

# Карты сайтов

Карта сайта (sitemap) позволяет легко собрать в одном месте ссылки на страницы Web-сайта, чтобы к ним могли обращаться как люди, так и поисковые механизмы. Обычно в карте сайта перечислены все или, по крайней мере, наиболее важные страницы сайта. Она удобна и для людей, поскольку облегчает навигацию по сайту, и для поисковых механизмов, т.к. они могут более быстро и полно просмотреть весь сайт.

В данной главе будут рассмотрены следующие темы:

- ❑ два вида карт сайтов: традиционные и карты для поисковых механизмов;
- ❑ стандарт карт сайтов Google XML;
- ❑ стандарт текстовых карт сайтов для Yahoo!;
- ❑ новый стандарт sitemaps.org, реализованный во всех поисковых механизмах.

Вы реализуете ASP.NET-код, который генерирует программным способом карты сайтов как для Google, так и для Yahoo!. Но начнем мы с рассмотрения традиционных карт сайтов.

## Традиционные карты сайтов

Традиционная карта сайта — это просто HTML-страница, содержащая ссылки на различные страницы Web-сайта. Как правило, эти ссылки разбиты на группы для облегчения их чтения. Такие карты сайта обычно создаются для облегчения навигации людям, но специалисты по поисковому маркетингу быстро поняли, что их побочным эффектом является облегчение просмотра сайта поисковыми агентами.

Раньше поисковые механизмы не погружались глубоко в иерархию Web-сайтов, и страница с картой сайта помогала им просмотреть “глубинные” страницы (т.е. до которых нужно добираться через множество страниц). В настоящее время эта проблема *обычно* не возникает (поисковые механизмы наловчились пробираться гораздо дальше в глубины сайтов), но карта сайта все равно может помочь быстрее просмотреть такие страницы. Кроме того, она может несколько повысить ранг этих страниц, поскольку содержит дополнительные внутренние ссылки.

Традиционные карты сайтов, так же как и карты для поисковых механизмов (см. следующий раздел), особенно удобны для указания страниц, на которые нет ссылок в других навигационных средствах Web-сайта. Например, на странице справки по карте сайта Google сказано: “Карты сайта особенно полезны там, где пользователи не могут добраться до всех областей Web-сайта с помощью обычного интерфейса”.

Традиционные карты сайтов создаются так же, как и любые другие Web-страницы. Их можно создать вручную или генерировать динамически с помощью ASP.NET. Ссылки на страницу с картой сайта должны присутствовать в области навигации, или в нижнем колонтитуле всех страниц Web-сайта — или, по крайней мере, домашней страницы. В крупных сайтах может иметь смысл создать несколько карт для различных разделов, поскольку мы не рекомендуем иметь на странице слишком много ссылок. Рекомендации по созданию внутренних ссылок и разбивке на страницы были даны в главе 6.

*Мы опять использовали этот туманный квалификатор — “слишком много”. Как обычно, конкретного определения этого термина не существует (все зависит от поискового механизма), но специалисты по поисковому маркетингу обычно упоминают максимальное количество в 50–100 ссылок на страницу.*

## Поисковые карты сайта

Поисковые карты сайта не предназначены для людей. Это специальные конструкции для облегчения просмотра Web-сайта поисковыми механизмами. Если на сайте добавлен или изменен Web-контент, до которого долго добираться стандартными средствами навигации, то может пройти много недель, прежде чем поисковый агент самостоятельно обратит внимание на эти изменения. Аналогично, если на Web-страницу нет ссылок в навигационной структуре сайта, она не будет просмотрена без дополнительной помощи.

Мы не знаем, насколько высокий ранг будет иметь такая покинутая страница, и мы все-таки рекомендуем в любом случае использовать традиционную карту сайта, т.к. она содержит внутреннюю ссылку, в то время как в карте сайта для поисковых механизмов таких ссылок нет.

Эту помощь и оказывают поисковые карты сайта. Подобные реализации имеются у Google и Yahoo. У MSN на момент написания этих строк ничего подобного не было. Но в конце этой главы описан стандарт, на который, в конце концов, перейдут все поисковые механизмы.

Поисковые карты сайтов *не* заменяют традиционное индексирование сайта, и оно выполняется, как и прежде. Но если поисковая система заметит изменение в такой карте, то она быстрее посетит указанные URL.

Понятно, что традиционные карты сайта во многом похожи на поисковые карты. Поскольку традиционная карта сайта хорошо заметна на Web-сайте, она просматривается часто. Поэтому помещение на страницу традиционной карты ссылки на глубоко размещенный контент решает большинство тех же задач. Хотя создание поисковых карт сайтов все-таки имеет смысл.

Например, можно сообщить Google приблизительную частоту, с которой ожидается появление изменений, или что на какой-то Web-странице появилось изменение с более поздней меткой времени. Можно даже “прозвонить” Google, чтобы сообщить об изменениях в традиционной карте сайта.

*Однако при наличии меток времени с истекшим сроком годности карта сайта может и повредить. Если вы хотите указывать метки времени, вам нужно аккуратно обновлять их при внесении изменений!*

Мы рекомендуем использовать как традиционные, так поисковые карты сайтов.

Существует и менее известное преимущество применения карт сайтов — снижение ущерба при краже контента. К сожалению, в Web имеются неприятные типы, которые без разрешения могут взять контент с вашего Web-сайта и поместить на своем.

Такие сайты имеют забавное прозвище “scraper site” (“поскребыши”), но если дело коснется вас, то вам будет не до смеха. Одна из наиболее трудных задач, с которой сталкиваются поисковые механизмы — это определение первоначального автора контента, копии которого имеются в нескольких местах. Как сказано в главе 5, поисковые механизмы стараются не помещать в свои индексы дублированный контент. Если вы будете отфильтрованы из результатов поиска из-за кражи вашего контента, справедливость будет очень трудно восстановить. Если подобный сайт с высоким рангом (а такие есть) будет проиндексирован с вашим контентом прежде вашего сайта, то *контент на вашем Web-сайте* может быть расценен как копия! Поскольку поисковые карты сайтов ускоряют индексирование Web-страниц, они могут помочь избежать некоторых неприятностей, связанных с кражей контента.

Протокол карт сайта Yahoo! не так популярен, как протокол Google, но все-таки в данной главе вы узнаете, каким образом создавать обе карты с помощью одного и того же кода ASP.NET. Так что лучше поддерживать оба формата, ведь на это не нужно много усилий. Поскольку протокол карт сайта Yahoo! использует лишь подмножество информации из протокола карт сайта Google, то при создании карты сайта Yahoo! такая лишняя информация (если будет обнаружена) просто будет проигнорирована.

*И Google, и Yahoo! поддерживают чтение новостных каналов в форматах RSS и Atom. Эти форматы могут пригодиться для блогов и некоторых систем управления контентом; зачастую они предоставляются такими приложениями по умолчанию. Проблема с подобными реализациями состоит в том, что в них лишь перечисляется самый свежий контент, а это приемлемо только для блогов. Если вы выполняете поисковую оптимизацию для блога, можете пропустить эту главу и воспользоваться возможностями работы с каналами, имеющимися в вашем приложении ведения блога. Кроме того, теоретически возможно создать RSS- или Atom-канал со всеми URL — наподобие карты сайта, но только для сайта, а не для блога. Однако, возможно, это не то, чего ожидает Yahoo! или Google, и поэтому мы не можем рекомендовать такой прием.*

## Использование карт сайтов Google

У Google имеется тщательно продуманный стандарт для составления карт сайтов. Он позволяет Web-мастерам предоставлять информацию в нескольких форматах, но предпочтительнее использовать формат на базе XML, разработанный Google. Google утверждает, что применение формата Google Sitemaps приводит к “более качественному индексированию, т.к. он позволяет сообщить дату последнего изменения страницы или частоту изменения этой страницы”. Более подробно о Google Sitemaps можно прочитать по адресу <http://www.google.com/webmasters/sitemaps/>. А по адресу <http://sitemaps.blogspot.com/> находится блог, посвященный Sitemaps.

Однако, как заявляет Google, “использование этого протокола не гарантирует, что ваши Web-страницы будут включены в поисковый индекс” и “использование этого протокола не влияет на ранг ваших страниц в Google”.

Для создания карты вашего сайта необходимо выполнить описанные ниже действия.

1. Создать учетную запись в Google, если у вас ее еще нет: <https://www.google.com/accounts/NewAccount>.
2. Создать файл карты сайта.
3. Указать карту сайта в своей учетной записи.
4. Проверить сайт. Для этого необходимо выполнить определенное изменение на сайте, чтобы Google удостоверился, что вы имеете право модифицировать сайт. Вам нужно будет добавить в сайт дескриптор meta или файл со случайным именем. Для использования возможностей Google Sitemap этот элемент должен постоянно присутствовать в вашем сайте.

Более подробное описание этой процедуры содержится по адресу <http://www.google.com/support/webmasters/bin/answer.py?answer=34592&topic=8482>.

*Служба Google Sitemaps позволяет также узнать о возможных проблемах, если они возникли при индексировании сайта; к ним относятся коды ошибок, которые возвратил сервер (404, 500 и т.д.), ошибки из-за работы сети и т.п. Он также может выдать список URL-адресов, исключенных в файле robots.txt, и различные статистические данные, полезные для анализа.*

Как только вы завершите процесс регистрации, вы можете создать в корневом каталоге Web-сайта файл карты сайта с именем sitemap.xml, а затем указать этот файл с помощью страницы Google Sitemaps. Файл sitemap.xml может иметь примерно такой вид:

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.google.com/schemas/sitemap/0.84">
  <url>
    <loc>http://www.cristiandarie.ro/</loc>
    <lastmod>2006-09-17</lastmod>
    <changefreq>weekly</changefreq>
    <priority>0.5</priority>
  </url>
  <url>
    <loc>http://www.cristiandarie.ro/books/</loc>
    <lastmod>2006-09-17</lastmod>
    <changefreq>weekly</changefreq>
    <priority>0.8</priority>
  </url>
  <url>
    <loc>http://www.cristiandarie.ro/forthcoming/</loc>
    <lastmod>2006-09-17</lastmod>
    <changefreq>weekly</changefreq>
    <priority>0.2</priority>
  </url>
</urlset>
```

Этот файл содержит элемент <url> для каждого URL-адреса, который нужно включить в индексирование. Подэлементы этого элемента имеют следующие значения.

- <loc> содержит URL.
- <lastmod> содержит дату последней модификации для указанного URL. Дата записывается в формате ISO 8601, т.е. ГГГГ-ММ-ДД. Стандарт поддерживает и ряд

альтернативных способов записи, а также указание времени. Хорошее описание ISO 8601 имеется по адресу <http://www.iso.org/iso/en/prods-services/popstds/datesandtime.html>.

- ❑ `<changefreq>` сообщает Google частоту изменения страницы. Возможны значения `always` (всегда — для страниц, которые изменяются при каждом запросе), `hourly` (каждый час), `daily` (раз в день), `weekly` (раз в неделю), `monthly` (раз в месяц), `yearly` (раз в год) и `never` (никогда).
- ❑ `<priority>` позволяет сообщить Google вашу оценку важности отдельных страниц сайта по сравнению с другими. Оценка должна быть числом от 0.0 до 1.0. Учтите, что элемент `<priority>` описывает относительную важность страниц *внутри* Web-сайта и *не* влияет на общий ранг Web-сайта!

## Использование карт сайтов Yahoo!

Протокол карт сайтов Yahoo! гораздо проще API Google. Он также поддерживает несколько форматов, в том числе и новостных каналов, но здесь мы рассмотрим формат в виде обычного текста со списком URL. В нем не используется XML, и не содержится никакая информация, кроме списка URL-адресов, разделенных переводами строки. Yahoo! требует, чтобы файл с именем `urllist.txt` находился в корневом каталоге Web-сайта, а, кроме того, требуется регистрация учетной записи Yahoo!.

Затем сайт следует добавить на странице <http://submit.search.yahoo.com/free/request>. Yahoo! выполняет примерно то же самое, что и Google, но без излишних сложностей: не требуется информация о датах последних изменений, оценка частоты изменения или относительная важность страниц. Кроме того, у Yahoo! нет возможности “прозвона” (ping) в случае обновления карты сайта.

Карта сайта Yahoo!, эквивалентная приведенной выше карте сайта Google, является текстовым файлом с именем `urllist.txt`, который находится в корневом каталоге Web-сайта и содержит следующие строки:

```
http://www.cristiandarie.ro/
http://www.cristiandarie.ro/books/
http://www.cristiandarie.ro/forthcoming/
```

Как и в случае Google, использование протокола карт сайтов Yahoo! не влияет на ранг Web-сайта, а просто помогает более быстро просмотреть его поисковому агенту.

## Извещение Google об изменениях

Вообще-то Google и так хорошо справляется с регулярным чтением карты сайта и с обнаружением всех обновлений; но вы все же можете сообщить Google, что карта вашего сайта изменилась, отправив запрос по следующему URL:

```
http://www.google.com/webmasters/sitemaps/ping?sitemap=http://seoasp/sitemap.xml
```

Если вы загрузите этот URL в свой Web-браузер, вы просто получите сообщение, что ваша карта сайта добавлена в очередь, и что вы должны зарегистрировать свою карту сайта на странице <http://www.google.com/webmasters/sitemaps>, если вы не сделали этого раньше (рис. 9.1).

Такой запрос можно легко создать и программным способом. Для этого нужно реализовать программную логику, когда указанный URL “прозванивается” при каждом изменении карты сайта для Google — например, при изменении какой-либо страницы с описанием продукта или страницы с оглавлением.

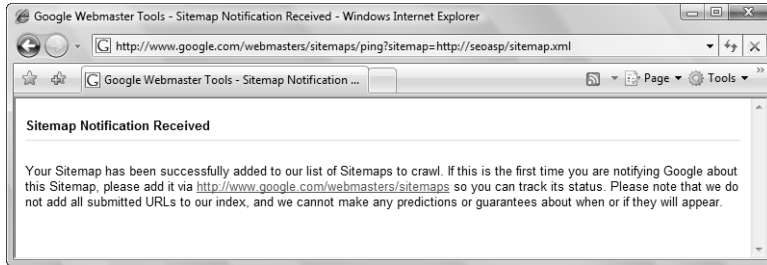


Рис. 9.1. Уведомление о карте сайта

## Стандартный протокол sitemaps.org

В последнее время появилась новая инициатива в деле стандартизации протокола карт сайтов для всех поисковых механизмов. Этот стандарт и информация о нем доступны по адресу <http://www.sitemaps.org/>. Он больше всего похож на стандарт Google, но с другим пространством имен XML:

```
<urlset xmlns="http://www.google.com/schemas/sitemap/0.84">
```

заменяется на

```
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
```

Кроме того, для применения этого протокола не требуется создавать какие-либо учетные записи на сайтах поисковых механизмов. Вместо этого нужно просто хотя бы раз “прозвонить” адрес этого поискового механизма и указать местоположение карты сайта (и, возможно, повторять при появлении обновлений):

```
<URL_поискового_механизма>/ping?sitemap=URL_карты_сайта
```

На момент написания этих строк данный стандарт поддерживает только один поисковый сайт — Google, который распознает запросы вроде

```
http://www.google.com/webmasters/sitemaps/ping?sitemap=www.example.com/sitemap.xml
```

В MSN такая возможность еще не реализована, а Yahoo! поддерживает указание карты сайта с помощью следующего (нестандартного) запроса:

```
http://search.yahooapis.com/SiteExplorerService/V1/updateNotification?appid=[YAHOO_ID_ВАШЕГО_ПРИЛОЖЕНИЯ]&url=http://www.example.com/sitemap.xml
```

Yahoo-идентификатор для вашего приложения можно получить по адресу [http://api.search.yahoo.com/webservices/register\\_application](http://api.search.yahoo.com/webservices/register_application).

А можно поместить URL карты сайта в файл `robots.txt` следующим образом:

```
Sitemap: http://www.example.com/sitemap.xml
```

Для получения наиболее свежей информации по этой теме посетите страницу обновлений для данной книги, которую ведет Кристиан Дари по адресу <http://www.cristiandarie.ro/seo-asp/>.

## Программная генерация карт сайтов

Было бы неплохо иметь библиотеку, которая может создавать карты сайтов для Yahoo! и Google программным путем, используя одну и ту же информацию. Такая библиотека демонстрируется в приведенном ниже упражнении.

Вы создадите класс с именем Sitemap, который может хранить несколько ссылок с вашего сайта и генерировать соответствующие файлы карты сайта для Yahoo! и Google.

### Генерация карт сайтов

1. Мы начнем это упражнение с проверки, имеются ли у вас все нужные компоненты из предыдущих глав. Во-первых, нужно, чтобы был установлен параметр SiteDomain в файле Web.config:

```
<configuration>
...
  <appSettings>
    <add key="SiteDomain" value="http://seoasp" />
  ...
```

2. В папке App\_Code должны находиться файлы LinkFactory.cs и SeoData.cs, которые были созданы в главах 4 и 5. Первый из них нужен для генерации URL-адресов для карт сайтов на основе данных о продуктах и категориях, а второй представляет собой фиктивную базу данных продуктов и категорий. В реальной ситуации SeoData.cs должен быть заменен базой данных с продуктами и категориями вашего приложения. (При необходимости вы можете взять нужные файлы со страницы загрузки кода для данной главы.)
3. Мы напишем обобщенный обработчик Sitemap.ashx, который будет генерировать данные для карт сайтов Google и Yahoo!. Запросы к sitemap.xml и urllist.txt будут перенаправляться в этот обобщенный обработчик. Если вы используете ISAPI\_Rewrite, то добавьте в файл httpd.conf следующие правила:

```
# Подмена запросов к sitemap.xml (карта сайта Google)
RewriteRule ^/sitemap.xml$ /Sitemap.ashx?Target=google [L]

# Подмена запросов к urllist.txt (карта сайтов Yahoo!)
RewriteRule ^/urllist.txt$ /Sitemap.ashx?Target=yahoo [L]
```

А если вы пользуетесь UrlRewriter.NET, то эти правила будут выглядеть так:

```
<!-- Подмена запросов к sitemap.xml (карта сайта Google) -->
<rewrite url="/sitemap.xml$" to="/Sitemap.ashx?Target=google"
processing="stop" />

<!-- Подмена запросов к urllist.txt (карта сайтов Yahoo!) -->
<rewrite url="/urllist.txt$" to="/Sitemap.ashx?Target=yahoo"
processing="stop" />
```

4. Глядя на только что написанные два правила подмены, легко понять, что запрос Sitemap.ashx?Target=google должен возвращать XML-структуру с картой сайта Google, а Sitemap.ashx?Target=yahoo — текст со списком URL-адресов. Сейчас мы напишем код для этой цели. Добавьте в корень вашего проекта файл обобщенного обработчика Sitemap.ashx и поместите в него следующий код:

```
<%@ WebHandler Language="C#" Class="Sitemap" %>

using System;
using System.Web;
using System.Xml;
using System.Configuration;
using System.Collections.Generic;

public class Sitemap : IHttpHandler
{
    // структура для представления элемента карты сайта
    struct SitemapItem
    {
        // общедоступные поля свойств элемента карты сайта
        public string Url, LastMod, ChangeFreq, Priority;

        // конструктор, инициализирующий поля
        public SitemapItem(string url,
            string lastMod, string changeFreq, string priority)
        {
            this.Url = url;
            this.LastMod = lastMod;
            this.ChangeFreq = changeFreq;
            this.Priority = priority;
        }
    }

    // точка входа - метод, вызываемый при загрузке обработчика
    public void ProcessRequest(HttpContext context)
    {
        // получение значения параметра Target
        string target = context.Request.QueryString["Target"];

        // если такого параметра нет - выход
        if (target == null) return;

        // генерация запрошенной карты сайта
        if (target.ToLower() == "google")
        {
            GenerateGoogleSitemap();
        }
        else if (target.ToLower() == "yahoo")
        {
            GenerateYahooSitemap();
        }
    }

    // генерация карты сайта для Google
    private void GenerateGoogleSitemap()
    {
        // получение текущего объекта HttpResponse
        HttpResponse response = HttpContext.Current.Response;

        // указание типа контента
        response.ContentType = "text/xml";

        // генерация карты сайта Google с помощью XmlWriter
        XmlWriter xmlWriter = XmlWriter.Create(response.OutputStream);
```



```
// запись первого элемента
xmlWriter.WriteStartElement("urlset",
    "http://www.google.com/schemas/sitemap/0.84");

// получение списка элементов карты
List<SitemapItem> sitemapItems = GetSitemapItems();

// генерация элементов карты
foreach (SitemapItem sitemapItem in sitemapItems)
{
    // генерация элемента <url> и его содержимого
    xmlWriter.WriteStartElement("url");
    xmlWriter.WriteElementString("loc", sitemapItem.Url);
    xmlWriter.WriteElementString("lastmod", sitemapItem.LastMod);
    xmlWriter.WriteElementString("changefreq", sitemapItem.ChangeFreq);
    xmlWriter.WriteElementString("priority", sitemapItem.Priority);
    xmlWriter.WriteEndElement();
}

// закрытие документа
xmlWriter.WriteEndElement();
xmlWriter.Flush();
}

// генерация карты сайта для Yahoo
private void GenerateYahooSitemap()
{
    // получение текущего объекта HttpResponse
    HttpResponse response = HttpContext.Current.Response;

    // указание типа контента
    response.ContentType = "text/plain";

    // получение списка элементов карты
    List<SitemapItem> sitemapItems = GetSitemapItems();

    // генерация элементов карты
    foreach (SitemapItem sitemapItem in sitemapItems)
    {
        response.Write(sitemapItem.Url + "\n");
    }
}

// создание списка элементов, которые нужно включить в карту сайта
private List<SitemapItem> GetSitemapItems()
{
    // объявление списка элементов карты
    List<SitemapItem> sitemapItems = new List<SitemapItem>();

    // создание списка URL-адресов для включения в карту
    sitemapItems.Add(new SitemapItem(
        ConfigurationManager.AppSettings["SiteDomain"] + "/Catalog.html",
        "2007/05/05", "daily", "0.3"));

    sitemapItems.Add(new SitemapItem(
        LinkFactory.MakeCategoryProductUrl(SeoData.Categories["12"], 12,
        SeoData.Products["45"], 45), "2007/05/05", "weekly", "0.6"));

    sitemapItems.Add(new SitemapItem(
        LinkFactory.MakeCategoryProductUrl(SeoData.Categories["2"], 2,
        SeoData.Products["42"], 42), "2007/05/05", "weekly", "0.6"));
}
```

```

sitemapItems.Add(new SitemapItem(
    LinkFactory.MakeCategoryProductUrl(SeoData.Categories["6"], 6,
        SeoData.Products["15"], 15), "2007/05/05", "weekly", "0.6"));

sitemapItems.Add(new SitemapItem(
    LinkFactory.MakeCategoryProductUrl(SeoData.Categories["6"], 6,
        SeoData.Products["31"], 31), "2007/05/05", "weekly", "0.6"));

// возврат списка объектов Sitemap
return sitemapItems;
}

public bool IsReusable
{
    get
    {
        return false;
    }
}
}

```

5. Загрузите файл <http://seoasp/sitemap.xml>. Вы должны получить карту сайта для Google, показанную на рис. 9.2.
6. Загрузите файл <http://seoasp/urllist.txt>. Вы должны получить карту сайта для Yahoo!, показанную на рис. 9.3.

Итак, обобщенный обработчик `Sitemap.ashx` умеет создавать карты сайтов и для Yahoo!, и для Google. Как он это делает? Текст класса `Sitemap` начинается с определения структуры `SitemapItem` (о концепции структур было сказано в главе 7).

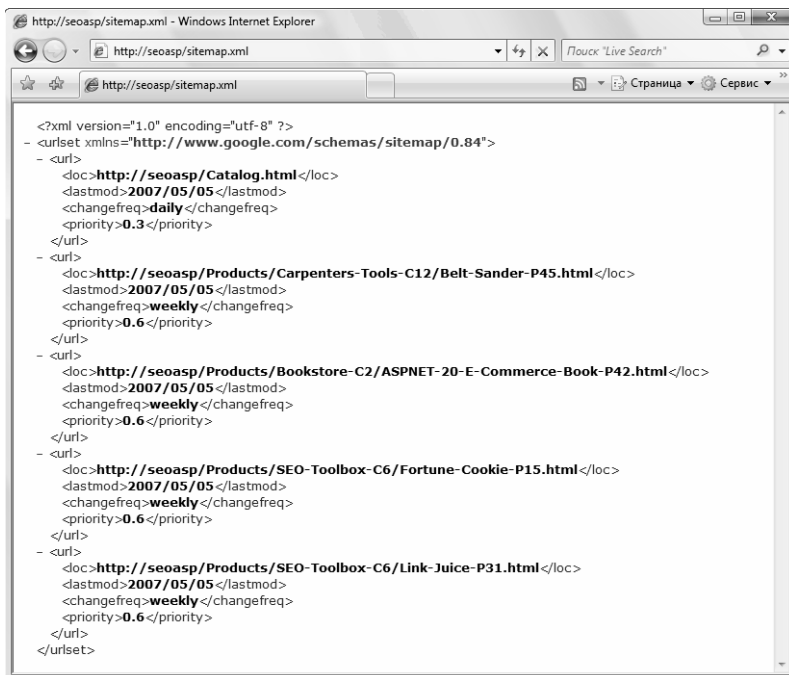


Рис. 9.2. Сгенерированная карта сайта для Google

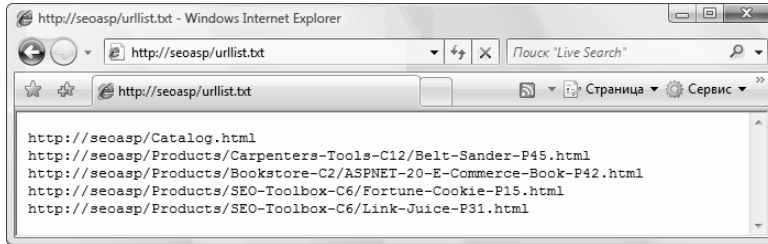


Рис. 9.3. Сгенерированная карта сайта для Yahoo!

Объекты этой структуры представляют собой элементы карты сайта, а их информация хранится в четырех общедоступных полях (Url, LastMod, ChangeFreq и Priority). У данной структуры имеется также конструктор для инициализации этих полей.

*Вообще-то обычно рекомендуется хранить информацию в частных полях и предоставлять доступ к ней с помощью общедоступных свойств. Но по отношению к структурам это правило не так строго, и мы решили не следовать ему, чтобы сделать код более коротким и понятным.*

Как уже было сказано раньше, при запросе .ashx-файла вызывается его метод `ProcessRequest()`, генерирующий выходные данные. В нашем случае этот метод читает из строки запроса параметр `Target` и в зависимости от его значения вызывает нужный метод для генерации запрошенной карты сайта:

```
// генерация запрошенной карты сайта
if (target.ToLower() == "google")
{
    GenerateGoogleSitemap();
}
else if (target.ToLower() == "yahoo")
{
    GenerateYahooSitemap();
}
```

Методы `GenerateGoogleSitemap()` и `GenerateYahooSitemap()` перебирают список элементов карты, возвращенный методом `GetSitemapItems()`, и создают из него выходные данные. Код создания карты сайты для Google ощутимо длиннее, т.к. она должна содержать больше информации для каждого URL, причем в формате URL. Для облегчения генерации XML-документа мы использовали класс `XmlWriter`, который выполняет за нас значительную часть реализации. Мы не будем подробно разбирать код, т.к. он достаточно понятен без всяких объяснений.

*Классы `XmlReader` и `XmlWriter` идеальны для выполнения простых действий с XML-документами. Они проще и понятнее, чем класс `Xml`, а также менее подвержены ошибочному использованию и более удобны, чем самостоятельная работа с XML-документами.*

Метод `GetSitemapItems()` возвращает объект `List<SitemapItem>` с помощью обобщений (generics) — новой возможности в .NET 2.0. Она была кратко описана в главе 6. В данном случае эту концепцию понять еще проще: `List<SitemapItem>` можно рассматривать как тип данных с “необычным” именем, который представляет собой коллекцию объектов `SitemapItem`.

*Если вы работаете в ASP.NET 1.1, вам нужно либо создать собственную коллекцию объектов `SitemapItem`, либо воспользоваться одним из встроенных классов коллекций для хранения объектов `Object` – таких как `ArrayList`. Например, код будет замечательно работать, если заменить в нем все упоминания `List<Sitemap>` на `ArrayList`, хотя это и не так эффективно.*

При создании списка элементов мы воспользовались методом `LinkFactory.MakeCategoryProductUrl()` для получения URL-адресов продуктов – генератор ссылок пригодился в очередной раз. Названия продуктов и категорий выбираются из класса `SeoData`, который заменяет базу данных.

В главе 14, при разработке сайта электронной коммерции, у нас будет более реальная ситуация, когда данные о продуктах и категориях будут читаться из базы данных, а не из жестко закодированного класса.

## Резюме

В данной главе вы познакомились с двумя форматами карт сайтов – традиционным и ориентированным на поисковые механизмы. Карты сайтов для поисковых механизмов просматриваются только поисковыми механизмами и помогают более быстро и полно выполнить индексирование Web-сайта. Традиционные карты сайтов предназначены для просмотра людьми, но они помогают и поисковым механизмам.

Можно создать обе формы карт сайта, которые не будут мешать друг другу. Правильно спроектированная карта сайта также повышает доступность информации для людей, чего не скажешь о карте сайта для поисковых механизмов. Мы рекомендуем создавать оба вида карт, чтобы повысить как доступность, так и скорость и полноту индексирования.