашего быстрого меню

(`IDR_CONTEXTMENU` в приложении-примере.) Обработчик

OnMouseUpInMap выглядит следующим образом:

```
// if right mouse button, display the context menu
BOOL CMapxSampleView::OnMouseUpInMap(
    short Button, short Shift, OLE_XPOS_PIXELS
x, OLE_YPOS_PIXELS y)
{
    if (Button == 2) { // right button
        CMenu menu;           // top-level menu
        CMenu *pMenu=NULL;    // pop-up menu

        // Load the menu resource.
        menu.LoadMenu(IDR_CONTEXTMENU);

        // TrackPopupMenu cannot display the top-
level menu, so get
        // the handle of the first pop-up menu.
        pMenu = menu.GetSubMenu(0);
        if (!pMenu) {
            return TRUE;
        }

        SetMenuDefaultItem(pMenu->m_hMenu,
ID_VIEW_PROPERTIES,
FALSE);
    }
}
```

```

        // Display the floating pop-up menu. Track
the right mouse
        // button on the assumption that this
function is called
        // during WM_CONTEXTMENU processing.
        POINT pt;
        GetCursorPos(&pt);

        pMenu->TrackPopupMenu(TPM_LEFTALIGN |
TPM_RIGHTBUTTON,
        pt.x, pt.y, this, NULL);

        // Destroy the menu.
        menu.DestroyMenu();
    }
    return TRUE;
}

```

Использование встроенных диалогов справочной системы из C++

Использование справочных диалогов нетрудная вещь. Есть раздел меню для вызова диалога MapX [Properties](#). Он может быть подключен к обработчику с помощью ClassWizard; это будет то, что ClassWizard помещает в mapxsamp.cpp:

```
ON_COMMAND(ID_VIEW_PROPERTIES, OnViewProperties)
```

И обработчик:

```
void CMapxSampleView::OnViewProperties()
{
    // easiest way to bring up property page
    try {
        m_ctrlMapX.PropertyPage();
    }
    catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    }
    catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}
```

Внимание: Не рекомендуется включать страницу свойств (property page) в окончательный вариант приложения. Это сильно помогает при написании и отладке приложений, но не предназначено для конечного пользователя, потому что происходит прямой доступ к слишком большому числу свойств и ситуация становится неконфигурируемой. Вместо этого

используйте совокупность диалогов (таких как LayersDlg).

Похожим образом, в диалоге [Управление слоями](#):

```
void CMapxSampleView::OnViewLayercontrol()
{
    try {
        // mark as optional since we don't have a
helpfile
        COptionalVariant vHelpFile, vHelpID;

        CMapXLayers layers =
m_ctrlMapX.GetLayers();
        layers.LayersDlg(vHelpFile, vHelpID);
    }
    catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    }
    catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}
```

Обработка исключительных ситуаций MapX с помощью C++

MapX генерирует сообщение о большинстве ошибок с использованием `COleDispatchException`. Это важно, что Вы перехватываете исключительную ситуацию конда вызывается MapX, потому что по умолчанию обработчик для MFC не перехватывает их, и приложение может закрыться. Это так же очень полезно для отображения описания ошибок в окне ошибок. Текст сообщения об ошибке поможет Вам определить используете ли Вы неправильно объекты MapX, или случились какие другие проблемы.

Класс `COleDispatchException` так же включает код ошибок в доступных элементах `m_wCode`, так что Ваша программа может идентифицировать и обработать различные типы ошибок. Список кодов ошибок и их описание находятся в Приложении *Коды ошибок MapX*.

Если произошла генеральная OLE ошибка, когда Вы пытались вызвать MapX (это может произойти при использовании просроченной версии MapX.h или MapX.cpp) MFC прогоняет `COleException`. Такой подход рекомендуется, если Вы хотите перехватить оба типа ошибок при вызове MapX.

MapX так же может (хотя редко) передавать ошибки в событие `Error`. Это случается только тогда, когда встречаются определенные ошибки во время асинхронного процессинга (типа перерисовки) который непосредственно не вызывался методами и свойствами MapX.

В этом примере, в обработчике событий, куда попадает событие **ThemeModifyRequested**, мы открываем диалог свойства тематика (Theme) который позволяет пользователю изменять цвета тематических объектов, и т.д. Обратите внимание на те случаи, которые сопровождают процесс обработки события.

```
// Note: in objects passed to events, the event
handler
// does not change the reference count
// ie: do not call release on the object
void
CMapxSampleView::OnThemeModifyRequested(LPDISPATCH
Theme)
{
    try {
        CMapXTheme theme;
        COptionalVariant vHelpFile, vHelpID;
        // mark as optional since we don't have a
        helpfile

        theme.AttachDispatch(Theme, FALSE); // don't
        auto release
        theme.ThemeDlg(vHelpFile, vHelpID);
        // could decide to bring up legend dlg here
        instead
        //CMapXLegend leg(theme.GetLegend());
        //leg.LegendDlg(vHelpFile, vHelpID);
    }
    catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    }
}
```

```

        catch (COleException *e) {
            e->ReportError();
            e->Delete();
        }
    }
}

```

Создание карты в диалоге C++

1. Вставьте MapX в диалог.
 - В редакторе ресурсов для шаблона диалога, в том месте где надо вставить MapX, щелкните правой кнопкой мышки и выберите команду меню **'Insert OLE Control'**. Затем выберите MapInfo MapX V4.
2. Создайте отдельный класс CDialog для диалога.
 - Выделите MapX и нажмите **Ctrl+W** что бы активизировать мастер **Class Wizard**. Выберите раздел **Create A New Class** и введите имя диалога (в примере используется 'CSampleDlg'). Нажмите **OK** что бы создать файлы **SampleDlg.h** и **SampleDlg.cpp**.
3. Добавьте CMapX в качестве члена класса диалогов.
 - В первую очередь, поместите **#include "MapX.h"** в файл заголовка диалога, и затем добавьте переменный член для MapX.

Внимание: Не используйте сгенерированные мастером классов (Class Wizard) wrapper-классы для MapX. Они неполные. Используйте файлы MapX.h и MapX.cpp, находящиеся в директории **Sample40\CPP**.

Пример

```
#include "MapX.h"
```

```

////////////////////////////////////
////////

```

```

// CSampleDlg dialog

class CSampleDlg : public CDialog
{
// Construction
public:
    CSampleDlg(CWnd* pParent = NULL); // standard
    constructor

// Dialog Data
   //{{AFX_DATA(CSampleDlg)
    enum { IDD = IDD_MAPXDLG };
    //}}AFX_DATA

    .
    :
    CMapX m_ctrlMapX;
    .
    :
};

```

Использование DDX_Control для связывания Wrapper-класса с управлением MapX

Вставьте строку в метод `DoDataExchange` класса диалогов используя `DDX_Control` следующим образом:

```

void CSampleDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CSampleDlg)
    DDX_Text(pDX, IDC_STATICX, m_strX);
    DDX_Text(pDX, IDC_STATICY, m_strY);
    //}}AFX_DATA_MAP

```



```

    // IDC_MAP1 is the ID given to the mapx control
    when it
    // was inserted into the dialog template.
    DDX_Control(pDX, IDC_MAP1, m_ctrlMapX);
}

```

Использование свойств MapX, методов и объектов для работы с Картой

```

// пример использует mapx для создания временного
// слоя и создания
// пользовательских инструментов, которые будут
// применяться позже.
BOOL CSampleDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    try {
        m_ctrlMapX.SetZoom(1000); // zoom in to 1000
        miles

        // create temporary layer on top for drawing on.
        CMapXLayer layer =
            m_ctrlMapX.GetLayers().CreateLayer("scratch
            layer", NULL, 1);

        // make it the animation layer
        m_ctrlMapX.GetLayers().SetAnimationLayer(layer);

        // create some custom tools for object drawing
        m_ctrlMapX.CreateCustomTool(MYTOOL_SYMBOL,
            miToolTypePoint, miCrossCursor);
    }
}

```

```

        m_ctrlMapX.CreateCustomTool(MYTOOL_LINE,
miToolTypePoly,
        miCrossCursor);
        m_ctrlMapX.CreateCustomTool(MYTOOL_REGION,
miToolTypePoly,
        miCrossCursor);
        m_ctrlMapX.CreateCustomTool(MYTOOL_TEXT,
miToolTypePoint,
        miIBeamCursor);

        m_ctrlMapX.SetCurrentTool(MYTOOL_REGION);
    }
    catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    }
    catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }

    return TRUE; // return TRUE unless you set the
focus to a
                //control. EXCEPTION: OCX Property Pages
should
                // return FALSE
}

```

Использование Мастера Классов (Class Wizard) для обработки событий

Обработка событий MapX в диалоге очень проста. Достаточно использовать в Мастере Классов таблицу ‘Message Maps’, выделить индекс ID для MapX (IDC_MAP1), и откроется список Событий. Выделите нужное событие и нажмите кнопку ‘Add Function’, и Мастер классов добавит обработчик событий и введет EVENT_SINK в ваш класс диалогов автоматически.

17

Распространение Ваших MapX приложений

- ♦ Распространение Ваших MapX приложений
- ♦ Установка MapX приложений
- ♦ Установка MapX ОСХ
- ♦ Установка обработчиков растровых форматов
- ♦ Установка карт и наборов данных Geosets
- ♦ Добавление ключей в реестр Windows
- ♦ Передача строки лицензии в MapX

Продукт MapX, который Вы получаете от MapInfo это developer kit. Он содержит MapX ОСХ, примеры программ, электронную справку, несколько карт-примеров и наборов данных geosets а так же различные утилиты и другие файлы поддержки. Эта глава касается распределения Ваших приложений MapX среди пользователей.

Установка MapX приложений

Когда Вы поставляете Ваши приложения среди покупателей, Вам надо установить несколько различных вещей. Это:

- Созданный вами software.
- MapX control, и связанные с ним файлы поддержки.
- Карты и geosets используемые в приложении.

Идеально было бы встроить установку MapX в вашу процедуру установки, так, чтобы Ваш конечный пользователь не нуждался в дополнительных процедурах установки MapX.

Существуют четыре главных шага установки и распространения приложений MapX:

1. Установите необходимые файлы на компьютер пользователя.
2. Запустите утилиты (такие как regsvr32.exe) для регистрации указанных файлов в системе пользователя.
3. Добавьте указанные ключи в реестр Windows.
4. Передайте в лицензионную строку MapX с клиентским приложением.

Файлы, установленные с MapX могут быть разделены на следующие категории:

Необходимые файлы:

- MapX OCX и связанные с ней библиотеки dlls
- Системные файлы Windows: включая шрифты, MFC и OLE dlls

Дополнительные файлы:

- Драйверы Dataset (используемые для связывания данных)
- Обработчики Import/Export растров
- Карты и наборы данных Geosets

Пример Installshield покажет как устанавливать необходимые файлы при работе с MapX Installation CD. Эта глава будет охватывает процесс установки и необходимых и дополнительных файлов.

Установка MapX ОСХ

Если Вы установили MapX на Ваш компьютер, то знаете, как много файлов использует MapX. В предыдущих версиях MapX, все файлы устанавливались в следующую директорию: `\Program Files\MapInfo MapX\Program`. MapX v4.0 и старше устанавливает программные файлы по следующему пути: `\Program Files\Common Files\MapInfo Shared\MapX Common`. Эта директория рекомендуется для необходимых файлов MapX, тогда как все необходимые файлы для Вашего приложения лучше размещать в специальной директории.

Некоторые из необходимых для MapX файлов являются системными файлами Windows dlls и должны размещаться в директории System. Например, MapX использует **MFC42** и **OLEAUT32 dlls** находящиеся в директории Windows System.

Файлы необходимые для MapX

Системные Windows DLLs

Файлы устанавливаются в директорию `Windows\System`. Проверьте версию перед перемещением этих файлов. Если у Вас старая версия, файлы могут использоваться во время перемещения. В этом случае понадобится перезагрузка.

Mfc42.dll	msvcp60.dll	msvcrt.dll
Olepro32.dll	oleaut32.dll	

Программные файлы MapX

Файлы устанавливаются в папку \Program Files\Common Files\MapInfo Shared\MapX Common. Эти файлы не надо регистрировать.

ALLTYPE.DLL	AllTypeRes.dll	ColLookupSystem.dll
CommandProcessor.dll	CommandProcessorRes.dll	COMPILER.DLL
COORDSYS.DLL	CoordSysRes.dll	CustomProperties.dll
DAENGINE.DLL	DAEngineRes.dll	DBINFO.DLL
DBInfoRes.dll	DBLAYER.DLL	DBLayerRes.dll
ExprPacket.dll	ExprPacketCreator.dll	ExprPacketCreatorRes.dll
ExprPacketRes.dll	FcnInfoServer.dll	FcnInfoServerres.dll
FIND.DLL	FINDRES.DLL	GEO.DLL
Tools.dll	UTILITY.DLL	UtilityRes.dll

Файлы устанавливаются в папку \Program Files\Common Files\MapInfo Shared\MapX Common. Для регистрации используется Regsvr32.exe

- Mapx40.ocx

Файлы устанавливаются в папку Windows\Fonts (для Windows 95 и Windows NT 4.0).

Шрифты также должны регистрироваться с Windows. Если Вы используете ПО третьей фирмы для создания установщика, то такое ПО может обработать для Вас регистрацию шрифтов. С другой стороны, Вы можете зарегистрировать шрифты вручную, вызовом программы Win32 AddFontResource.

ARIAL.TTF	MAPIS__.TTF	MAPSYM.TTF	TTMIAR__.TTF
TTMICG__.TTF	TTMIMI__.TTF	TTMIOG__.TTF	TTMIOS__.TTF
TTMIRE__.TTF	TTMITC__.TTF	TTMIWE__.TTF	

Дополнительные файлы

Поддержка файлов NADCON

Файлы устанавливаются в папку \Program Files\Common Files\MapInfo Shared\MapX Common. Они требуются, если Вам нужна поддержка конвертирования NADCON из системы координат NAD 27 в NAD 83 и обратно.

ALASKA.LAS	ALASKA.LOS	CONUS.LAS
CONUS.LOS	HAWAII.LAS	HAWAII.LOS
PRVI.LAS	PRVI.LOS	TGEORGE.LAS
STGEORGE.LOS	STLRNC.LAS	STLRNC.LOS
STPAUL.LAS	STPAUL.LOS	

Растровые символы

Устанавливаются в папку \Program Files\Common Files\MapInfo Shared\MapX Common\CustSymb.

TOWE1-32.BMP	POLI1-32.BMP	MBOX2-32.BMP
GOLF1-32.BMP	TOWE2-32.BMP	GLOB1-32.BMP
PIN6-32.BMP	MBOX1-32.BMP	

Поддержка растровых изображений

Файлы устанавливаются в папку \Program Files\Common Files\MapInfo Shared\MapX Common.

Поддержка ODBC

Файлы устанавливаются в папку \Program Files\Common Files\MapInfo Shared\MapX Common.

Регистрируются с помощью Regsvr32.exe.

- MODBCDataset.dll
- MMapXColumnInfo.dll

Регистрируются с помощью RegTypLib.exe

- mdatasetint.tlb

Поддержка документов Notes

Файлы устанавливаются в папку `\Program Files\Common Files\MapInfo Shared\MapX Common`.

Регистрируются с помощью Regsvr32.exe.

- MNotesDataset.dll
- MMapXColumnInfo.dll

Регистрируются с помощью RegTypeLib.exe

- mdatasetint.tlb

Программа-пример Installshield устанавливает приведенные выше файлы. Этот пример может быть найден в MapX SDK CD или может быть загружен с MapX website (www.mapx.com).

Драйверы Регистрации файлов и установки Dataset

Как было показано в предыдущей таблице, некоторые файлы должны быть ‘зарегистрированы’, и это осуществляется запуском утилиты regsvr32.exe:

- regsvr32 /s <filespec of mapx.ocx>
- regsvr32 /s <filespec of olepro32.dll>
- regsvr32 /s <filespec of oleaut32.dll>

Если Ваше приложение использует или драйвер ODBC или драйвер данных Notes, Вы должны зарегистрировать **typelibrary** следующим образом:

- regtypelib <filespec of mdatasetint.tlb>

Драйверы dataset позволяют MapX связать данные со слоями карты. Более подробная информация о связывании данных находится в главе: "Размещение данных на карте".

Следующая таблица иллюстрирует различные настройки для связывания данных и все необходимые для этого дополнительные установки:

Тип источника Dataset	Драйверы Dataset	Необходимые установки
Lotus Notes	Mnotesdataset.dll	Драйвер Notes dataset требует, что бы nnotes.dll была размещена в системной папке.
ODBC	Modbcdataset.dll	Odbc32.dll должна быть размещена в системной папке
Delphi v3	MgenDSetDrvr.dll, DSLIBP.DLL	Delphimm.dll и Borlndmm.dll должны быть размещены в системной папке
Delphi v4	MgenDSetDrvr.dll, Dslibp4.dll	Borlndmm.dll должна быть размещена в системной папке
Safe Array	MsafeArrayDataset.dll	
ADO v2.0	MapXADODS.dll	
RDO v2.0	MapXRDODS.dll	
MapInfo ODBC	Miodbc.dll	Odbc32.dll должна быть размещена в системной папке
Oracle 8i Spatial	Mioci.dll	
Oracle Express Objects		

Установка обработчиков растровых форматов

MapX также позволяет разработчику встраивать растровые изображения в его приложения. Приложение MapX может открывать или сохранять различные типы растров. Соответствующие драйверы должны быть включены, если Вы будере распространять приложение. MapX может использовать 1 или несколько различных библиотек для загрузки растровых изображений. Когда MapX загружает растровое изображение, он начинает искать эти DLLs и спрашивает, может ли данный файл быть прочитан этой DLL. Если это так, MapX знает, какая DLL обрабатывает формат этого файла. Обработчики форматов будут называться "**xxxxxxxx.RNx**". Базовая часть имени, соответствует формату раstra. Расширение всегда начинается с букв RN, но может кончатся на любую букву (A-Z). Когда производится поиск обработчика формата, MapX стартует с RNA, затем переходит к RNB, и так до RNZ. Это позволяет MapX определять приоритеты, какой именно обработчик использовать. Например, файлы SPOT нуждаются в проверке перед любым другим форматом, потому что они имеют построчные данные. Расширение обработчика SPOT это RND. Формат обработчиков Halo имеет расширение RNV. Lead Tools имеют расширение RNХ.

LEADTOOLS Win32 Pro обеспечивается ©LEAD Technologies, Inc и библиотеками HALO Imaging фирмы Media Cybernetics и включены в поставку MapX. Вы можете использовать некоторые из них для отображения большинства растров в MapX. LeadTools будет загружать целое растровое изображение в память, каждый раз, когда к нему происходит обращение в MapX. Это означает, что изображение будет долго загружаться, но масштабирование и прокрутка будут быстрее. HALO будет загружать в память только то, что надо отобразить на экране, поэтому загрузка растра проходит быстро, а масштабирование и прокрутка проходят медленно. По умолчанию, первой попыткой будет применение HALO. Вы можете изменить этот порядок, не включая библиотеку HALO в поставку Вашего приложения, или сменить расширение обработчика LeadTools с **.rhx** на **.rhu** или что-нибудь еще, используемое перед буквой “х”. Могут так же встретиться форматы, не поддерживаемые данными библиотеками, и требующие своих библиотек.

Внимание: Все эти растровые обработчики должны устанавливаться в ту же директорию, что и MapX OCX, которая обычно размещается в папке: **\Program Files\Common Files\MapInfo Shared\MapX Common**.

Список обработчиков, включенных в поставку MapX:

Обработчик формата	Требуемые файлы	Поддерживаемые типы файлов
Все типы форматов	Migeoreg.dll, MIRASTER.DLL	
Lead Tools	LTFIL70N.DLL, LTKRN70N.DLL, leadtool.rhx	JPG-LFCMP70N.DLL, GIF-LFGIF70N.DLL, TIF-LFTIF70N.DLL, LFFAX70N.DLL PNG- LFPNG70N.DLL, PSD- LFPSD70N.DLL, WMF-LFWMF70N.DLL,

		BMP-LFBMP70N.DLL
Halo Libraries	Halo.rhv, mihiffl.dll	BMP-miffbmp.dll, GIF-miffgif.dll, JPG-miffjpeg.dll, PCX-miffpcx.dll, TARGA-mifftga.dll, TIF-mifftiff.dll
TIF	TIFF.RHL	Или Вы можете использовать библиотеку TIF поставляемую с Lead Tools или Halo.
Все типы растров	Migeoreg.dll, MIRASTER.DLL	
Lead Tools	LTFIL70N.DLL, LTKRN70N.DLL, leadtool.rhx	JPG-LFCMP70N.DLL, GIF-LFGIF70N.DLL, TIF-LFTIF70N.DLL, LFFAX70N.DLL PNG- LFPNG70N.DLL, PSD- LFPSD70N.DLL, WMF-LFWMF70N.DLL, BMP-LFBMP70N.DLL
Halo Libraries	Halo.rhv, mihiffl.dll	BMP-miffbmp.dll, GIF-miffgif.dll, JPG-miffjpeg.dll, PCX-miffpcx.dll, TARGA-mifftga.dll, TIF-mifftiff.dll
TIF	TIFF.RHL	Или Вы можете использовать библиотеку TIF поставляемую с Lead Tools или Halo.
SPOT	Spot.rhd	

MRSID	Mrsid.rhe	SID-MRSID32.DLL
Все типы Grid	MIGRID.DLL, MIRASTER.DLL	
MapInfo GRID и Hillshading	MIG.GHL	MIG-GRIDDLL.DLL

Установка карт и наборов данных Geosets

Регистрирование наборов данных geoset это удобный путь регистрации каждой таблицы MapInfo ассоциированной с данным geoset в геословаре GeoDictionary. При регистрации таблицы MapInfo в GeoDictionary, эта таблица может использоваться для автоматического связывания. Более подробно об автоматическом связывании смотрите главу: " *Размещение данных на карте*".

Если ситуация такова, что Вам надо использовать GeoDictionary когда регистрируется geoset, GeoDictionaryManager40.exe добавит соответствующие компоненты к Geodictionary (geodict.dct). Если Geodictionary не существует, GeoDictionaryManager40.exe создаст его.

- GeoDictionaryManager40.exe <full filespec (with drive) of geoset>

Этот шаг должен быть сделан после всех установок geosets и ассоциированных таблиц на машине пользователя. Дважды проверьте Ваш geoset что бы убедиться, что необходимые пути проложены для каждой таблицы. Например, если Вы создаете geoset используя таблицы MapInfo, находящиеся в разных местах hard диска, то geoset укажет полный путь к каждой такой таблице. Если Вы будете поставлять такой набор geoset пользователям, то таблицы должны так же размещаться на своих привычных местах. Что бы избежать связанных с этим проблем, скопируйте все таблицы MapInfo в ту же директорию, где создавался geoset. Затем создайте geoset в этой же директории.

Добавление ключей в реестр Windows

MapX так же использует следующие три регистрационных ключа, которые Ваш установщик должен создать в системе конечного пользователя, если они не существовали ранее. MapX создает эти три ключа, когда Вы устанавливаете MapX в Вашей системе. Таким образом, если Вы хотите увидеть эти регистрационные ключи, запустите программу regedit.exe.

Ключ	Описание
HKEY_LOCAL_MACHINE\Software\MapInfo\MapX\4.0 GeoDictionary	Строка Ключ GeoDictionary имеет спецификацию для файла geodictionary. Пример: C:\Program Files\myappdir\Maps\GeoDict.DCT
HKEY_LOCAL_MACHINE\Software\MapInfo\MapX\4.0 SearchPaths	Строка Ключ SearchPaths имеет точку с запятой в качестве разделителя в файле, указывающего, где могут находиться

	файлы карт и наборы данных geosets; по умолчанию там пустая строка ("") если они располагаются в директории, указанной в ключе GeoDictionary.
HKEY_LOCAL_MACHINE\Software\MapInfo\MapX\4.0 CommonDLLDir	Строка Показывает размещение папки, в которой находятся OCX и файлы поддержки. Пример: \Program Files\Common Files\MapInfo Shared\MapX Common

Внимание: Путь GeoDictionary используется при добавлении geoset к объекту MapX без указания полного пути к Geoset и когда используется автоматическое связывание. В добавок, ключ SearchPaths не является необходимым, если не используется GeoDictionary. Посмотрите раздел в этой главе, называемый “Установка карт и Geosets” для определения того, нужны ли эти ключи для Вашего приложения.

Передача строки лицензии в MapX

В порядке инициализации клиентской копии MapX на компьютере Вашего пользователя, строка лицензирования должна передаваться в момент создания объекта при работе приложения.

Внимание: Если Вы используете MapX в диалоге с Visual C++, VisualBasic или Delphi, MapX создается с корректной строкой лицензирования. В этих случаях можете пропустить этот раздел. Таким образом, если Вы создаете управление MapX используя метод `CMapX::Create()` в Visual C++ или используя PowerBuilder, то надо сделать следующие шаги.

Visual C++

Напротив, если используется MapX в диалоге сеанса Visual C++, Вам не надо выполнять этот шаг.

Для успешного создания объекта MapX, Вы должны передать строку лицензирования в качестве параметра `bstrLicKey` в `CMapx::Create()`. Пример ниже показывает как это сделать. Вам надо скопировать первую строку из файла MapX40.lic, который Вы получили вместе с MapX development kit и использовать ее в качестве строки лицензирования ниже.

Внимание: Более подробно о создании приложений MapX с помощью C++ в среде Visual Studio написано в главе : "*Работа с Visual C++*".

Пример на C++

```
BOOL CMapxSampleView::Create(LPCTSTR lpszClassName,
LPCTSTR lpszWindowName, DWORD dwStyle, const RECT&
rect, CWnd* pParentWnd, UINT nID, CCreateContext*
pContext)
{
```

```

        if (CWnd::Create(lpszClassName, lpszWindowName,
dwStyle, rect, pParentWnd, nID, pContext) == FALSE)
        {
            return FALSE;
        }

        try {
            // create map with default size - resize message
will cause it to be
            // sized to the client area of the view
            // pass a license key so the .lic file is not
needed
            //Внимание: Пожалуйста, используйте первую
строку из Вашего файла MapX30.lic на месте
нижеследующей строки.
            // Строка ниже работать не будет
            CString strLic = _T("uQnZi2sFw22L0-MRa8pYX-
1E4P9413-5N7M4407-3C122214-77777-9999");
            BSTR bstrLic = strLic.AllocSysString();
            BOOL b = m_ctrlMapX.Create(NULL, WS_VISIBLE,
CRect(0,0,100,100), this, 100, NULL, FALSE,
bstrLic);
            ::SysFreeString(bstrLic);

            if (!b) {
                return FALSE;
            }

            // теперь Вы можете работать с MapX
m_ctrlMapX.GetTitle().SetVisible(FALSE);
m_ctrlMapX.SetCurrentTool(miZoomInTool);

```

```

    }
    catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    }
    catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }

    return TRUE;
}

```

Приложение А: Управление данными MapX

В дополнение к программному обеспечению MapX, установщик MapX поместит карты и демографические данные) на Ваш компьютер и Вы сразу можете начать работу с MapX.

Внимание: Используйте поставляемые данные в соответствии с лицензионным соглашением [MapInfo Standard License Agreement - Software & Data](#).

По умолчанию, карты и геонаборы устанавливаются в директории:

- [C:\Program Files\MapInfo MapX 4.0\Maps](#)

Демографические данные, в форме фабрики Microsoft Access (MAPSTATS.MDB), устанавливаются в директории:

- [C:\Program Files\MapInfo MapX 4.0\Data](#)

Точный список файлов, установленных на Ваш компьютер зависит от того, какие настройки при установке Вы выбрали. (По умолчанию, все данные устанавливаются; но установщик позволяет не устанавливать определенные данные и геонаборы.) Список установленных файлов от того какая версия программного обеспечения установлена; установка MapX CD обеспечивает большим количеством данных чем загрузка оценочной версии.

Используйте [MapX GeoDictionary Manager](#) для регистрации информации о таблицах MapInfo которые могут сравниваться с MapX во время автоматической связи данных. По умолчанию, эта утилита расположена по следующему пути:

- [C:\Program Files\Common Files\MapInfo Shared\MapX Common\GeoDictionaryManager40.EXE](#)

Используйте утилиту [MapX Geoset Manager](#) для создания и управления геонаборами. По умолчанию, эта утилита размещена по следующему пути:

- [C:\Program Files\Common Files\MapInfo Shared\MapX Common\GeosetManager40.EXE](#)

Для более подробной информации о Geodictionary Manager, смотрите Приложение С: *Использование утилиты GeoDictionary Manager*.

Источники данных

Ниже приводятся данные и их источники, поставляемые с MapInfo MapX. Более полная информация о продуктах MapInfo находится в разделе “*MapInfo Data Products Catalog*” или обратитесь прямо в MapInfo Corporation.

Имя набора данных	Copyright
Japan Geoset	
Japan Cased Roads	GisNET data licensed to MapInfo by GISdata Limited. © GDC Ltd 1993
Japan Cities	MapInfo from Digital Chart of the World
Japan Country background	GisNET data licensed to MapInfo by GISdata Limited. © GDC Ltd 1993
Japan Highways	GisNET data licensed to MapInfo by GISdata Limited. © GDC Ltd 1993
Japan Major Cities	MapInfo from Digital Chart of the World
Japan Rivers and Lakes	GisNET data licensed to MapInfo by GISdata Limited. © GDC Ltd 1993
World Ocean (Lat / Long)	MapInfo
Australia Geoset	
Australia Cities	MapInfo from Digital Chart of the World
Australia Highways	MapInfo from Digital Chart of the World
Australia Major Cities	MapInfo from Digital Chart of the World
Australia State Boundaries	MapInfo
Australia State Capital Cities	MapInfo from Digital Chart of the World
World Ocean (Lat / Long)	MapInfo

Europe Geoset	
Asia	MapInfo from Digital Chart of the World
European Capitals	MapInfo from Digital Chart of the World
European Cities	MapInfo from Digital Chart of the World
European Country Boundaries	Data copyright and produced by URPI 1998 and GisNET data licensed to MapInfo by GISdata Limited. © GDC Ltd 1993
European Highways	GisNET data licensed to MapInfo by GISdata Limited. © GDC Ltd 1993
European Major Cities	MapInfo from Digital Chart of the World
European NUTS 1 Level Boundaries	Copyright URPI 1998
European NUTS 2 Level Boundaries	Copyright URPI 1998
World Ocean (Lat / Long)	MapInfo
France Geoset	
European Country Boundaries	Data copyright and produced by URPI 1998 and GisNET data licensed to MapInfo by GISdata Limited. © GDC Ltd 1993
France Cities	MapInfo from Digital Chart of the World
France Highway Map	GisNET data licensed to MapInfo by GISdata Limited. © GDC Ltd 1993
France Major Cities	MapInfo from Digital Chart of the World
France NUTS 2 Level Administrative Boundaries	Copyright URPI 1998
World Ocean (Lat / Long)	MapInfo
UK Geoset	
European Country Boundaries	Data copyright and produced by URPI 1998 and GisNET data licensed to MapInfo by GISdata Limited. © GDC Ltd 1993
United Kingdom Cities	MapInfo from Digital Chart of the World
United Kingdom Class A Roads	GisNET data licensed to MapInfo by GISdata Limited. ©

	GDC Ltd 1993
United Kingdom Motorways	GisNET data licensed to MapInfo by GISdata Limited. © GDC Ltd 1993
United Kingdom Standard Regions	Copyright URPI 1998
World Ocean (Lat / Long)	MapInfo
Canada Geoset	
Canada Cities	MapInfo from Digital Chart of the World
Canada Highways	MapInfo from Digital Chart of the World
Canada Major Cities	MapInfo from Digital Chart of the World
Canadian Province Boundaries	MapInfo from Digital Chart of the World
Canadian Province Capital Cities	MapInfo from Digital Chart of the World
US State Boundaries	MapInfo from Digital Chart of the World
World Ocean (Lat / Long)	MapInfo
US Geoset	
Canadian Province Boundaries	MapInfo from Digital Chart of the World
Mexico State Boundaries	MapInfo from the Bureau of Transportation Statistics
US Cities	MapInfo from Digital Chart of the World
US County Boundaries	MapInfo from Census Bureau
US Highways	MapInfo from the Bureau of Transportation Statistics
US Major Cities	MapInfo from Digital Chart of the World
US State Boundaries	MapInfo from Digital Chart of the World
US State Capitals	MapInfo from Digital Chart of the World
US top 20 Cities	MapInfo from Digital Chart of the World
World Ocean (Lat / Long)	MapInfo
World Geoset	
World Capitals	MapInfo from Digital Chart of the World
World Countries	MapInfo from Digital Chart of the World
World Graticule	MapInfo
World Ocean (Robinson)	MapInfo
World Top 25 Cities	MapInfo from Digital Chart of the World

DC Geoset	
Dc Area Landmarks	MapInfo
Dc City Boundaries	MapInfo
DC Highways	MapInfo from the Bureau of Transportation Statistics
DC Interstate Roads	MapInfo from the Bureau of Transportation Statistics
DC Landmarks	MapInfo
DC Roads	MapInfo from the Bureau of Transportation Statistics
DC State Roads	MapInfo from the Bureau of Transportation Statistics
Dc Water Layer	MapInfo
DC Zips	© 1995 Geographic Data Technology, Inc.
Asia Geoset	
Asia	MapInfo from Digital Chart of the World
Asia Capitals	MapInfo from Digital Chart of the World
Asia Major Cities	MapInfo from Digital Chart of the World
European Country Boundaries	Data copyright and produced by URPI 1998 and GisNET data licensed to MapInfo by GISdata Limited. © GDC Ltd 1993
World Ocean (for Asia)	MapInfo
Mid-Atlantic Geoset	
Mid-Atlantic Capitals	MapInfo from Digital Chart of the World
Mid-Atlantic Cities	MapInfo from Digital Chart of the World
Mid-Atlantic Counties	MapInfo
Mid-Atlantic Highway	MapInfo from the Bureau of Transportation Statistics
Mid-Atlantic Major Cities	MapInfo from Digital Chart of the World
Mid-Atlantic States	MapInfo from Digital Chart of the World
World Ocean (Lat / Long)	MapInfo
US State Boundaries	MapInfo from Digital Chart of the World
China Geoset	
Asia	MapInfo from Digital Chart of the World
Asia Major Cities	MapInfo from Digital Chart of the World

China Cities	MapInfo from Digital Chart of the World
China Country Bdy	MapInfo from Digital Chart of the World
China Highways	MapInfo from Digital Chart of the World
China Major Cities	MapInfo from Digital Chart of the World
World Ocean (for Asia)	MapInfo
Mexico Geoset	
Mexico Cities	MapInfo from Digital Chart of the World
Mexico Highways	MapInfo from Digital Chart of the World
Mexico Major Cities	MapInfo from Digital Chart of the World
Mexico State Boundaries	MapInfo from the Bureau of Transportation Statistics
Mexico State Capital Cities	MapInfo from Digital Chart of the World
US State Boundaries	MapInfo from Digital Chart of the World
World Ocean (Lat / Long)	MapInfo
North American Geoset	
Canada Cities	MapInfo from Digital Chart of the World
Canada Highways	MapInfo from Digital Chart of the World
Canada Major Cities	MapInfo from Digital Chart of the World
Canadian Province Boundaries	MapInfo from Digital Chart of the World
Canadian Province Capital Cities	MapInfo from Digital Chart of the World
Mexico Cities	MapInfo from Digital Chart of the World
Mexico Highways	MapInfo from Digital Chart of the World
Mexico Major Cities	MapInfo from Digital Chart of the World
Mexico State Boundaries	MapInfo from the Bureau of Transportation Statistics
Mexico State Capital Cities	MapInfo from Digital Chart of the World
US Cities	MapInfo from Digital Chart of the World
US Highways	MapInfo from the Bureau of Transportation Statistics
US Major Cities	MapInfo from Digital Chart of the World
US State Boundaries	MapInfo from Digital Chart of the World
US State Capitals	MapInfo from Digital Chart of the World

World Ocean (Lat / Long)	MapInfo
Germany Geoset	
European Country Boundaries	Data copyright and produced by URPI 1998 and GisNET data licensed to MapInfo by GISdata Limited. © GDC Ltd 1993
Germany Cities	MapInfo from Digital Chart of the World
Germany Highways	GisNET data licensed to MapInfo by GISdata Limited. © GDC Ltd 1993
Germany Major Cities	MapInfo from Digital Chart of the World
Germany NUTS 2 Level Bdys	MapInfo from Digital Chart of the World
World Ocean (Lat / Long)	MapInfo
Italy Geoset	
European Country Boundaries	Data copyright and produced by URPI 1998 and GisNET data licensed to MapInfo by GISdata Limited. © GDC Ltd 1993
Italy Cities	MapInfo from Digital Chart of the World
Italy Highways	GisNET data licensed to MapInfo by GISdata Limited. © GDC Ltd 1993
Italy Major Cities	MapInfo from Digital Chart of the World
Italy NUTS 2 Level Bdys	Copyright URPI 1998
World Ocean (Lat / Long)	MapInfo
Portugal Geoset	
European Country Boundaries	Data copyright and produced by URPI 1998 and GisNET data licensed to MapInfo by GISdata Limited. © GDC Ltd 1993
Portugal Highways	GisNET data licensed to MapInfo by GISdata Limited. © GDC Ltd 1993
Portugal Major Cities	MapInfo from Digital Chart of the World
Portugal NUTS 2 Level Boundary	Copyright URPI 1998
World Ocean (Lat / Long)	MapInfo

Argentina Geoset	
Argentina Cities	MapInfo from Digital Chart of the World
Argentina Country Boundary	MapInfo from Digital Chart of the World
Argentina Major Cities	MapInfo from Digital Chart of the World
South America Country Boundaries	MapInfo from Digital Chart of the World
World Ocean (Lat / Long)	MapInfo
Brazil Geoset	
Brazil Major Cities	MapInfo from Digital Chart of the World
Brazil Cities	MapInfo from Digital Chart of the World
Brazil Country Boundary	MapInfo from Digital Chart of the World
South America Country Boundaries	MapInfo from Digital Chart of the World
World Ocean (Lat / Long)	MapInfo
Israel Geoset	
Africa Country Boundary	MapInfo from Digital Chart of the World
Asia	MapInfo from Digital Chart of the World
Israel Cities	MapInfo from Digital Chart of the World
Israel Country Boundary	MapInfo from Digital Chart of the World
Israel Major Cities	MapInfo from Digital Chart of the World
World Ocean (Lat / Long)	MapInfo
South Korea Geoset	
Asia	MapInfo from Digital Chart of the World
South Korea Cities	MapInfo from Digital Chart of the World
South Korea Country Boundary	MapInfo from Digital Chart of the World
South Korea Major Cities	MapInfo from Digital Chart of the World
World Ocean (for Asia)	MapInfo
Spain Geoset	
European Country Boundaries	Data copyright and produced by URPI 1998 and GisNET data licensed to MapInfo by GISdata Limited. © GDC Ltd 1993
Spain Cities	MapInfo from Digital Chart of the World

Spain Highways	GisNET data licensed to MapInfo by GISdata Limited. © GDC Ltd 1993
Spain Major Cities	MapInfo from Digital Chart of the World
Spain NUTS 2 Boundaries	Copyright URPI 1998
World Ocean (Lat / Long)	MapInfo
India Geoset	
Asia	MapInfo from Digital Chart of the World
India Capital Cities	MapInfo from Digital Chart of the World
India District Bdys	Risk Management Solutions, Inc
India Major Cities	MapInfo from Digital Chart of the World
India Minor Cities	MapInfo from Digital Chart of the World
India State Bdys	Risk Management Solutions, Inc
World Ocean (for Asia)	MapInfo
US Detail Geoset	
Canadian Province Boundaries	MapInfo from Digital Chart of the World
Landmark Map	MapInfo from USGS
Mexico State Boundaries	MapInfo from the Bureau of Transportation Statistics
US Cities	MapInfo from Digital Chart of the World
US County Boundaries	MapInfo from Census Bureau
US Highways	MapInfo from the Bureau of Transportation Statistics
US Major Cities	MapInfo from Digital Chart of the World
US State Boundaries	MapInfo from Digital Chart of the World
US State Capitals	MapInfo from Digital Chart of the World
US top 20 Cities	MapInfo from Digital Chart of the World
US Zipcode Boundaries	© 1997 Geographic Data Technology, Inc.
World Ocean (Lat / Long)	MapInfo
World Detail Geoset	
World Capitals	MapInfo from Digital Chart of the World
World Countries	MapInfo from Digital Chart of the World
World Graticule	MapInfo

World Major Cities	MapInfo from Digital Chart of the World
World Minor Cities	MapInfo from Digital Chart of the World
World Ocean (Robinson)	MapInfo
World Top 25 Cities	MapInfo from Digital Chart of the World
Dallas, TX Geoset	
Crime County Map	MapInfo
Crime Demo Map	MapInfo
Dallas City Boundary A	MapInfo from USGS
Dallas City Boundary B	MapInfo from USGS
Dallas Locations	MapInfo
Dallas Major Highways A	MapInfo from USGS
Dallas Major Highways B	MapInfo from USGS
Dallas Streets A	MapInfo from USGS
Dallas Streets B	MapInfo from USGS
Dallas Streets C	MapInfo from USGS
Dallas Water Bodies	MapInfo from USGS
Dallas Water Rivers	MapInfo from USGS
Dallas, TX Raster	Copyright © 1997, Horizons Technology, Inc
Other Files	
Dallas City Boundary from MapXsite Streets 1.0	MapInfo from USGS
Dallas Major Highways from MapXsite Streets 1.0	MapInfo from USGS
Dallas Streets from MapXsite Streets 1.0	MapInfo from USGS
Dallas Water Bodies from MapXsite Streets 1.0	MapInfo from USGS
Dallas Water Rivers from MapXsite Streets 1.0	MapInfo from USGS
US 5-Digit Zipcode Points	(c) 1999 Geographic Data Technology, Inc.
US 5-Digit Zipcode Points (compressed	(c) 1995 Geographic Data Technology, Inc.

point file)	
MapStats.mdb	
Demographics for Asia	Copyright United Nation Demographic Yearbook, 1994
Demographics for Australia	MapInfo Australia
Demographics for DC	copyright 1998, The Polk Company, All Rights Reserved
Demographics for Mid-Atlantic States	copyright 1998, The Polk Company, All Rights Reserved
Demographics for the World	Copyright United Nation Demographic Yearbook, 1994
Demographics for US	copyright 1998, The Polk Company, All Rights Reserved
US County Age Demographics	copyright 1998, The Polk Company, All Rights Reserved
US County Age Demographics by Gender	copyright 1998, The Polk Company, All Rights Reserved
US County Household demographics	copyright 1998, The Polk Company, All Rights Reserved
US County Household demographics by Age, Income	copyright 1998, The Polk Company, All Rights Reserved
US County Household Income	copyright 1998, The Polk Company, All Rights Reserved
US County Housing Values	copyright 1998, The Polk Company, All Rights Reserved
US County Population Demographics	copyright 1998, The Polk Company, All Rights Reserved
US Customer Database	MapInfo

Приложение В: Утилита Geoset Manager

Используйте утилиту Geoset Manager что бы иметь быстрый и удобный доступ к коллекции слоев карты и их настройкам. Рабочие наборы (или геонаборы, Geosets) помогают Вам избегать операций, занимающих много времени, таких как открытие слоев и отображение их каждый раз, кагда Вам надо с ними работать.

Компьютерные карты организованы в слои. Представьте, что это все карты, нарисованные на прозрачной пленке и располагающиеся одна над другой. Начало запуска геонабора состоит в открытии карты.

Каждый слой карты состоит из различных объектов, таких как полигоны, точки, линии и текст.

Например, один слой может содержать границы штатов, второй слой может содержать символы, обозначающие столицы и третий слой может состоять из текстовых подписей. Накладывая один слой поверх другого, Вы получите целую карту.

После того как геонабор создан, можно настраивать тот порядок, в котором отображаются слои, добавлять, удалять или переупорядочивать их.

Из утилиты Geoset Manager осуществляется работа с геонаборами, или создание собственных геонаборов.

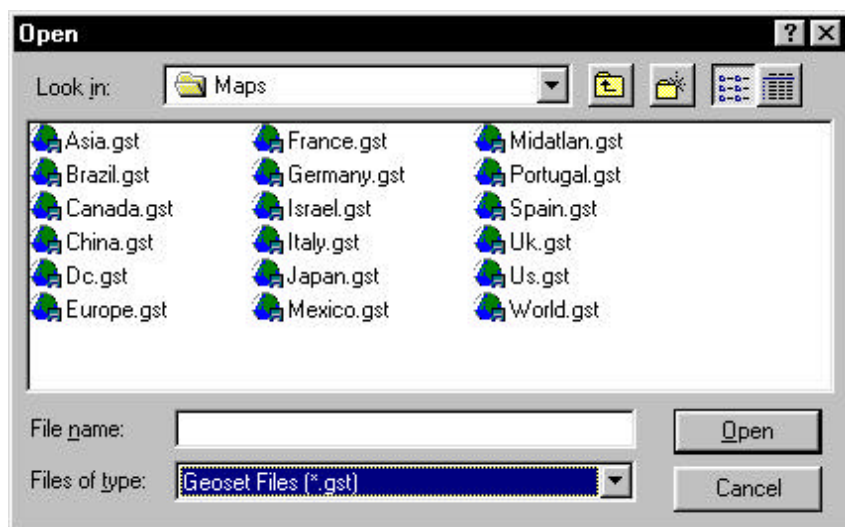
Что бы запустить утилиту Geoset Manager выберите [Geoset Manager](#) из [MapX Program Group](#) в стартовом меню.

Внимание: В Trial-версии MapX, Geoset Manager будет функционировать на машине разработчика.

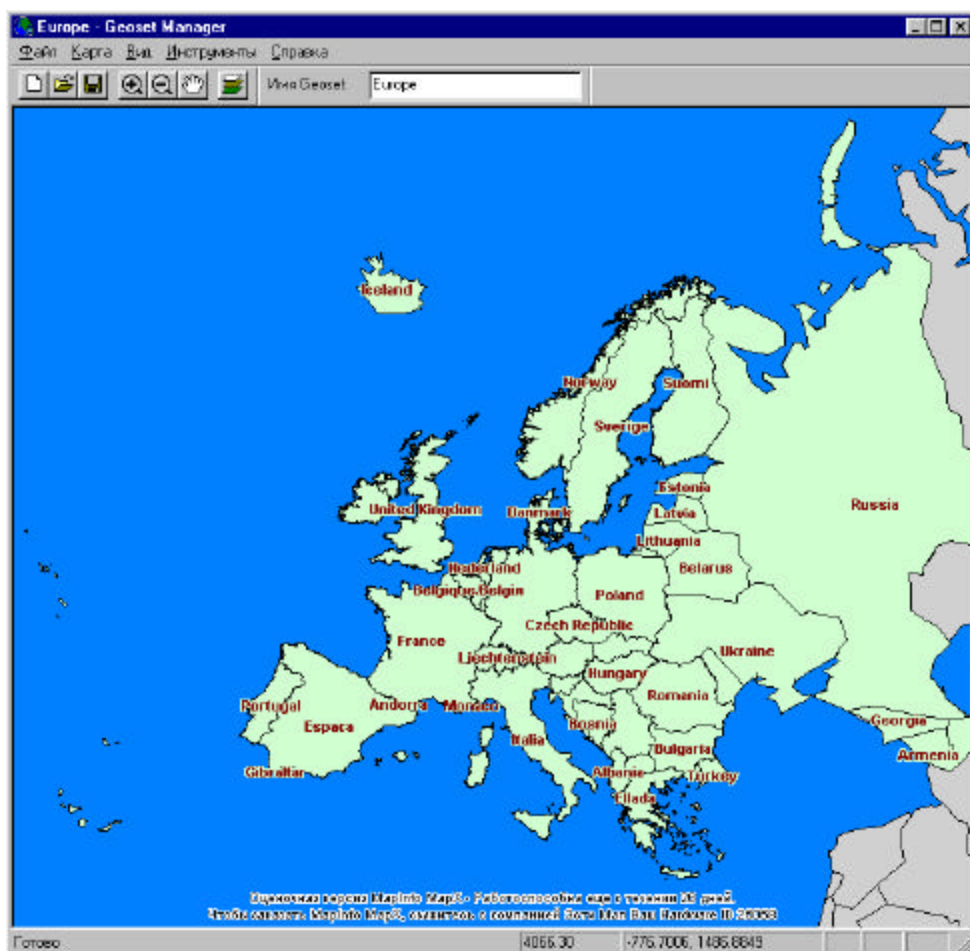
Открытие существующего геонабора

MapX включает несколько примеров геонаборов. Используйте их как стартовую точку для Вашего нового геонабора или используйте их такими какие они есть. Что бы открыть существующий геонабор:

1. Выполните [Файл > Открыть геонабор](#). Откроется диалог [Открыть](#).



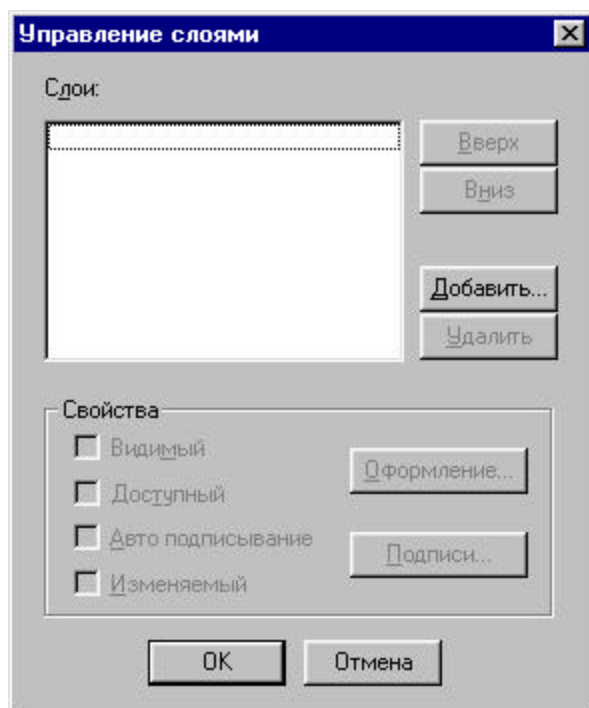
2. Выберите геонабор, который Вам нужен и щелкните на кнопке **Открыть**. Откроется геонабор.



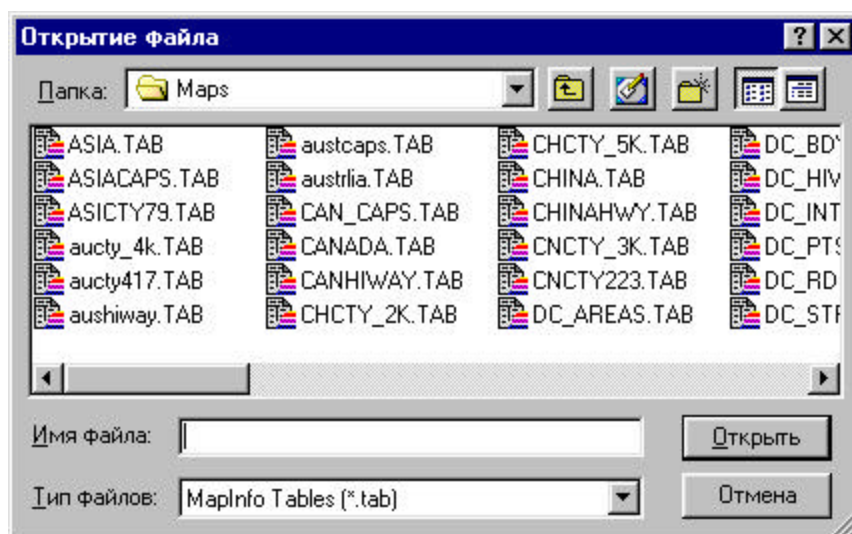
Из этого места Вы можете изменять настройку различных слоев используя команды меню **Карта** и меню **Вид**. Вы можете также вставлять другие наборы, чтобы показывать их в данном геонаборе. Затем сохраните эти новые настройки существующего геонабора или **Сохраните как**, что бы создать новый геонабор. Подробнее мы это обсудим в следующем разделе.

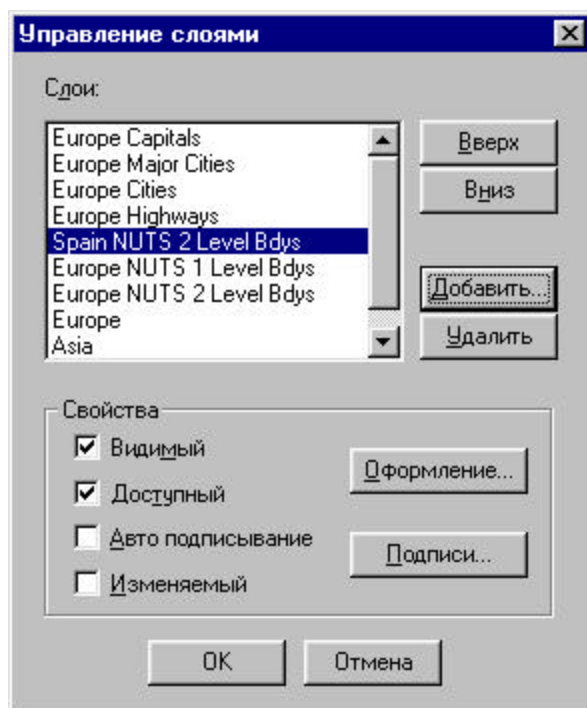
Создание геонабора

1. Выполните команду **Файл > Новый геонабор**. Откроется диалог **Управление слоями**.

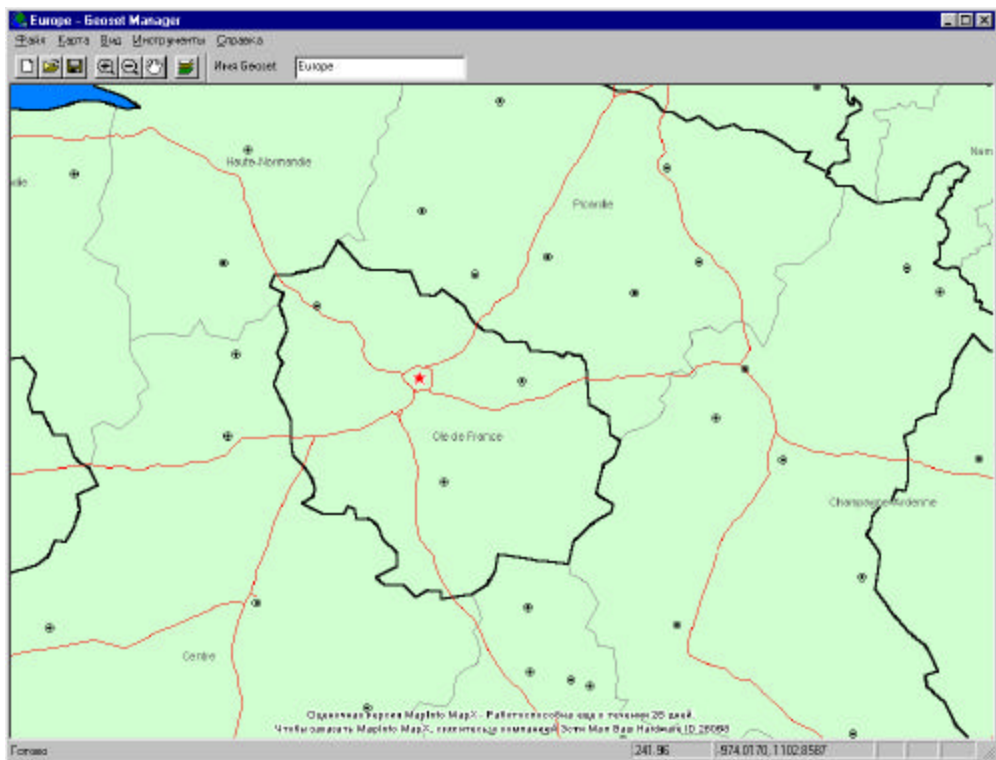


2. Нажмите кнопку **Добавить** что бы открыть диалог **Открыть**.





3. Выберите слои, которые Вы хотите использовать как часть геонабора. Нажмите клавише **Ctrl** пока производите выбор, что бы выделить несколько слоев. Выбранные слои покажутся в диалоге **Управление слоями**.



Начиная с этого места можно настраивать подписи к слоям, менять порядок расположения слоев, добавлять или удалять слои, устанавливать видимость слоев, автоматическое подписывание и возможность делать выборки со слоя. Смотрите описание диалога [Управление слоями](#) в следующем разделе. Вы можете редактировать эти свойства после отображения первоначального геонабора geoset.

4. Нажмите **ОК** что бы показать геонабор geoset. Покажется только что созданный рабочий набор.

Используйте информацию из этого раздела что бы манипулировать слоями и настраивать слои в геонаборах geoset. Когда Вы завершите создание Вашего геонабора, необходимо его сохранить.

Что бы сохранить Ваш геонабор geoset:

1. Выполните **Файл > Сохранить**. Откроется диалог [Сохранить как](#).



2. Введите имя и нажмите кнопку **Сохранить** что бы записать слой с определенными атрибутами.

Действия со слоями при помощи меню Карта

Используйте команды, доступные из меню Карта что бы контролировать способы изображения карты.

Увеличить

Используйте **Увеличивающую лупу** чтобы рассмотреть конкретную область на карте или слое.

1. Выполните команду **Карта > Увеличить**, или нажмите на инструмент на панели инструментов, или нажмите правую кнопку мышки в окне **Geoset Manager** и из быстрого меню выберите команду **Увеличить**.

Появится курсор **Увеличить**.

2. Щелкните увеличивающим курсором на центре области, которую Вы хотите увеличить, произойдет автоматическое увеличение в два раза. Точка на которой Вы щелкните, будет центром увеличенной карты. Повторите эту процедуру столько раз, сколько потребуется.

Что бы увеличить прямоугольную область:

1. Выполните **Карта > Увеличить**, или щелкните на кнопке **Увеличить** на инструментальной панели, или в щелкните правой кнопкой в окне Geoset manager и выберите из быстрого меню **Увеличить**.

Появится увеличивающий курсор.

2. Выделите курсором прямоугольник на карте или отчете. Область внутри прямоугольника будет увеличена.

Уменьшение

Используйте инструмент **Уменьшить**, что бы увидеть более широкую область на карте или слое.

1. Выполните команду **Карта > Уменьшить**, или щелкните на кнопке **Уменьшить** на инструментальной панели, или в щелкните правой кнопкой в окне **Geoset manager** и выберите из быстрого меню **Уменьшить**.

Появится уменьшающий курсор.

2. Выделите курсором прямоугольник на карте или отчете. Область внутри прямоугольника будет уменьшена. Точка на которой Вы щелкните, будет центром уменьшенной карты. Повторите эту процедуру столько раз, сколько потребуется.

Чтобы уменьшить в прямоугольной области:

1. Выполните команду **Карта > Уменьшить**, щелкните на кнопке **Уменьшить**, или щелкните правой кнопкой мышки в окне **Geoset Manager** и из быстрого меню выберите **Уменьшить**.

Появится уменьшающий курсор.

2. Выделите уменьшающим курсором прямоугольник на карте или слое. Выделенная область уменьшится и на карте Вы увидите более обширный район.

Сдвиг

Используйте **Сдвиг** что бы поменять положение карты в окне.

Что бы переместить изображение карты:

1. Выполните команду **Карта > Сдвиг**, или нажмите кнопку **Сдвиг** или щелкните правой кнопкой мышки в окне Geoset Manager и из быстрого меню выберите **Сдвиг**.

Появится сдвигающий курсор.

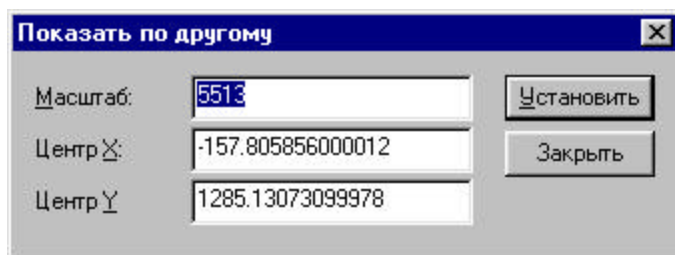
2. Щелкните в области карты.
3. Держите нажатой левую кнопку мышки и перемещайте карту в нужном Вам направлении. Когда Вы отпустите кнопку, Geoset Manager перерисует карту в новом положении.

Показать по-новому

Эта команда устанавливает координаты центра окна X и Y и масштаб.

Выполните команду **Карта > Показать по-новому**.

Откроется диалог **Показать по-новому**.



Введите нужные координаты и масштаб.

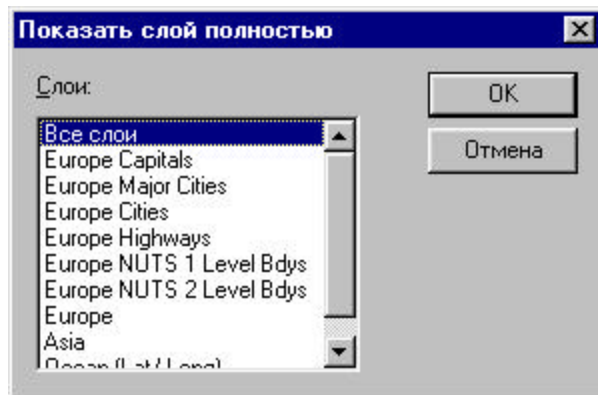
Показать слой полностью

Используйте команду **Показать слой полностью** чтобы показать слой или слои целиком в окне карты.

Используйте эту команду если карта содержит слои, покрывающие различные территории. Например, Ваша карта содержит графства штата Нью-Йорк, дороги, ZIP коды и улицы для города Utica. Если Вы выберете Все слои, масштаб изменится так, что бы показать в окне все слои карты. Но если надо посмотреть только улицы города Utica выберите слой с улицами города Utica. Масштаб изменится так, что бы все улицы поместились в окне.

Что бы показать целиком карту или отдельный слой:

1. Выполните команду **Карта > Показать слой полностью** или нажмите правую кнопку мышки в окне Карты и из быстрого меню выберите **Показать слой полностью**. Откроется диалог **Показать слой полностью**.



2. Выберите нужный слой или **Все слои**. Нажмите ОК что бы отобразить слой (слои).

Управление слоями

Используйте диалог **Управление слоями** для того что бы:

- Изменить отображение слоев карты в активном окне

- Определить, какие слои отобразятся, какие будут удалены, будут выделяемыми, масштабируемыми и подписываемыми
- Изменять порядок расположения слоев.

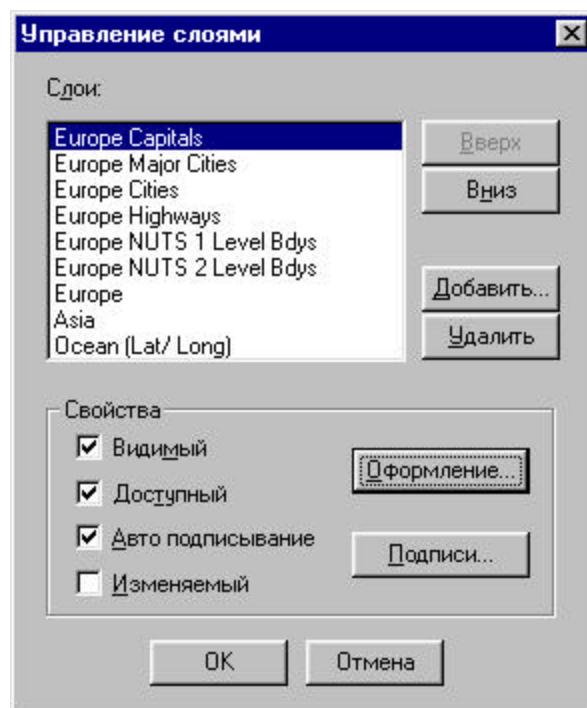
Что бы получить доступ к [Управлению слоями](#):

- Выполните команду [Карта > Управление слоями](#), нажмите кнопку [Управлению слоями](#) или щелкните правой кнопкой мышки на карте и выберите из быстрого меню [Управлению слоями](#) или создание нового геонабора geoset.

Диалог [Управлению слоями](#) контролирует отображение карты на экране.

Здесь находится следующая информация:

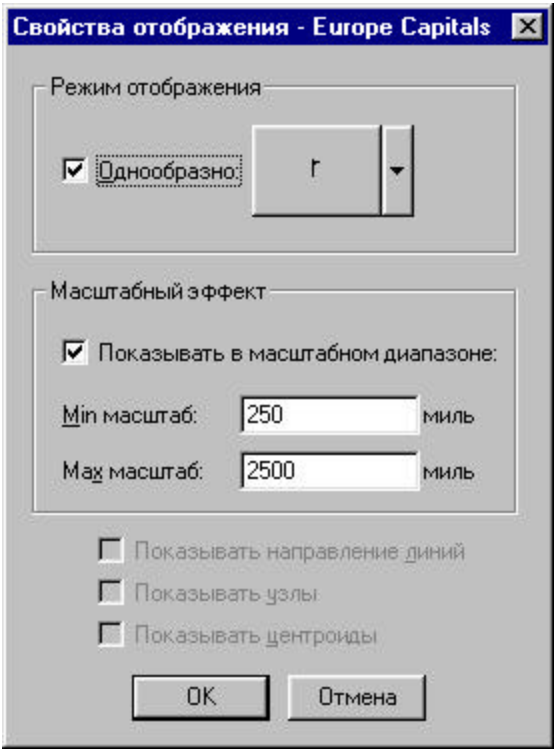
- Диалог [Управлению слоями](#) показывает список слоев в текущей окне карты и дает информацию о том, какой слой является активным, редактируемым или с отображенными автоматичеки подписями.
- Слои представляют собой таблицы, растровые изображения или тематические слои.
- Порядок слоев в диалоге [Управлению слоями](#) соответствует порядку слоев на карте. Например, когда слои с границами располагаются ниже слоя с точками, то точки на карте будут видны.
- Для работы со слоем выберите слай и щелкните левой кнопкой мышки. Установите или снимите флажки видимости, редактируемости и автоматического подписывания.



Up	Перемещает слой (слои) вверх.
Down	Перемещает слой (слои) вниз.
Add	Добавляет один или более слоев к карте. Выберите из открывающегося списка доступных таблиц.
Remove	Удаляет один или более слоев из карты.
Visible	Указывает, является ли слой видимым. Установите флажок для слоя или слоев которые хотите сделать видимыми. Иногда может понадобиться включить слои в геонабор для вычислительных целей а не для того что бы их показывать.
Selectable	Установите флажок что бы сделать слой выбранным (что бы можно было выбирать с него объекты. Слои должны быть выбранными, что бы выбирать отдельные объекты или разместить подписи. Одновременно могут быть выбранными несколько слоев. Одновременно можно

	выбирать объекты только с одного слоя.
Automatic Lables	Установите флажок что бы подписывать карту автоматически. Подписи берутся из колонки которая выбирается в определенном диалоге.
Display	Открывает диалог Display Options. Используйте его что бы задать атрибуты для слоев карты.
Labels	Открывает диалог Label Properties. Используйте этот диалог для задания атрибутов подписей к объектам карты.

Что бы задать атрибуты для слоя карты, нажмите кнопку [Показать](#) в диалоге [Управление слоями](#).



Оформление

Единообразно	Установите флажок чтобы присвоить слою стиль символа по умолчанию.
Кнопка Стиль	Кнопка активна когда установлен флажок Единообразно . Эта

	кнопка применима к выбранным слоям, содержащим следующие типы объектов Полигон, Линия, Символ или Текст. После нажатия кнопки откроется соответствующий диалог.
--	---

Масштабный эффект

Показывать в пределах	Установите флажок что бы активизировать масштабный фактор. Масштабный фактор позволяет устанавливать минимальное и максимальное расстояние в пределах которого этот слой будет видимым. Например, если надо увидеть отдельные улицы, когда линейный размер окна карты меньше 3 мили, установите минимальный размер на ноль а максимальный на 3.
Минимум	Укажите минимальное значение расстояния при котором будет виден слой.
Максимум	Укажите минимальное значение расстояния при котором будет виден слой.

Автоматическое подписывание

Для автоматического подписывания слоя на карте используется информация из этого слоя:

1. Из диалога [Управление слоями](#), выберите слой, который надо автоматически подписать и установите флажок [Автоматическое подписывание](#).
2. Нажмите [ОК](#).

Карта будет заново отрисована с подписями из указанной колонки, настройка из диалога [Подписывание](#).

Только одна колонка для таблицы может использоваться одновременно для подписывания.

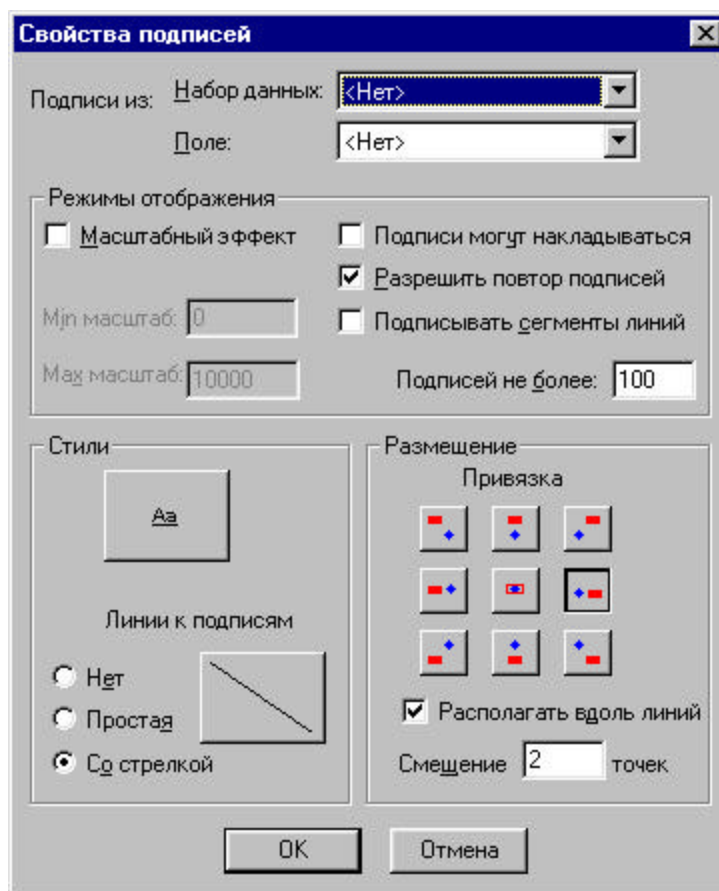
Используйте диалог **Подписывание** для изменения видимости, содержания, шрифтов, цвета текста, слияния линий и позиции размещения подписей.

Диалог Подписывание

Диалог **Подписывание** определяет содержания, шрифтов, цвета текста, слияния линий и позиции размещения подписей. Эти настройки используются и для автоматического и для интерактивного подписывания при использовании инструмента **Подпись**. Когда используется инструмент **Подпись**, то подпись разместится там, где Вы щелкните мышкой.

Что бы открыть диалог **Подписывание**:

1. Выполните команду **Карта > Управление слоями**.
2. Нажмите кнопку **Подпись**. Откроется диалог **Подписывание**.



Из колонки	Выберите набор данных и имя поля, из которого надо взять подписи.
Режимы показа	
Показать	Установите переключатель что бы сделать видимыми подписи.
Скрыть	Установите переключатель, что бы не показывать подписи.
Показ в пределах	Установите переключатель что бы активизировать масштабный эффект. Масштабный эффект позволит Вам установить минимальное и максимальное расстояние при котором будут показаны подписи. Например, если

	<p>надо увидеть подписи, когда линейный размер окна карты меньше 3 мили, установите минимальный размер на ноль а максимальный на 3.</p> <p>Размер подписи не меняется при изменении масштаба.</p>
Минимум	Введите минимальное расстояние окна карты, при котором подписи видны.
Максимум	Введите максимальное расстояние окна карты, при котором подписи видны.
Подписей не более	Введите число что бы одна и та же подпись изображалась более одного раза.
Подписи могут повторяться	Позволяет изображать более чем один объект с одинаковым текстом.
Подписи могут накладываться	Установите флажок что бы подписи к разным объектам могли накладываться.
Стиль	
Кнопка Стиль	Нажмите эту кнопку и откроется диалог Стиль текста.
Указка	Здесь выбирается тип линии или ее отсутствие для связки подписи и объекта.
Нет	Не показывать линию.
Линия	Создается линия связывающая подпись объекта и центрост объект. Линия появляется после того, как Вы отодвигаете подпись от объекта.
Стрелка	Создается линия в виде стрелки связывающая подпись объекта и центрост объект. Линия появляется после того, как Вы отодвигаете подпись от объекта.
Положение	
Привязка	Нажмите на одну из кнопок, отражающих положение подписи относительно объекта. Синий ромбик соответствует объекту, красный прямоугольник соответствует подписи. Нажатая кнопку будет

	выглядеть “утопленной”.
Располагать вдоль линий	Установите флажок что бы располагать текст параллельно линиям сегментов. Эта настройка не действует для объектов типа точек.
Смещение_точек	Введите число точек (единицы измерения - размер текста), на которое подпись смещается от места привязки.

Проекция

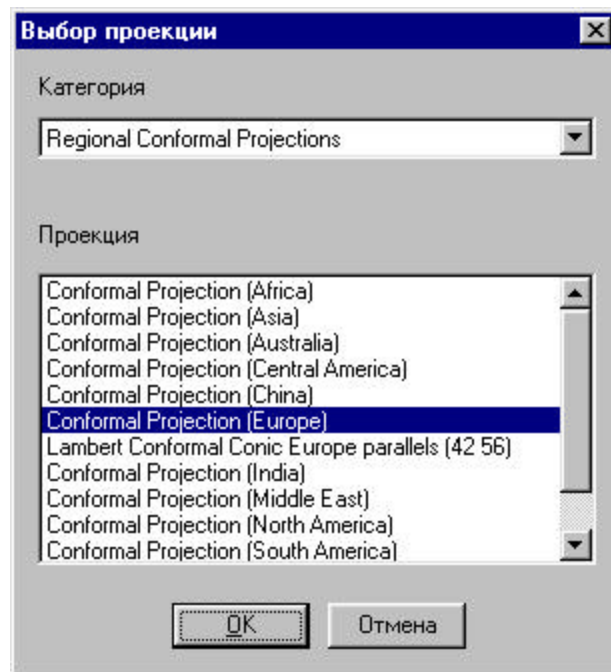
Как преобразуется искривленная поверхность Земли в плоскую на карте Вашего экрана? Вы используете проекцию. Проекция это система преобразований объектов с поверхности сфероида в плоские. Можно изобразить карты в разных проекциях.

Когда Вы трансформируете объекты, происходит изменение их формы.

Проекция это метод минимизации искажений, происходящих при переводе объектов из сферической поверхности на плоскую. Есть много различных типов проекций, каждая из которых по своему минимизирует эти искажения в зазных областях Земли.

Что бы выбрать проекцию:

Выполните команду [Карта > Проекция](#) или нажмите правую кнопку мышки и выберите проекцию из быстрого меню.



Использование меню View

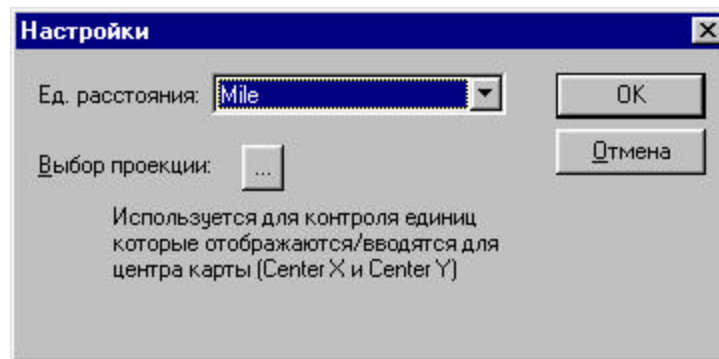
Используйте **Настройки (Options)** в меню **View** для изменения систем координат и единиц измерения расстояний на карте и отображения строки состояния или панели инструментов.

Отображение панелей

Чтобы показать/скрыть панель **Map**, **Geoset Name** или **Status Bar** установите или снимите соответствующий флажок в меню.

Диалог Options

Используйте диалог **Options (Настройки)** для определения единиц измерения и системы координат, используемых на карте.



Для настройки этих параметров выполните команду [View > Options](#).
Откроется диалог [Options](#).

Distance Units	Выберите единицы измерения, в которых измеряется расстояние на карте.
Numeric Coordinates	Если выбрано “numeric coordinates” то Geoset Manager будет искать координаты X и Y . Система координат это ряд параметров, сообщающие о том, как интерпретируются координаты объекта. Каждая точка в геометрическом объекте представлена двумя числами. Такие числа являются координатами для этих точек.

Программы и утилита GeoDictionary Manager

Используйте меню [Программы \(Tools\)](#) для доступа к утилите GeoDictionary manager. GeoDictionary manager позволит Вам регистрировать слои в геословаре. Слои, зарегистрированные в геословаре могут использоваться для связи данных в других сеансах MapX.

Для более подробной информации о Geodictionary Manager, обратитесь к приложению: *Использование утилиты GeoDictionary Manager*.

Приложение С: Использование утилиты Geodictionary Manager

Этот раздел описывает особенности утилиты MapX Geodictionary Manager. Geodictionary (Геословарь) это файл содержащий регистрационную информацию о таблицах MapInfo которые могут распознаваться MapX во время автоматической связи данных. Только те таблицы MapInfo которые будут опознаны, зарегистрируются в Геословаре.

Внимание: Нет необходимости регистрировать каждый .tab файл, который приложение использует в Геословаре.

Приложение Geodictionary Manager позволяет работать с геословарем. Это приложение независимое ([GeoDictionaryManager40.exe](#)) и может быть запущено из графического интерфейса или с командной строки. Предпочтительно вызывать приложение [GeoDictionaryManager40.exe](#) из инсталляционной программы что бы регистрировать таблицы, которые приложению MapX могут понадобиться из Геословаря.

История

Предыдущие версии MapX использовали MapInfo Data Installer разработанный для Microsoft Map что бы управлять Геословарем. Data Installer так же управляет геонаборами (группами таблиц MapInfo которые могут быть отображены совместно), что приводило к конфликтам с Geodictionary Manager. Теперь есть приложение Geoset Manager которое должно использоваться для создания и управления геонаборами. Новое приложение Geodictionary Manager не изменяет геонаборы, и не удаляет файлы с диска пользователя.

В MapX 4.0, файл GeoDictionary не требует запуска MapX. Смотрите ниже детали по файлу GeoDictionary.

Здесь были изменены настройки командной строки, которая требует, что бы файл или имя пути были в кавычках. Можно создать командную строку, которая ранее бы не работала.

Файл Geodictionary

Geodictionary хранится в бинарном формате, всегда называемом geodict.dct. Файл geodictionary может быть помещен или удален с помощью регистрационного ключа

HKKEY_LOCAL_MACHINE\SOFTWARE\MapInfo\MapX

4.0\GeoDictionary. Есть три варианта для этого ключа:

- Может содержать и имя файла и полный путь (e.g. "**C:\Program Files\MapInfo MapX\Maps\geodict.dct**"). Эффект в том, что директория с данными устанавливается так что бы быть одновременно и путем к файлу Geodictionary, названному по этому ключу. Геословарь, попадающий в память, инициализируется из файла Geodictionary и слои содержащиеся в Geodictionary могут распознаваться автоматически. Слои добавленные к карте одновременно добавляются к геословарю, который в памяти, так что они могут распознаваться автоматически.
- Может содержать только путь (например, "**C:\Program Files\MapInfo MapX\Maps**"). Директория с данными устанавливается по имени этого ключа. Геословарь, находящийся в памяти инициализируется пустым. Слои добавленные к карте, так же добавляются к геословарю, находящемуся в памяти, так что они могут распознаваться автоматически.
- Может совсем отсутствовать. Геословарь, находящийся в памяти инициализируется пустым. Слои добавленные к карте, так же добавляются к геословарю, находящемуся в памяти, так что они могут распознаваться автоматически.

Внимание: Команда Uninstall всего MapX оставит файл geodict.dct. Это произойдет по тому, что файл Geodictionary мог существовать и до установки (из более ранних версий MapX), или мог использоваться другими приложениями

Приложения MapX могут использовать один файл Geodictionary, или использовать собственные. Часто приложение MapX использует собственный файл Geodictionary (или не использует никакого), свойство Map.GeoDictionary может быть изменено в процессе отладки приложения и в нем будет указан собственный ключ регистрации. Затем, при запуске приложения, MapX запросит следующий ключ:

NKEY_LOCAL_MACHINE\SOFTWARE\MapInfo\MapX\4.0\<value of Map.GeoDictionary>

Значение этого ключа интерпретируется так, как описывалось выше. По умолчанию значение Map.GeoDictionary это "GeoDictionary" так что MapX будет использовать по умолчанию его.

Geodictionary Manager всегда читает и обновляет файл Geodictionary по умолчанию, ссылаясь на следующий ключ

NKEY_LOCAL_MACHINE\SOFTWARE\MapInfo\MapX\4.0\GeoDictionary. Если Вы меняете положение файла Geodictionary в пользовательском интерфейсе, то регистрационный ключ обновляется с указанием нового местоположения.

Geodictionary содержит элемент для каждой зарегистрированной таблицы (mapinfo .tab файл). Данные хранящиеся в каждом элементе это следующее:

- Имя ссылающееся на таблицу и отражающееся в диалогах управления слоями. По умолчанию это ярлык 'Description' в .tab файле или имя файла если никакого не существует. Оно может редактироваться через пользовательский интерфейс.
- Список индексированных полей. Пользовательский интерфейс позволяет Вам устанавливать, какое из индексированных полей надо MapX рассматривать как возможную колонку для идентификации во время автоматической связи данных. Поля, не помеченные Вами, будут игнорироваться. Когда таблица регистрируется первый раз, все индексированные поля изначально устанавливаются по умолчанию.

- Детализированное имя таблицы. Некоторые таблицы, типа, графства США, содержат индексированные колонки, не являющиеся уникальными. В этой ситуации, детализированное имя таблицы необходимо для точного поиска данных. Если таблица имеет неуникальные индексированные колонки, будет доступен список детализированных колонок и пользователь сможет указать детализированное имя таблицы.
- Список имен файлов Geoset. Когда таблица выбирается во время автоматической связи данных, список геонаборов передается в событие ResolveDataBind для программы MapX, что бы была выбрана одна. MapX укажет на первую попавшуюся по умолчанию, если событие не обработано.

Внимание: Таблица, которая подошла, всегда является загруженной, даже если ее нет в геонаборе, который загружен. Таким образом, это не ошибка, не иметь в списках геонаборов таблицы из геословаря (Geodictionary). В этом случае, загружается только таблица. В завершение, только файлы геонаборов, которые располагаются в той же директории, где и файл Geodictionary могут быть добавлены к списку.

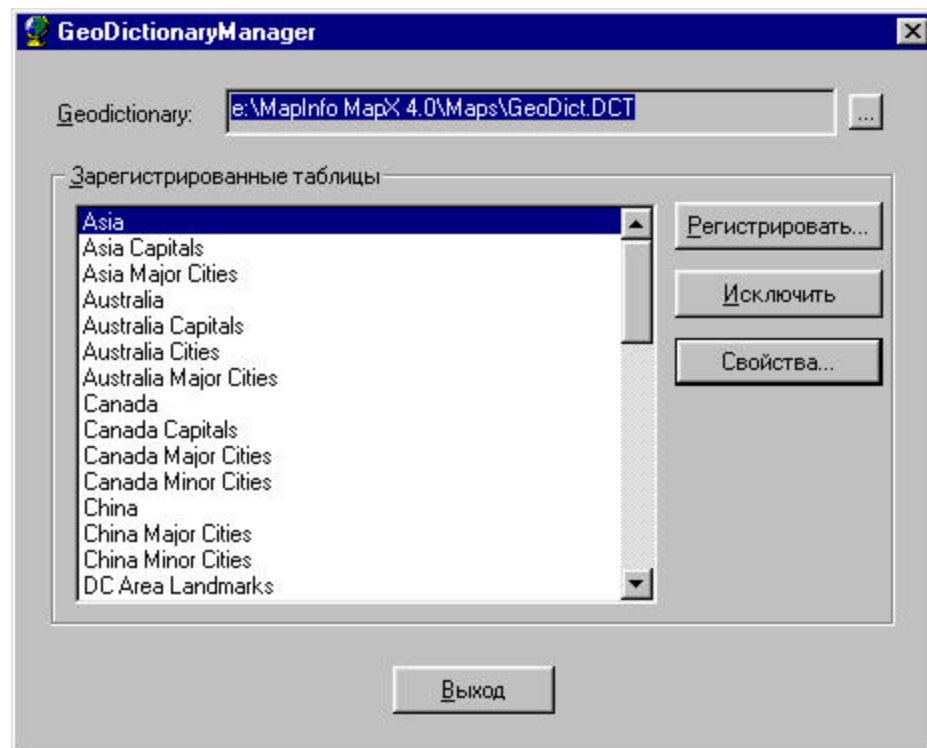
Пользовательский интерфейс

Запуск GeoDictionary Manager

Запускайте GeoDictionary Manager когда надо регистрировать слои вручную.

Выполните команду **Tools > Run GeoDictionary Manager**.

Откроется диалог **Geodictionary**.

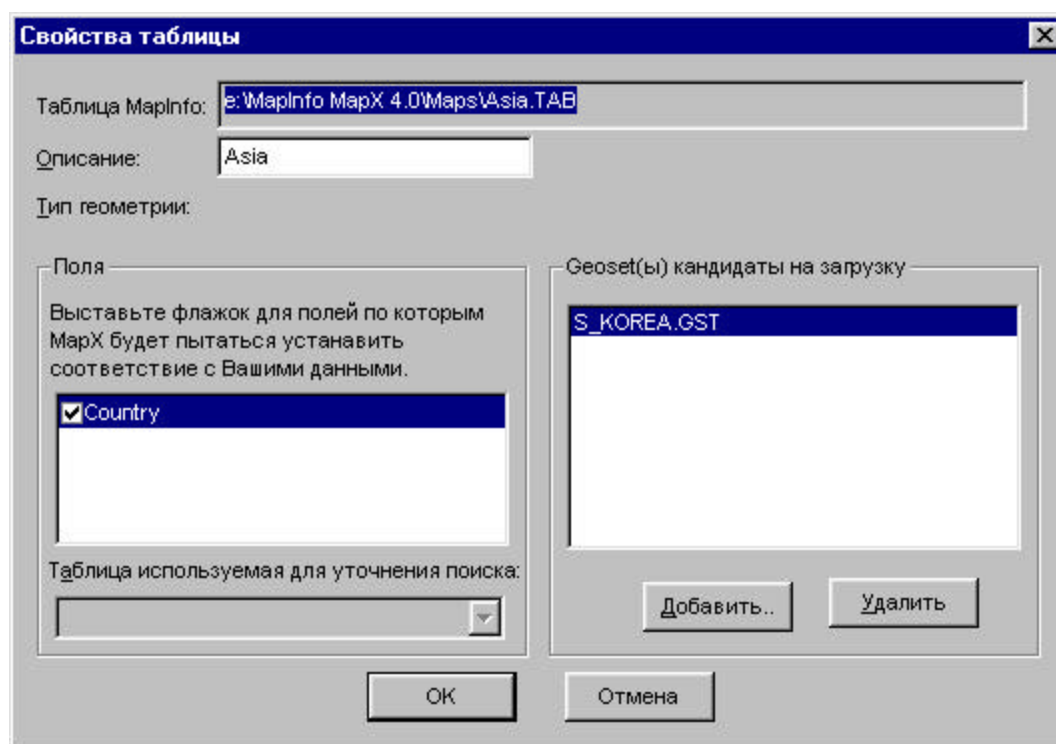


Geodictionary	Редакторское окошко Geodictionary доступно только для чтения и содержит полный путь к Geodictionary над которым предстоит управление. Эта информация находится при просмотре ключа MapX в registry. Кнопка (справа от редакторского окошка Geodictionary) позволяет пользователю
---------------	--

	<p>просматривать другие Geodictionary для управления ими. Просмотр других Geodictionary изменяет регистрационные ключи MapX так, что все последовательные сессии MapX будут использовать новый словарь. Кнопка так же позволяет Вам изменять по умолчанию путь поиска MapX где ищутся пути, хранящиеся в ключе "HKEY_LOCAL_MACHINE\SOFTWARE\MapInfo\MapX\SearchPaths".</p>
--	--

Registered tables	Список Registered Tables содержит список имен для всех таблиц, зарегистрированных в Geodictionary.
Register	Открывает общий диалог открытия файла с окошком типов файлов, установленным на тип "MapInfo Tables (*.tab)". Скопируйте таблицу в директорию с данными (директорию, содержащую Geodictionary), или добавьте директорию, содержащую таблицу для пути поиска данных. После того как таблица будет установлена (или копированием или добавлением директории для пути поиска), откроется диалог Table Properties. Пользователь может выбрать несколько файлов для регистрации, и также для снятия регистрации можно выбрать несколько таблиц, для этого нажмите клавиши Ctrl /Shift - и регистрация отменится. В случаях выбора нескольких таблиц, кнопка свойств (properties button) будет неактивна. Смотрите диалог Table Properties .
Unregister	Удаляет выделенную таблицу из Geodictionary. Кнопка Unregister не удаляет таблицу с диска.
Properties	Открывает диалог Table Properties для выбранной таблицы. Смотрите описание диалога Table Properties ниже.

Установите поля Geodictionary для данной таблицы используя диалог **Table Properties**.



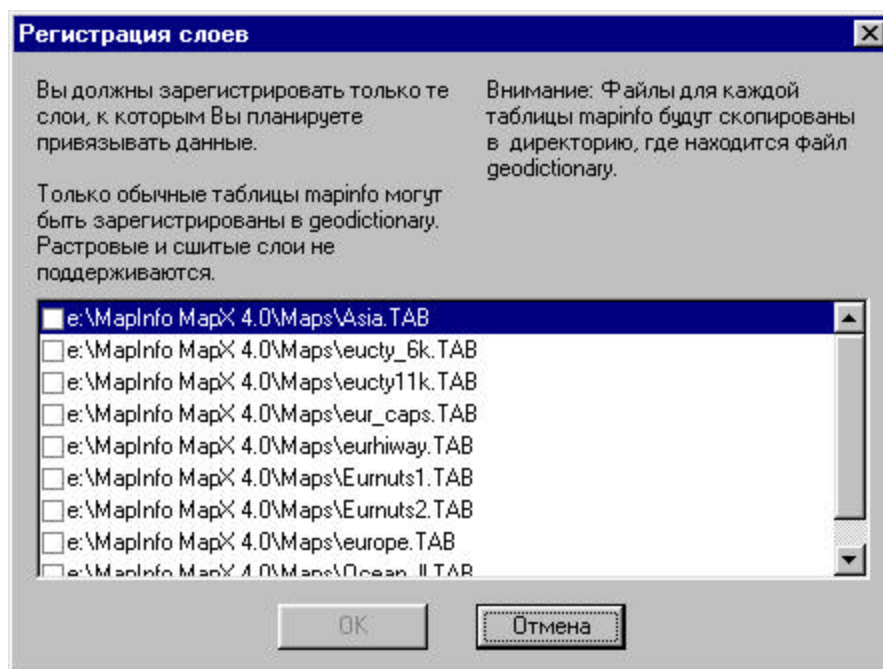
MapInfo Table	Окно только для чтения, содержащее имя файла таблицы MapInfo если она находится в той же директории, где и Geodictionary, или полный путь к файлу если он находится в другой директории.
Description	Обеспечивает механизм для изменения имени таблицы. Этот контроль обеспечивается по умолчанию разделом Description в файле .TAB, или именем файла если раздел Description не найден, но может быть изменен пользователем. Обратите внимание, что изменения в описании в Geodictionary Manager будут храниться в

	Geodictionary и не повлияют на таблицу. Это обстоятельство позволяет Geodictionary Manager легко работать с данными открытыми только для чтения, то есть с данными на CD-ROM.
Geometry	Geometry static текст контролирует изображение типов заголовков, хранящихся в таблице.
Field Information	Содержит список индексированных колонок в таблице. Если флажок напротив соответственной колонки установлен, то по этому полю будет искаться сходство при процессе сравнения.
Refining Table	Некоторые таблицы, типа графств США, содержат индексированные колонки, которые не уникальны. В этой ситуации, уточнение таблицы необходимо для определения наиболее точного совпадения данных. Если таблица имеет не уникальные индексные колонки, то будет активным окно с уточненными колонками и пользователь сможет выбрать уточненную таблицу.
Candidate Geosets	<p>Список имен файлов геонаборов, являющихся вероятными кандидатами на загрузку если таблица MapInfo в процессе связывания данных оказалась выбранной. В списке может вообще не быть имен геонаборов или быть несколько. Когда таблица выбрана во время автоматического связывания данных, список геонаборов для этой таблицы передается в событие ResolveDataBind чтобы программа MapX выбрала один из них. MapX выберет самый первый по умолчанию, если событие не обработано.</p> <p>Обратите внимание:</p> <ul style="list-style-type: none"> • Подошедшая таблица всегда оказывается загружена даже если ее нет в геонаборе, который загружается.

	<ul style="list-style-type: none"> • Не является ошибкой не иметь списка геонаборов для таблицы в геословаре Geodictionary. В этом случае загружается только таблица. • Только те файлы геонаборов, которые размещаются в той же директории, что и файл Geodictionary может быть добавлен к списку. • Несколько геонаборов - кандидатов могут быть одновременно добавлены или удалены
Add	Отображает окошко в котором список имен файлов геонаборов, находящихся в той же директории что и файл Geodictionary. Выбор одного из них добавляет его к списку геонаборов-кандидатов.
Remove	Удаляет выделенное имя файла геонабора из списка геонаборов-кандидатов.

Регистрация слоев в геословаре GeoDictionary

Если Вы хотите что бы Geoset Manager автоматически регистрировал Ваши слои, выполните команду **Tools > Register Layers**. Откроется диалог **Регистрация слоев**.



Установите флажок напротив того слоя который надо зарегистрировать и нажмите **ОК**.

Настройки командной строки

Командная строка GeoDictionary Manager обеспечивает механизм для добавления и удаления таблиц к/из геословаря Geodictionary.

geosetpath	<p>Путь к файлу геонабору содержащий список таблиц. Обычно требуется путь, указание только одного имени воспринимается как ошибка.</p> <ul style="list-style-type: none">• Используйте ключевое слово <code>geosetpath</code> для регистрации всех подходящих <code>.tab</code> файлов в геонаборе с геословарем Geodictionary. (Растры, Сшитые и Списки в геонаборе игнорируются.)• Используйте настройку <code>geosetpath /remove</code> для отмены регистрации всех <code>tab</code> файлов в геонаборе из.
tablepath	<p>Путь к таблице MapInfo, добавляемой к Geodictionary (ключевое слово без аргументов).</p> <ul style="list-style-type: none">• Используйте <code>tablepath</code> для регистрации файла в Geodictionary.• Используйте <code>tablepath /remove</code> для снятия регистрации файла из геословаря Geodictionary.
geosetfile	<p>Имя файла только того геонабора, который расположен в той же директории, что и файл Geodictionary.</p> <ul style="list-style-type: none">• Используйте <code>tablepath /geoset=geosetfile</code> для регистрации файла с геословарем Geodictionary и добавления <code>geosetfile</code> к его списку геонаборов-кандидатов для поиска подходящего.• Используйте <code>tablepath /geoset=geosetfile /remove</code> для удаления <code>geosetfile</code> из списка геонаборов-кандидатов для поиска подходящего.
commandfilepath	<p>Полный путь <code>pathname</code> к текстовому файлу, который содержит <code>migm30</code> командную строку в каждой строке файла.</p>

	<ul style="list-style-type: none"> • Используйте /file=commandfilepath для обработки текстового файла командных строк, с 1 командной строкой в каждой строчке. Не включайте 'migm30.exe' в каждой строке файла. • Используйте /file=commandfilepath /remove для обработки текстового файла командных строк, с 1 командной строкой в каждой строке файла. Ключевая команда / remove будет добавляться к каждой командной строке перед тем как выполняться. Не включайте 'migm30.exe' в каждой строке файла.
--	--

Внимание: Пути или имена файлов с пробелами должны быть заключены в кавычки. Там где это возможно, использование командной строки Geodictionary Manager будет осуществляться немедленно. Не является ошибкой попытка зарегистрировать файл более одного раза. Не является ошибкой попытка снять регистрацию с той таблицы, которая не была зарегистрирована.

Приложение D: Поддержка собственных наборов данных Dataset

Если у Вас нет возможности получить доступ к данным посредством стандартной техники связывания данных MapX, Вы можете создать собственную поддержку наборов данных dataset. MapX версии 3 и позднее обеспечивает выполнимую программную архитектуру, позволяющую Вам подключать пользовательские типы наборов данных dataset через базирующийся на COM интерфейс набора данных dataset.

Создание поддержки пользовательского набора данных это дополнительная, усложненная тема. Эта дискуссия предполагает, что Вы уже знакомы с созданием COM объектов.

Об объекте Static Dataset

Объект Static Dataset это COM объект, который MapX может использовать для поиска данных из произвольного источника данных. Данные выбираются и объединяются, когда источник данных является связанным (вызовом метода `Datasets.Add(...)`) и затем кэшируется самим MapX. Изменения в источнике данных не отображаются на карте до тех пор, пока не произойдет обновления данных, кэшированных самим MapX (вызовом метода `Dataset.Refresh(...)`). Все необходимые процедуры для запроса и кэширования данных таким путем группируются в три интерфейса: `IMMapXDataset`, `IMMapXStaticDataset` и `IMMapXColumnInfoContainer`. Объект Static Dataset должен осуществлять все три эти интерфейса (обратите внимание, что `IMMapXStaticDataset` это производное от `IMMapXDataset`). Смотрите ниже более подробное описание того как сформированы эти интерфейсы.

Специальная регистрация

При регистрации, сервер объекта Static Dataset должен осуществить некоторые специальные дополнения к registry так что бы MapX знал когда их использовать:

Добавьте ключ:

```
HKKEY_LOCAL_MACHINE\SOFTWARE\MapInfo\MapX\DatasetEngin  
es\<DatasetID >
```

Добавьте строковую величину:

```
HKKEY_LOCAL_MACHINE\SOFTWARE\MapInfo\MapX\DatasetEngin  
es\<DatasetID > \DE_CLSID =  
CLSID объекта Static Dataset (в формате "{xxxxxxxx-  
xxxx-xxxx-xxxx-xxxxxxxxxxxxx}").
```

Где <DatasetID> это уникальный числовой идентификатор для Вашего объекта Static Dataset. Когда параметр “Type” метода Dataset.Add(...) принимает это значение, Ваш сервер будет опрошен на предмет объекта Static Dataset посредством которого MapX в конце концов отыщет данные. Значения от 0 до 1000 зарезервированы MapInfo и не должны использоваться.

Внимание: Не забывайте удалять эти ключи когда Ваш сервер незарегистрирован!

Интерфейсы

Интерфейс IMapXDataset

```
[
    uuid(96e0f395-caec-11d0-9d99-00aa00a478cb) ,
    helpstring("IMapXDataset Interface"),
    version(1.0)
]
interface IMapXDataset : IUnknown
{
    HRESULT Init(
        [in] short sType,
        [in] VARIANT* pvSourceData,
        [in] VARIANT* pvFields );
    HRESULT GetSample(
        [in] long lColNum,
        [in] long lNumSampleValuesRequested,
        [in] VARTYPE vtRequested,
        [out] VARIANT * pvarData,
        [out] long * pNumRecordsFetched);
}
```

IMapXDataset::Init

HRESULT Init

[in]	short	sType	Требуется тип набора данных dataset (параметр Type для метода Datasets.Add). MapX уже проидентифицировал Ваш сервер как сервер для наборов данных этого типа (путем ввода ключей в registry, которые сделаны, когда регистрировался Ваш COM сервер).
[in]	VARIANT	pvSourceDa	Ссылка на источник данных. (Параметр

	*	ta	SourceData метода Dataset.Add). Например, этот параметр содержит объект ODBCQueryInfo когда связываемый (bind) тип это miDatasetODBC и указатель IDAORcordset когда связываемый тип это miDataSetDAO.
[in]	VARIANT*	pvFields	Содержит массив VARIANTS. Каждый VARIANT идентифицирует (или по имени или по индексу) колонку из источника данных, который клиент программист хочет связать с искомым слоем карты. Если здесь пусто (pvFields->vt = VT_EMPTY) то все колонки из источника данных будут связаны.

:)

Описание

Инициализирует объект Static Dataset. Этот метод будет вызван немедленно после создания объекта Static Dataset. Это дает объекту Static Dataset возможность устанавливать соединение с источником данных и идентифицировать какие колонки из источника данных будут связаны с картой. Объект Static Dataset должен, перед возвращением из этого вызова, построить внутреннюю коллекцию объектов ColumnInfo (объектов, которые выполнены [IMMapXColumnInfo](#)). Коллекция должна содержать инициализированный объект ColumnInfo для каждой связанной колонки. Эта коллекция выставляется для MapX посредством интерфейса [IMMapXColumnInfoContainer](#).

Возвращаемое значение

S_OK возвращается в случае успеха.
E_OUTOFMEMORY возвращается, если превышены размеры пространства допустимой памяти.

E_FAIL возвращается в случае других сбоев.

IMMapXDataset::GetSample

HRESULT GetSample();

[in]	Long	lColNum	Колонка в источнике данных откуда берется пример.
[in]	Long	lNumSamplesRequested	Число требуемых для примера данных.
[in]	VARTYPE	vtRequested	Предпочтительный тип для возвращаемых данных.
[out]	VARIANT *	pvData	VARIANT для получения примера данных. Когда возвращается GetSample, этот VARIANT должен содержать массив требуемого типа. MapX будет пытаться конвертировать данные, если они принадлежат другому типу, но это будет очевидно менее эффективно.
[out]	Long *	plNumSamplesValuesFetched	Число данных примера в pvData.

Описание

Используется для сбора примера данных из заданной колонки источника данных. Обратите внимание, что возвращаемая переменная (pvData) должна содержать массив, даже если этот массив содержит только одно значение данных.

Возвращаемое значение

- S_OK возвращается в случае успеха.
- E_OUTOFMEMORY возвращается, если превышены размеры пространства допустимой памяти.
- E_FAIL возвращается в случае других сбоев.

Интерфейс IMapXStaticDataset

```
[
    uuid(2e6d4cc0-d132-11d0-9da0-00aa00a478cb),
    helpstring("IMapXStaticDataset Interface"),
    version(1.0)
]
interface IMapXStaticDataset : IMapXDataset
{
    HRESULT FetchData(
        [in] long lColNum,
        [in] long lIdxRow,
        [in] VARTYPE vtRequested,
        [out] VARIANT* pvarData,
        [out] BOOL * pbNoMoreData);
    HRESULT BeginFetch();
    HRESULT EndFetch();
}
```

IMapXStaticDataset::FetchData

HRESULT FetchData();

[in]	Long	lColNum	Колонка источника данных из которой выбираются данные.
[in]	Long	lIdxRow	Строка источника данных из которой выбираются данные.

[in]	vtRequested	vtRequested	Предпочитаемый тип возвращаемых данных.
[out]	VARIANT *	pvarData	VARIANT переменная для удержания выбранных данных. Должна быть такого типа, который определен vtRequested. MapX будет пытаться конвертировать данные, если они принадлежат другому типу, но это будет очевидно менее эффективно
[out]	BOOL *	pbNoMoreData	Флаг, показывающий, что выборка дошла до конца доступного источника данных. Если имеется 10 строк источника данных, установите этот флаг, когда MapX попытается выбрать 11-ю.

Описание

Выбирает ячейку данных из источника данных.

Возвращаемое значение

S_OK возвращается в случае успеха.

E_OUTOFMEMORY возвращается, если превышены размеры пространства допустимой памяти

E_FAIL возвращается в случае других сбоев.

IMapXStaticDataset::BeginFetch

HRESULT BeginFetch();

Описание

Указывает начало последовательности выборки. Может использоваться для размещения/установки ресурсов необходимых для выборки данных.

Возвращаемое значение

S_OK возвращается в случае успеха.

E_OUTOFMEMORY возвращается, если превышены размеры пространства допустимой памяти

E_FAIL возвращается в случае других сбоев.

IMMapXStaticDataset::EndFetch

HRESULT EndFetch();

Описание

Указывает, что последовательность выборки завершена. Может использоваться что бы освободить ресурсы, которые используются для выборки данных.

Возвращаемое значение

S_OK возвращается в случае успеха.

E_OUTOFMEMORY возвращается, если превышены размеры пространства допустимой памяти

E_FAIL возвращается в случае других сбоев.

Интерфейс IMMapXColumnInfoContainer

```
[
    uuid(1e584f00-d2a5-11d0-9da3-00aa00a478cb),
    helpstring("IMMapXColumnInfoContainer
Interface"),
    version(1.0
]
interface IMMapXColumnInfoContainer : IUnknown
{
    HRESULT GetColumnInfoByName(
        [in] BSTR bstrColumnName,
```

```

        [out] IMapXColumnInfo** ppIMapXColumnInfo);
HRESULT GetColumnInfoByIndex(
    [in] long lIndex,
    [out] IMapXColumnInfo** ppIMapXColumnInfo);
HRESULT GetColumnInfoEnumerator(
    [out] IEnumMapXColumnInfo**
ppIEnumMapXColumnInfo);
}

```

IMapXColumnInfoContainer::GetColumnInfoByName

HRESULT GetColumnInfoByName();

[in]	BSTR	bstrColumnName	Имя колонки источника данных.
[out]	IMapXColumnInfo **	ppIMapXColumnInfo	Объект ColumnInfo идентифицированный bstrColumnName

Описание

Используется для для поиска объекта ColumnInfo по имени колонки.

Возвращаемое значение

S_OK возвращается в случае успеха.

E_OUTOFMEMORY возвращается, если превышены размеры пространства допустимой памяти

E_FAIL возвращается в случае других сбоев.

IMapXColumnInfoContainer::GetColumnInfoByIndex

HRESULT GetColumnInfoByIndex ();

[in]	Long	lIndex	Индекс колонки источника данных.
------	------	--------	----------------------------------

[out]	IMMapXColumnInfo **	ppIMMapXColumnInfo	Объект ColumnInfo определенный индексом Index.
-------	---------------------	--------------------	--

Описание

Используется для запроса объекта ColumnInfo по номеру колонки.

Возвращаемое значение

S_OK возвращается в случае успеха.

E_OUTOFMEMORY возвращается, если превышены размеры пространства допустимой памяти

E_FAIL возвращается в случае других сбоев.

IMMapXColumnInfoContainer::GetColumnInfoEnumerator

HRESULT GetColumnInfoEnumerator();

[out]	IEnumMapXColumnInfo **	ppIEnumMapXColumnInfo	Нумератор
-------	------------------------	-----------------------	-----------

Описание

Используется для получения объекта ColumnInfo Enumerator (объект который выполняет IMMapXEnumColumnInfo). Этот объект может использоваться для нумерации коллекции объектов ColumnInfo содержащихся в наборе данных dataset.

Возвращаемое значение

S_OK возвращается в случае успеха.

E_OUTOFMEMORY возвращается, если превышены размеры пространства допустимой памяти

E_FAIL возвращается в случае других сбоев.

Об объекте ColumnInfo Enumerator

Объект ColumnInfo Enumerator используется для нумерации коллекции объектов ColumnInfo, содержащихся в объекте Dataset. Он использует интерфейс IEnumMapXColumnInfo (описан ниже)

Интерфейс IEnumMapXColumnInfo

```
[
    uuid(96e0f396-caec-11d0-9d99-00aa00a478cb),
    helpstring("IEnumColumnInfo Interface"),
    version(1.0)
]
interface IEnumMapXColumnInfo : IUnknown
{
    HRESULT Next([in] ULONG celt, [out]
IMMapXColumnInfo ** rgelt, [out] ULONG *
pceltFetched);
    HRESULT Skip([in] ULONG celt);
    HRESULT Reset();
    HRESULT Clone([out] IEnumMapXColumnInfo **
ppIMMapXColumnInfo);
}
```

IEnumMapXColumnInfo это стандартный интерфейс для нумерации, созданный после IEnumUnknown, который используется для нумерации указателей IMMapXColumnInfo. Более полная документация об IEnumUnknown, находится в OLE32 SDK.

Об объекте ColumnInfo

Объект ColumnInfo содержит информацию о колонке источника данных. Он приводит в действие интерфейс IMapXColumnInfo (описанный ниже)

Интерфейс IMapXColumnInfo

```
[
    uuid(96e0f394-caec-11d0-9d99-00aa00a478cb),
    helpstring("IMapXColumnInfo Interface"),
    version(1.0)
]
interface IMapXColumnInfo : IUnknown
{
    HRESULT Init(
        [in] BSTR bstrName,
        [in] VARTYPE vt,
        [in] long lColNum);
    HRESULT GetName([out] BSTR *pbstrName);
    HRESULT GetDataType([out] VARTYPE *pvt);
    HRESULT GetColumnNumber([out] long *plColNum);
}
```

IMapXColumnInfo::Init

HRESULT Init(

[in]	BSTR	bstrName	Имя колонки источника данных.
[in]	VARTYPE	vt	Тип колонки источника данных.
[out]	long	lColNum	Индекс колонки источника данных.

Описание

Инициализирует объект ColumnInfo. MapX никогда не вызовет этот метод, это задача объекта Dataset инициализировать объект ColumnInfo после его создания. Обратите внимание, что MapX в данное время поддерживает только числовые и стринговые типы данных.

Возвращаемое значение

S_OK возвращается в случае успеха.

E_OUTOFMEMORY возвращается, если превышены размеры пространства допустимой памяти

E_FAIL возвращается в случае других сбоев.

IMapXColumnInfo::GetName

```
HRESULT GetName(  
[out]          BSTR *          pbstrName  
Имя колонки   );  
источника  
данных.
```

Описание

Отыскивает имя колонки источника данных.

Возвращаемое значение

S_OK возвращается в случае успеха.

E_OUTOFMEMORY возвращается, если превышены размеры пространства допустимой памяти

E_FAIL возвращается в случае других сбоев.

IMapXColumnInfo::GetDataType

```
HRESULT GetDataType(  

```

[out]VARTYPE *pvt тип колонки источника данных.);

Описание

Отыскивает тип данных колонки источника данных. Обратите внимание, что MapX в настоящее время поддерживает только числовые и строковые типы данных.

Возвращаемое значение

S_OK возвращается в случае успеха.

E_OUTOFMEMORY возвращается, если превышены размеры пространства допустимой памяти

E_FAIL возвращается в случае других сбоев.

IMapXColumnInfo::GetColumnNumber

HRESULT GetColumnNumber(

[out]long *pColNumThe номер колонки источника данных);

Описание

Отыскивает номер колонки источника данных.

Возвращаемое значение

S_OK возвращается в случае успеха.

E_OUTOFMEMORY возвращается, если превышены размеры пространства допустимой памяти

E_FAIL возвращается в случае других сбоев.