

Начало работы с MapServer

Статья написана по результатам первого знакомства с MapServer. В ней коротко описаны основные моменты, с которыми должен столкнуться человек, заинтересовавшийся этим пакетом.

[Обсудить в форуме](#) Комментариев — 30

Введение

Описание основных принципов работы MapServer, создания карт и шаблонов приводится в учебнике <http://biometry.gis.umn.edu/tutorial/>, доступного только на английском языке. Приведенные ниже примеры базируются на этом учебнике, поэтому, при возникновении каких-либо вопросов, первым делом следует найти соответствующий раздел учебника.

Все упоминаемые в тексте статьи файлы (html-шаблоны, map-файлы, shp-файлы и т.д.) [доступны для скачивания](#). Для того, чтобы работали примеры, разобранные в этой статье, нужно распаковать папку example в каталог /ms4w/apps/example.

Основу MapServer составляет CGI-программа **mapserv.exe** (в случае MS4W хранящаяся в каталоге \ms4w\Apache\cgi-bin\), которая принимает от пользователя параметры (файл описания карты, слои карты, размеры текущей карты и т.п.), указанные в адресной строке браузера, после чего отображает требуемую карту пользователю. Обычно это происходит так: получив от пользователя запрос, MapServer генерирует растровый файл и встраивает его в html-документ, отсылаемый пользователю. Какие слои будут участвовать при генерации файла, как именно они будут отображаться, будут ли подписаны объекты на карте, а также многое другое указывается в специальном файле с расширением .map, который передается программе mapserv.exe в качестве одного из параметров, заданных пользователем в строке адреса.

Большая часть работы над созданием web-ресурса как раз и состоит из написания map-файла. Подразумевается, что MapServer уже установлен и работает ([подробнее про установку MapServer для Windows](#)).

Дальнейшая инструкция по шагам демонстрирует как, используя готовые векторные данные, от простого к сложному создать свою интерактивную карту и разместить ее с помощью MapServer в интернет.

Оглавление

1. [Карта из одного слоя](#)
2. [Использование нескольких классов](#)
3. [Создание собственных условных обозначений](#)
4. [Создание подписей к объектам](#)
5. [Использование растровых слоев](#)
6. [Проекции](#)
7. [Создание интерактивной карты](#)
8. [Инструменты масштабирования карты](#)
9. [Масштабная линейка](#)
10. [Управление слоями карты](#)
11. [Добавление обзорной карты](#)
12. [Создание легенды](#)
13. [Запросы](#)

1. Карта из одного слоя

Для создания карты нам понадобится по крайней мере один слой. MapServer поддерживает множество форматов для растровых и векторных слоев ([список векторных форматов](#)). Интересно, что среди поддерживаемых форматов есть, к примеру, файлы Mapinfo, а также, что MapServer позволяет использовать информацию, хранящуюся в базах данных (например, MS SQL Server, PostgreSQL, Oracle и др.). Однако, доступ к данным в этих форматах обеспечивается специальным пакетом (OGR), "родной формат" для MapServer - shp. Для MS4W пакет OGR (заранее скомпилированный и настроенный) уже включен в поставку, тем не менее, в

этой статье рассматривается работа только с shp-файлами. Таким образом, для дальнейшей работы нам потребуется несколько shp-файлов (любых), которые скопируем в каталог "\\ms4w\\apps\\example\\shp\\". Можно воспользоваться [уже упоминаемыми shp-файлами](#).

Теперь начнем разбираться с форматом map-файла. Создадим в каталоге "\\ms4w\\apps\\example" файл polt.map. Заносим туда что-то вроде следующего:

```
# Аннотированный map-файл (за основу взят файл из учебника
# http://biometry.gis.umn.edu/tutorial/)
#
# Все, что идет за символом решетки - комментарий и программой не обрабатывается
#
# Map-файлы начинаются с ключевого слова MAP, обозначающего начало
# "map"-объекта. Закрывает map-объект ключевое слово END в конце файла. Вся
# карта, которая будет отображаться пользователю описывается внутри.

MAP
  IMAGETYPE      GIF
  EXTENT         34.59 49.58 34.63 49.6
  SIZE           400 300
  SHAPERPATH     "/ms4w/apps/example/shp/"
  IMAGECOLOR     255 255 255

# Внутри MAP-объекта определяются новые объекты - слои (LAYER).
# Обязательно нужно определить по крайней мере один слой.
# Количество слоев ограничено сверху (по умолчанию - не больше 100 слоев),
# если нужно большее количество слоев, придется перекомпилировать
# MapServer (см. заголовочный файл map.h)

LAYER # Определяем полигональный слой
  NAME          veget
  DATA         Poltava10_Vegetation_region
  STATUS        ON
  TYPE          POLYGON

# Внутри слоя нужно определить как минимум один класс. Классов может быть
# несколько, но не больше 10 (иначе опять придется перекомпилировать MapServer)

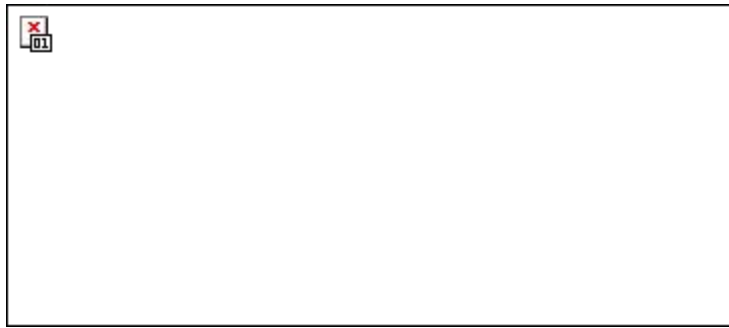
CLASS
  NAME          "Растительность"

# Внутри класса определяются стили: как именно данный класс отобразить
# на карте.
  STYLE
    COLOR        232 232 232
    OUTLINECOLOR 32 32 32
  END
END
END # Конец определения слоя
END # Конец определения карты
```

Чтобы посмотреть результат, наберите в окне браузера:

```
http://localhost/cgi-
bin/mapserv.exe?map=/ms4w/apps/example/polt.map&layer=veget&mode=map
```

Если вы работаете не на локальной машине, то вместо localhost нужно указать имя вашего сервера. Должно появиться следующее изображение:



Разберемся, что же написано в тар-файле ([описание этих и других параметрах тар-объекта](#)):

IMAGETYPE GIF - тип раstra, который сгенерирует MapServer для отсылки пользователю.

EXTENT x1 y1 x2 y2- координаты нижнего левого и верхнего правого углов карты (географический охват), отображаемый на экране. В нашем случае слои карты представлены в географических координатах (десятичные градусы, широта/долгота, WGS84) и все они укладываются в прямоугольник с координатами углов: (34.59 49.58, 34.63 49.6).

SIZE a b - размеры карты в пикселах.

SHAPEPATH "адрес" - местоположение shp-файлов.

IMAGECOLOR r g b - цвет фона карты в формате rgb.

Параметры объекта "LAYER":

NAME "имя" - название слоя.

DATA "имя файла " - название shp-файла, соответствующего данному слою.

STATUS ON|OFF|DEFAULT - переключатель, означающий должен ли отображаться слой на карте. Похоже что MapServer самостоятельно переключает этот параметр. При попытке вручную изменить его значения на ON и на OFF ничего не происходило. Будет показываться слой или не будет, зависело от параметров в адресной строке или шаблоне (об этом сказано ниже)).

TYPE point|line|polygon|circle|annotation|raster|query|chart - Тип слоя. Тут все более-менее понятно, хотя и есть свои особенности. Значение выражений point|line|polygon очевидно - слой будет точечный, линейный или полигональный. Однако, если у вас есть shp-файл с полигонами, вы можете присвоить ему тип line. При этом слой будет отображаться как линейная тема. Мало того, можно один и тот же shp-файл разбить на два слоя. результат будет тот же, что и выше, только достигнут он другим способом.

```
LAYER # Определяем полигональный слой
  NAME          veget_p
  DATA          Poltava10_Vegetation_region
  STATUS         ON
  TYPE           POLYGON
  CLASS
    NAME         "Растительность"
    STYLE
      COLOR      232 232 232
    END
  END
END
LAYER # Определяем линейный слой
  NAME          veget_l
  DATA          Poltava10_Vegetation_region
  STATUS         ON
  TYPE           LINE
  CLASS
    NAME         "Растительность"
    STYLE
      COLOR      0 0 0
    END
  END
END
```

Разбирая этот пример, осталось выяснить, как осуществляется передача параметров в MapServer для последующей обработки. Вспомним, что мы набрали в адресной строке браузера адрес:

```
http://localhost/cgi-bin/mapserv.exe?map=/ms4w/apps/example/polt.map&layer=veget&mode=map
```

Этот адрес состоит из нескольких частей: первая часть (**http://localhost/cgi-bin/mapserv.exe**) - вызов CGI программы mapserv.exe. Остальное - это параметры, передаваемые программе mapserv.exe. Вызов программы отделяется от параметров знаком "?". Параметры передаются парами "параметр=значение", и отделяются друг от друга знаком "&" например, параметр "map=/ms4w/apps/example/polt.map" указывает MapServer где находится map-файл, на основе которого будет строиться карта. Параметр "layer=veget" указывает MapServer, что в карту должен быть включен слой veget. И, наконец, параметр "mode=map" означает, что MapServer должен послать полученную карту пользователю сразу, без создания временного изображения на диске. Использование некоторых других параметров описано в следующих разделах. [\(более подробная информация об этих и других параметрах\)](#).

Map-файл, использованный в этом разделе: polt.map.

2. Использование нескольких классов

С файлом Poltava10_Vegetation_region.shp связана атрибутивная таблица в виде dbf-файла. Среди полей dbf-файла есть поле "CodeТopo" в котором хранится код объекта. Например, код 71100000 соответствует древесной растительности, код 71314000 - травяной, код 72310000 - болотистой местности. Поскольку объекты с разными кодами должны выглядеть на карте различным образом, то нам нужен инструмент, позволяющий определить разные условные обозначения для одного и того же слоя в зависимости от значений атрибутов объектов (то есть классификацию). С этой целью в map-файле разрешено вводить несколько классов, соответствующих одному слою. Создадим файл polt1.map следующего содержания:

```
MAP
  IMAGETYPE      GIF
  EXTENT          34.59 49.58 34.63 49.6
  SIZE            400 300
  SHAPEPATH       "/ms4w/apps/example/shp/"
  IMAGECOLOR      255 255 255

  LAYER
    NAME          veget
    DATA          Poltava10_Vegetation_region
    STATUS         ON
    TYPE           POLYGON
    CLASSITEM      "CodeТopo" # Название поля, которое будет использоваться
                              # для определения класса

    CLASS
      NAME         "Деревья"
      EXPRESSION    '71100000' # Выражение, определяющее класс
      STYLE
        COLOR      12 200 12
      END
    END
    CLASS
      NAME         "Трава"
      EXPRESSION    '71314000'
      STYLE
        COLOR      12 255 12
      END
    END
    CLASS
      NAME         "Болота"
      EXPRESSION    '72310000'
      STYLE
        COLOR      120 120 255
```

```
END
END
END
END
```

Введите адрес и убедитесь в том, что результат соответствует следующему изображению:

```
http://localhost/cgi-
bin/mapserv.exe?map=/ms4w/apps/example/poltt1.map&layer=veget&mode=map
```



Объект LAYER содержит параметр CLASSITEM, в котором указывается поле, используемое для классификации объектов (в нашем случае CodeTopo). В объекте CLASS появляется дополнительный параметр EXPRESSION. В самом простом случае после слова EXPRESSION указывается строка, и, если атрибут shp-объекта совпадает с указанной строкой, то shp-объект относят к данному классу. Вместо строки можно использовать различные логические выражения, например: EXPRESSION (CODETOPO eq '72310000' OR 'TYPE_RU' eq 'заболоченный луг') ([подробнее о классификации](#)).

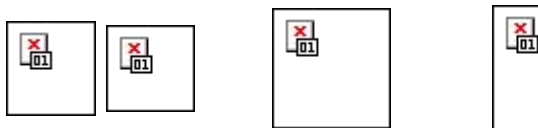
Мар-файл, использованный в этом разделе: poltt1.map.

3. Создание собственных условных обозначений

Вероятно, не каждому понравится, что болото обозначается на карте заливкой, поэтому займемся оформлением: [согласно документации](#) для каждого класса можно использовать собственный стиль условных обозначений, для этого в предложении STYLE нужно указать параметр SYMBOL. Кроме встроенного набора символов, задаваемых по умолчанию, MapServer поддерживает также подключение отдельных файлов, в которых собраны описания условных знаков ([пример файла с набором дополнительных символов](#)).

Теперь несколько примеров того, как создавать свои знаки. Мы создадим четыре новых стиля - два будут использоваться для точечных объектов, один для штриховки полигональной темы и еще один - для создания пунктирной линии. Примеры знаков представлены в следующей таблице.

triangle cross downwarddiagonalfill dashed1



Далее приводятся описания соответствующие каждому из этих знаков:

В команде POINTS указываются координаты узлов (x y). Несколько неожиданным может оказаться то, что ось Y направлена вниз, а не вверх как обычно. Координаты точек приводятся в условной системе координат. Символ "Треугольник":

```
SYMBOL
NAME "triangle"
TYPE vector
POINTS # ниже идут координаты точек - узлы объекта
0 4 # первая точка - начало движения: из точки с координатами (x=0,y=4)
2 0 # проводится линия в точку (x=2,y=0), далее линия пойдет
```

```

    4 4      # в точку (x=4,y=4)
    0 4      # последняя точка совпадает с первой
END
END

```

Символ "Крест". Особенность состоит в том, что крест, в отличие от треугольника невозможно нарисовать одним росчерком пера, не отрывая его от бумаги. Отрицательные координаты обозначают место, в котором "перо поднимается".

```

SYMBOL
  NAME "cross"
  TYPE vector
  POINTS
    0 0
    1 1
    -99 -99
    0 1
    1 0
  END
END

```

"Диагональная заливка" для полигональных объектов. Можно использовать параметр TRANSPARENT: 0 - не прозрачный объект, 100 - полностью прозрачный.

```

SYMBOL
  NAME "downwarddiagonalfill"
  TYPE vector
  TRANSPARENT 0      # прозрачность
  POINTS
    0 10
    10 0
  END
END

```

"Пунктирная линия". Предложение "STYLE 10 5 5 10 END" означает, что линия тянется 10 пикселей, потом 5 пикселей разрыв, потом линия появляется снова (5 пикселей) и опять разрыв в 10 пикселей.

```

SYMBOL
  NAME 'dashed1'
  TYPE ELLIPSE
  POINTS 1 1 END
  FILLED true
  STYLE 10 5 5 10 END
END

```

Как сказано выше, описания вновь созданных условных знаков удобно хранить в отдельном файле и ссылаться на него по мере необходимости. Чтобы подключить файл, в котором содержатся описания условных знаков, нужно к объекту MAP добавить параметр SYMBOLSET "путь_к_файлу/имя_файла". А в предложении CLASS нужно сделать ссылку на нужный символ предложением SYMBOL "имя_символа".

```

MAP
  ...
  SYMBOLSET      "./symbols/symbols35.sym"

  LAYER
    ...
    CLASS
      NAME        "Болота"
      EXPRESSION '72310000'
      STYLE
        SYMBOL 'Line1'      # ссылка на название стиля из стилевого файла "symbols35.sym"
        COLOR 120 120 255
      END
    END
  END
END

```

END

Результат использования только созданного символа для отображения класса "Болота" показан на следующем изображении:



С помощью MapServer возможно создавать и [более сложные знаки](#).

Мар-файл, использованный в этом разделе: polt2.map.

4. Создание подписей к объектам

Добавим к карте слой улиц. Он отличается от ранее рассмотренного предложения LAYER тем, что в нем использовано вложенное предложение LABEL, которое отвечает за подписи. ([подробнее о LABEL](#)).

```
LAYER
  NAME          street
  DATA          Poltava10_Street_polyline
  STATUS         ON
  TYPE          LINE
  LABELITEM      "RusName"    # подписи хранятся в поле "RusName"
  CLASS
    NAME         "Улицы"
    STYLE
      COLOR      12 12 12
    END
    LABEL
      COLOR 132 31 31
      SHADOWCOLOR 218 218 218
      SHADOWSIZE 2 2
      TYPE TRUETYPE
      FONT arial-italic
      SIZE 7
      ANTIALIAS TRUE
      POSITION CL
      PARTIALS FALSE
      MINDISTANCE 100
      BUFFER 3
      ENCODING CP1251
    END
  END
END
END
```

Для того, чтобы указать, как подписывать объекты, используется параметр LABELITEM "имя_поля", в котором указывается название поля, хранящего подписи. В предложение LABEL указывается множество параметров, но назначение наиболее важных параметров (COLOR, SHADOWCOLOR, SIZE, TYPE) очевидно.

Для русских пользователей возникает вечная проблема с кодировкой, решить ее можно добавив параметр ENCODING. Для shp-файлов значение кодировки равно CP1251 (кодировка Windows-1251). На тему кодировок в MapServer существует [подробное описание](#), поэтому всем, у кого появились проблемы в этой области, рекомендуем для начала ознакомиться с ним.

К map-объекту, до начала описаний слоев, нужно добавить параметр FONTSET "./fonts/fonts.list". Файл fonts.list (можно назвать файл и по-другому) хранит информацию о шрифтах, доступных MapServer. Пример

содержимого файла fonts.list:

```
# Это список шрифтов
```

```
arial                                arial.ttf
```

```
arial-italic                        ariali.ttf
```

Таким образом, параметр "FONT arial-italic", из предложения LABEL ссылается на шрифт ariali.ttf. Для определенности приведем содержимое каталога fonts:

```
fonts.list
```

```
arial.ttf
```

```
ariali.ttf
```

Файлы шрифтов arial.ttf и ariali.ttf можно взять из стандартного каталога Windows.

Результат запроса по адресу:

```
http://localhost/cgi-bin/mapserv.exe?map=/ms4w/apps/example/pol2.map&layer=veget&layer=street&mode=map
```



Мар-файл, использованный в этом разделе: polt2.map.

5. Использование растровых слоев

Кроме векторных наборов данных, MapServer также поддерживает использование растровых слоев. Подробное описание работы с растрами доступно [здесь](#), а в этом разделе будет дана краткая сводка основных моментов.

Обычно MapServer использует библиотеку GDAL для работы с растровыми изображениями (конечно, при условии, что MapServer был скомпилирован с поддержкой GDAL). К примеру, такая операция, как перепроектирование раstra "на лету", возможна только при использовании этой библиотеки.

Среди поддерживаемых растровых форматов есть все основные типы графических файлов: TIFF/GeoTIFF (для привязки используется или [world-файл](#), или внедренные в GeoTIFF координаты), GIF, PNG, JPEG, Erdas .LAN/.GIS (для привязки этих форматов используется world-файл).

В самом простом случае подключение растрового слоя состоит в добавлении предложения LAYER к мар-файлу:

```
LAYER
  NAME "JacksonvilleNC_CIB" # название слоя
  DATA "Jacksonville.tif"  # имя файла с расширением (пути прописываются относительно
  SHAPESPATH)
  TYPE RASTER
  STATUS ON
```


END

Также, как и у обычного слоя, у растрового слоя могут быть определены параметры PROJECTION, METADATA, PROCESSING, MINSCALE, и MAXSCALE. Но растровый слой не может иметь меток (LABEL) и параметров CONNECTION, CONNECTIONTYPE, или FEATURE.

Подобно векторному слою, при работе с растрами можно разбить один слой (один растр) на несколько классов, отображающихся по разному в зависимости от значений цветов:

```
LAYER
  NAME "JacksonvilleNC_CIB"
  DATA "Jacksonville.tif"
  TYPE RASTER
  STATUS ON
  CLASSITEM "pixel"
  # класс использует простое сравнение строк, эквивалентно (pixel = 0)
  CLASS
    EXPRESSION "0"
    STYLE
      COLOR 0 0 0
    END
  END
  # использование значений pixel.
  CLASS
    EXPRESSION (pixel >= 64 AND pixel < 128)
    STYLE
      COLOR 255 0 0
    END
  END
  # использование значений RGB (red/green/blue)
  CLASS
    NAME "near white"
    EXPRESSION (red > 200 AND green > 200 AND blue > 200)
    STYLE
      COLOR 0 255 0
    END
  END
  # Использование регулярных выражений для выделения пикселей, чье значение
  заканчивается на "1"
  CLASS
    EXPRESSION /*1/
    STYLE
      COLOR 0 0 255
    END
  END
END
```

При работе с растрами больших размеров могут возникнуть проблемы, связанные с медленным отображением карт. Для увеличения быстродействия можно разрезать исходный растр на несколько небольших кусочков и получить таким образом "мозаичный" растр. Тогда для отображения части карты уже не придется манипулировать одним большим растром, а нужно будет собрать нужную часть карты из отдельных кусочков. Тем не менее, для MapServer такая мозаика по-прежнему будет выглядеть одним-единственным слоем.

Делается это следующим образом: создается shp-файл, хранящий полигоны - границы, по которым растр был разрезан на части. При этом в атрибутах каждого полигона хранится название файла, в котором содержится соответствующий кусочек растра.

Простейший пример:

```
LAYER
  NAME "hpool"
  STATUS ON
```

```

TILEINDEX "hp2.shp" # название shp-файла, содержащего границы кусочков
                    # (путь прописывается относительно SHAPERATH)
TILEITEM "Location" # название поля, в котором хранятся имена файлов с кусочками
растра
TYPE RASTER
END

```

Вообще, операцию создания индексного shp-файла можно проделать автоматически при помощи программы `gdaltindex` (часть пакета GDAL). Параметры использования:

```
gdaltindex -tileindex имя_поля имя_индексного_файла имя_растров*
```

Например:

```
gdaltindex wm_index.shp WorldMap/*.tif
```

При этом, если файл `wm_index.shp` не существовал, то он будет создан. Если не был указан параметр `-tileindex`, то будет использоваться название поля по умолчанию - `Location` ([полный перевод](#) главы про работу с растрами из описания Mapserver).

6. Проекция

Для работы с проекциями MapServer использует библиотеку PROJ.4. Описания проекций состоят из набора специальных ключевых слов которые понимает PROJ.4, также вместо описания проекции может указываться ссылка на соответствующее готовое описание в файле `epsg`. Этот файл является частью дистрибутива PROJ и соответственно, Mapserver.

Для работы с проекциями в map-файле предусмотрен [целый раздел](#), который так и называется PROJECTION. Для работы с проекциями необходимо создать описание проекции для карты в целом (объект MAP), также следует задать описания проекции слоям, которые изначально находятся в проекции отличной от выходной проекции карты. PROJECTION содержит параметры проекций и должен располагаться вне описаний слоев. Если предложение PROJECTION встречается внутри предложения LAYER, то это означает, что данный слой хранится на диске в этой проекции.

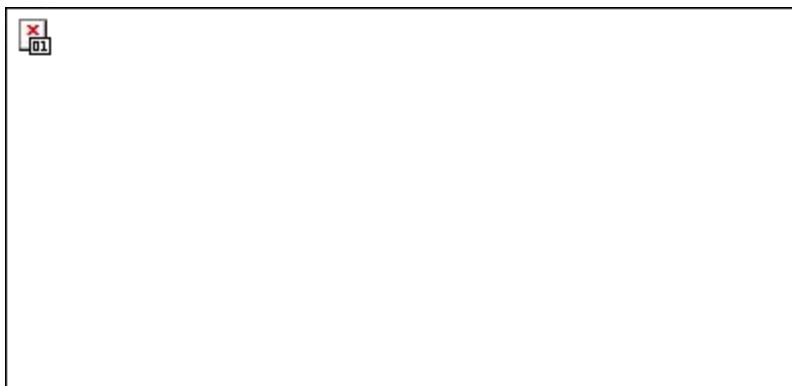
Для того, чтобы отобразить карту в географической системе координат (WGS84):

```

MAP
...
PROJECTION
  "proj=latlong" # проекция широта/долгота
  "datum=WGS84"  # WGS84
END

LAYER
...
END
END

```



Для того, чтобы отобразить карту в спроектированной системе координат (UTM, зона 36, WGS84):

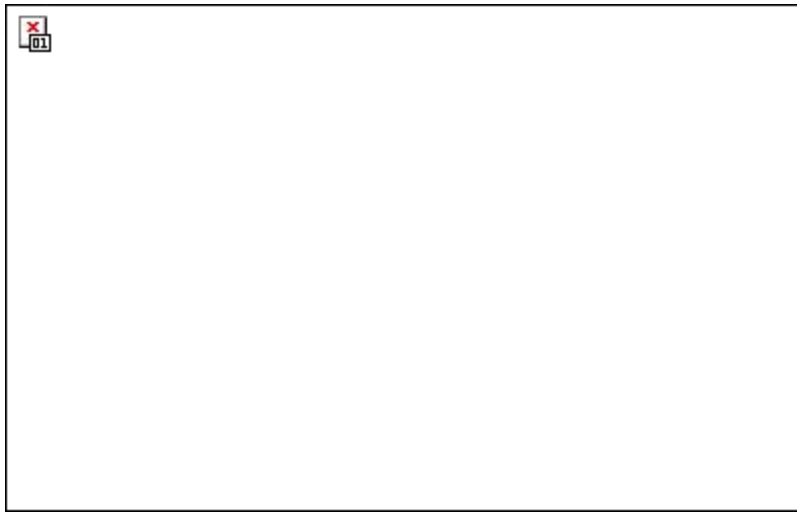
```
PROJECTION
    "proj=utm"
    "ellps=WGS84"
    "datum=WGS84"
    "zone=36"
    "units=m"
    "north"
    "no_defs"
END
```

Два важных момента: добавление предложения PROJECTION к MAP-объекту еще не означает, что вся карта тут же изменит проекцию - по умолчанию считается, что слои карты хранятся в той же проекции, которая указана в PROJECTION. Поэтому нужно добавить предложение PROJECTION и в слои карты (в предложение LAYER). Второе: после добавления предложения PROJECTION в map-файл нужно будет изменить параметр EXTENT так, чтобы угловые точки карты были представлены в новых координатах. Для определения новых координат можно воспользоваться программой proj для пересчета координат из географической в спроектированную систему координат (\ms4w\proj\bin\), или получить координаты в любой ГИС: Arcview, MapInfo и т.д.

```
MAP
# EXTENT          34.59 49.58 34.63 49.6          # старые координаты
EXTENT            615120 5493650 617760 5495760    # новые координаты
...
PROJECTION
    "proj=utm"
    "ellps=WGS84"
    "datum=WGS84"
    "zone=36"
    "units=m"
    "north"
    "no_defs"
END

LAYER
...
PROJECTION
    "proj=latlong"
    "datum=WGS84"
END
CLASS
...
END
END
...
END
```

То получим следующий результат:



Мар-файл, использованный в этом разделе: polt3.map.

7. Создание интерактивной карты

Сведений, приведенных в предыдущих разделах достаточно, чтобы нарисовать приличную на вид карту. Однако, у ней будет большой недостаток - ее не может изменить пользователь. Добавить интерактивности созданной карте достаточно просто, для этого нужно, во-первых, изменить один параметр в адресной строке: вместо "mode=map" нужно написать "mode=browse", и, во-вторых, написать небольшой html-шаблон, который будет содержать карту и элементы оформления\навигации.

Создадим папку templates и положим туда файл template.html следующего содержания:

```
<html>
<head>
<title>Пример карты</title>
</head>
<body>
<!-- Начало формы MapServer -->
<form name="mapserv" method="GET" action="/cgi-bin/mapserv.exe">
  <!-- Скрытые CGI переменные -->
  <input type="hidden" name="map" value="map">
  <input type="hidden" name="imgext" value="mapext">
  <input type="hidden" name="imgxy" value="center">
  <input type="hidden" name="layer" value="veget">
  <input type="hidden" name="layer" value="street">
  <input type="hidden" name="layer" value="grid">
  <input type="hidden" name="mode" value="browse">
  <table align="center" width="410" border="1">
  <tr>
    <td colspan="3" align="center" valign="top">
      <input type="image" name="img" src="img" width="400" height="300" border="0">
    </td>
  </tr>
  </table>
</form>
</body>
</html>
```

После этого создадим в мар-файле еще один раздел:

МАР

...

WEB

```
TEMPLATE    './templates/template.html'
```

```
IMAGEPATH   '/ms4w/tmp/ms_tmp/'
```

```
IMAGEURL     '/ms_tmp/'
```

END

LAYER

...

END

...

END

набираем в строке адреса:

```
http://localhost/cgi-bin/mapserv.exe?map=/ms4w/apps/example/polit4.map&layer=veget&layer=street&layer=grid&mode=browse
```

Итак, о том, что же произошло: строка `TEMPLATE './templates/template.html'` указывает, в каком каталоге находится шаблон и как он называется. После того, как пользователь ввел адрес (или кликнул по карте), MapServer обрабатывает этот шаблон: находит CGI переменные (заключенные в квадратные скобки) и заменяет их на значения этих переменных. Например, **map** будет заменено на `"/ms4w/apps/example/polit4.map"`, а **mapext** на `"615120 5493650 617760 5495760"`. Тэг **img** будет заменен на имя созданного временного файла, в котором хранится графическое изображение карты (к примеру, `"\ms4w\tmp\ms_tmp\MS11877753442492.gif"`). После этого вновь полученный html-файл отсылается пользователю.

Майл-файл, использованный в этом разделе: `polit4.map`. Html-шаблон: `template.html`

8. Инструменты масштабирования карты

В предыдущем примере при щелчке мышкой на карте, MapServer сдвигал изображение так, что точка в которую пришелся щелчек, оказывалась в центре карты. Кроме сдвигов, MapServer может производить и другие операции, например, масштабировать карту. Для этого необходимо изменить шаблон.

Вот содержимое файла `template1.html`:

```
<html>
<head>
<title>Пример карты</title>
</head>
<body>
<!-- Начало формы MapServer -->
<form name="mapserv" method="GET" action="/cgi-bin/mapserv.exe">
  <input type="hidden" name="map" value="map">
  <input type="hidden" name="imgext" value="mapext">
  <input type="hidden" name="imgxy" value="center">
  <input type="hidden" name="layer" value="veget">
  <input type="hidden" name="layer" value="street">
  <input type="hidden" name="layer" value="grid">
  <input type="hidden" name="mode" value="browse">
  <table align="center" width="680" border="0">
    <tr>
      <td>
        <table width="400" border="1">
          <tr>
```

```

        <td>
        <!-- SPECIFY MAP MODE -->
        <div align="center">Map Mode:<br>
            <select name="mode">
            <option value="browse">Browse</option>
            <option value="map">Map</option>
            </select>
        </div>
        </td>
        <td>
        <!-- FORM SUBMIT BUTTON -->
        <div align="center">
            <input type="submit" name="submit"
value="Refresh">
            </div>
        </td>
        <td>
        <!-- ZOOM/PAN CONTROLS -->
        <div align="center">Map Control: <br>
            <select name="zoom">
                <option value="4" zoom_4_select>Zoom In
4x</option>
                <option value="3" zoom_3_select>Zoom In
3x</option>
                <option value="2" zoom_2_select>Zoom In
2x</option>
                <option value="1"
zoom_1_select>Recenter</option>
                <option value="-2" zoom_-2_select>Zoom
Out 2x</option>
                <option value="-3" zoom_-3_select>Zoom
Out 3x</option>
                <option value="-4" zoom_-4_select>Zoom
Out 4x</option>
            </select>
        </div>
        </td>
    </tr>
    <tr>
    <!-- DISPLAY THE MAPSERVER-CREATED MAP IMAGE -->
        <td colspan="3" align="center" valign="top">
            <input type="image" name="img" src="img"
width="400" height="300" border="0">
            </td>
    </tr>
</table>
</td>
</tr>
</table>
</form>
</body>
</html>

```

Самое интересное происходит в разделе "ZOOM/PAN CONTROLS": там приводится пример того, как используется переменная "zoom". Если ее значение равно 1, то масштаб карты при нажатии кнопки "Refresh" не меняется, если значение переменной zoom положительное, то масштаб увеличивается в zoom раз. Если zoom<0 то масштаб карты уменьшается в zoom раз.

Мар-файл, использованный в этом разделе: polt5.map. Html-шаблон: template1.html

9. Масштабная линейка

Масштабную линейку легко добавить, создав соответствующий раздел map-файла:

```
MAP
...
SCALEBAR
  IMAGECOLOR 255 255 255
  LABEL
    COLOR 0 0 0
    SIZE TINY
  END
  STYLE 1
  SIZE 100 2
  COLOR 0 0 0
  UNITS METERS
  INTERVALS 2
  TRANSPARENT FALSE
  STATUS ON
END

LAYER
...
END
...
END
```

После этого нужно добавить в html-шаблон еще одну ячейку таблицы:

```
<td align="right"></td>
```

Назначение параметров LABEL, COLOR и др. - очевидно. Также очевидно назначение тега **scalebar** в html-шаблоне. Поэтому, единственная сложность, которая может возникнуть, - скорее эстетического характера, - хочется, чтобы шкала содержала "круглые" числа (1000м, 500м и т.п.). И для того, чтобы шкала получалась именно такая, приходится сначала подобрать соответствующий EXTENT, добиваясь появления подходящей шкалы, а потом разрешить пользователю выполнять только "подходящий" zoom (увеличение масштаба или в 2, или в 4 раза).

Map-файл, использованный в этом разделе: polt6.map. Html-шаблон: template2.html

10. Управление слоями карты

MapServer позволяет включать и выключать слои делая их видимыми и невидимыми, т.е. можно указывать, какие слои будут показаны на карте, а какие нет. Об этом подробно написано [вот здесь](#).

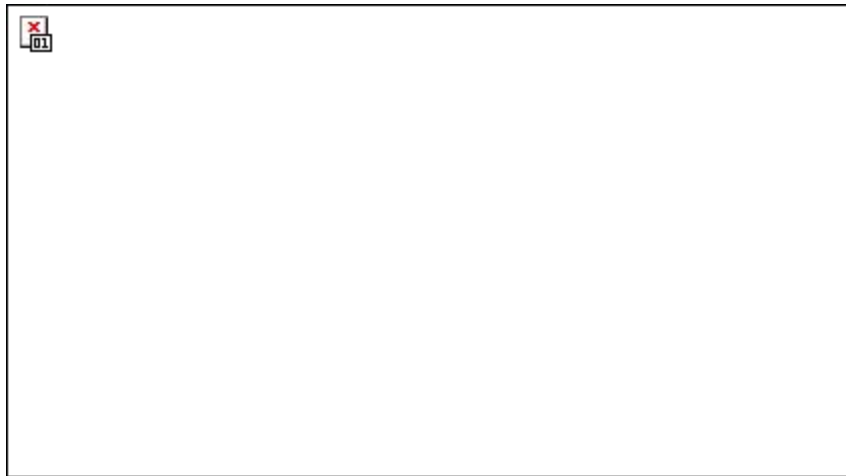
Если коротко, то суть в следующем: в html-шаблон вместо строчек вида:

```
<input type="hidden" name="layer" value="veget">
```

нужно вставить строки вида:

```
<input type="checkbox" name="layer" value="veget" veget_check> Растительность
```

вместо **veget_check** в общем случае используется тег **layername_check**, где layername - имя слоя, описанного в map-файле. Этот тег, в зависимости от того, поставлена ли галочка в соответствующем checkbox, при обработке будет заменен или строкой "checked", или пустой строкой (""). И, соответственно, слой на карте будет сделан или видимым, или прозрачным.

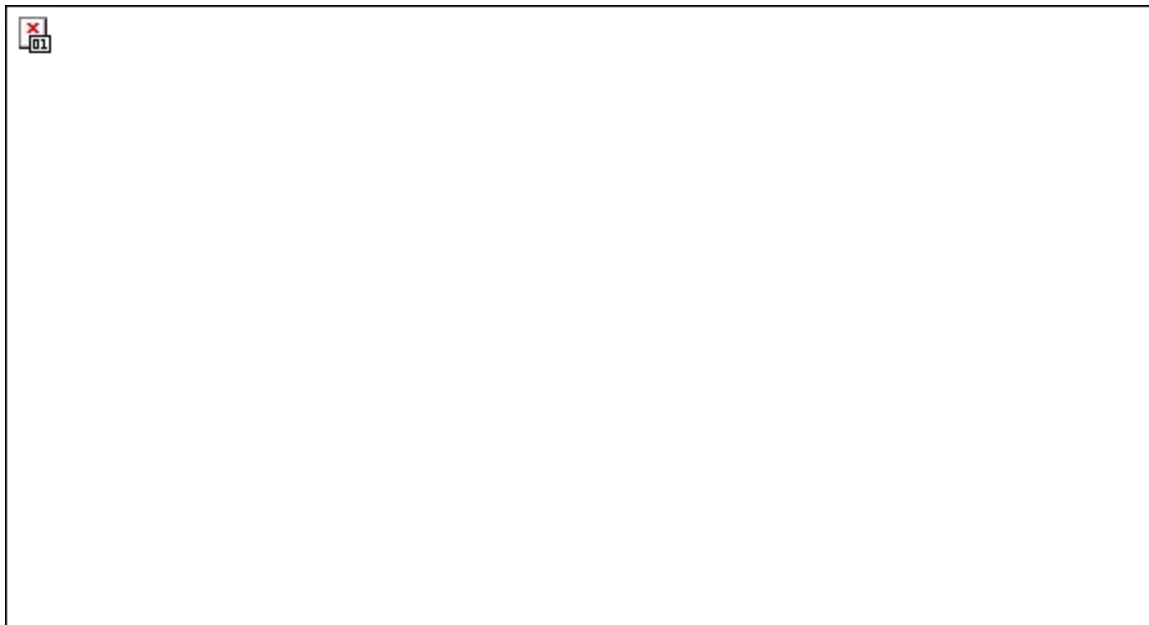


Есть еще один важный момент: если в map-файле, при описании слоя в предложении LAYER установить параметру STATUS значение DEFAULT, то этот слой будет показываться в любом случае, независимо от того, отмечен ли соответствующий слой флажком в форме запроса.

Мар-файл, использованный в этом разделе: polt5_1.map. Html-шаблон: template1_2.html

11. Добавление обзорной карты

MapServer позволяет создать обзорную карту - уменьшенную копию исходной карты, на которой будет показываться текущий EXTENT.



Чтобы получить такую карту, нужно создать еще один раздел в map-файле:

MAP

...

REFERENCE

IMAGE './images/ref.gif' # файл, в котором хранится изображение обзорной карты

SIZE 155 105 # размер обзорной карты в пикселях

EXTENT 615120 5493650 617760 5495760

координаты углов обзорной карты


```

STATUS ON

MINBOXSIZE 10 # Наименьший возможный размер (в пикселях) прямоугольника,
               # отражающего текущий EXTENT н обзорной карте.
               # Если размеры прямоугольника будут меньше, то вместо
               # него будет показана точка (см. MARKER)

MAXBOXSIZE 150 # Максимальный возможный размера прямоугольника

COLOR -1 -1 -1 # Цвет заливки прямоугольника
               # отрицательные значения - заливки нет.

OUTLINECOLOR 128 0 0 # Цвет границы прямоугольника

MARKERSIZE 8 # Размеры маркера

MARKER 'star' # Символ для маркера

END

LAYER

...

END

...

END

```

А для того, чтобы показать обзорную карту, в html-шаблон нужно вставить еще одну ячейку таблицы:

```

<td>

  <p>Обзорная карта:<br>

  <input type="image" name="ref" src="ref" border="0"></p>

</td>

```

Однако, для того, чтобы обзорную карту можно было показать, нужно сначала самостоятельно создать графический файл с этой картой ('./images/ref.gif').

Мар-файл, использованный в этом разделе: polt7.map. Html-шаблон: template3.html

12. Создание легенды

MapServer может автоматически создавать легенды к картам.



Для этого нужно добавить еще один раздел в map-файл и обновить html-шаблон:

```
MAP
...
  LEGEND
    KEYSIZE 12 12
    LABEL
      TYPE TRUETYPE
      COLOR 0 0 89
      SIZE 9
      FONT arial
      ENCODING 'Windows-1251'
    END
    STATUS ON
  END

  LAYER
    ...
  END
...
END
```

```
<td>
  <p>Обзорная карта:<br>
  </p>
  <p>Легенда:<br>
  </p>
</td>
```

Map-файл, использованный в этом разделе: polt8.map. Html-шаблон: template4.html

13. Запросы

Пришло время рассмотреть еще одну возможность, которую предоставляет MapServer - пространственные запросы. Простейший запрос можно выполнить, щелкнув мышкой на каком-либо объекте. В результате, используя html-шаблоны, MapServer сгенерирует страничку, на которой будет дана информация об этом объекте.

Допустим, нас интересует информация о названии улицы. Тогда к соответствующему описанию LAYER нужно добавить четыре параметра:

```

LAYER
  NAME          street
  DATA         Poltava10_Street_polyline
  ...
  TOLERANCE 5   # точность "попадания" - 5 пикселей
  HEADER './templates/templ_head.html'   # местонахождение
  FOOTER './templates/templ_footer.html'  # шаблонов для генерации
  TEMPLATE './templates/templ_query.html' # ответов на запрос
  ...

```

TEMPLATE - местоположение и название файла, содержащего основной текст шаблона запроса. HEADER и FOOTER - соответственно шапка и завершение шаблона.

Содержание файла templ_head.html:

```

<b>Результаты запроса: названия улиц</b>
<p>
<table cellpadding=5 cellspacing=2 border=0>
<tr bgcolor=#CCCCCC>
<td bgcolor=#ffffff> </td>
<th>Тип</th>
<th>Название</th>
</tr>

```

Содержание файла templ_footer:

```

</table>
<p>Нажмите "НАЗАД", чтобы вернуться к карте</p>
<p>

```

Содержание файла templ_query.html:

```

<tr>
<td>lrm</td>
<td>CLASSNAMER</td>
<td>RUSNAME</td>
</tr>

```

CLASSNAMER и RUSNAME - названия полей в Poltava10_Street_polyline.dbf, а lrm - название переменной, которая содержит порядковый номер объекта.

Важное замечание: хотя названия полей в dbf-файле могут быть записаны строчными буквами, в шаблоне обязательно нужно использовать заглавные.

Map-файл, использованный в этом разделе: polt9.map. Html-шаблоны: template5.html, templ_footer.html, templ_head.html, templ_query.html.

Заключение

Данная статья демонстрирует лишь некоторые из возможностей MapServer по оформлению картографической информации и публикации ее в интернет. Мы планируем более подробно рассказать об этих и других возможностях MapServer в следующих статьях. Если у Вас есть свой опыт, которым Вы хотели бы поделиться с пользователями, мы с удовольствием поможем его опубликовать на нашем сайте.

[Обсудить в форуме](#) Комментариев — 30

Ссылки по теме

- [Создание картографических сервисов с использованием MapServer. Введение](#)

[Установка MapServer для Windows](#)

Последнее обновление: October 07 2010

Дата создания: 10.10.2007

Автор(ы): [Дмитрий Колесов](#)