Endeca® Sitemap Generator

Developer's Guide

Version 2.1.0 • December 2011

Contents

Copyright and disclaimer

Preface

Endeca® InFront enables businesses to deliver targeted experiences for any customer, every time, in any channel. Utilizing all underlying product data and content, businesses are able to influence customer behavior regardless of where or how customers choose to engage — online, in-store, or on-the-go. And with integrated analytics and agile business-user tools, InFront solutions help businesses adapt to changing market needs, influence customer behavior across channels, and dynamically manage a relevant and targeted experience for every customer, every time.

InFront Workbench with Experience Manager provides a single, flexible platform to create, deliver, and manage content-rich, multichannel customer experiences. Experience Manager allows non-technical users to control how, where, when, and what type of content is presented in response to any search, category selection, or facet refinement.

At the core of InFront is the Endeca MDEX Engine,™ a hybrid search-analytical database specifically designed for high-performance exploration and discovery. InFront Integrator provides a set of extensible mechanisms to bring both structured data and unstructured content into the MDEX Engine from a variety of source systems. InFront Assembler dynamically assembles content from any resource and seamlessly combines it with results from the MDEX Engine.

These components — along with additional modules for SEO, Social, and Mobile channel support — make up the core of Endeca InFront, a customer experience management platform focused on delivering the most relevant, targeted, and optimized experience for every customer, at every step, across all customer touch points.

About this guide

This guide describes the Endeca Sitemap Generator and provides instructions for using it to generate sitemaps for an Endeca application.

It assumes that you are familiar with Endeca's terminology and basic concepts. This guide covers only the features of the Endeca Sitemap Generator, and is not a replacement for the available material documenting other Endeca products and features.

If you are using the Sitemap Generator in conjunction with the Endeca URL Optimization API for either Java or the RAD Toolkit for ASP.NET, please read the appropriate URL Optimization API guide in addition to this guide:

- *Endeca URL Optimization API for Java Developer's Guide*
- *Endeca URL Optimization API for the RAD Toolkit for ASP.NET Developer's Guide*

Note: The *Endeca Sitemap Generator Developer's Guide* is not a replacement for the URL Optimization API documentation.

Who should use this guide

This guide is intended for developers who are building applications that leverage the Endeca Sitemap Generator.

This document assumes that the reader has a working knowledge of the following software and concepts:

- Basic Endeca concepts such as dimensions, dimension values, refinements, ancestors, records, aggregate records, and so on
- Configuring Endeca dimensions using Developer Studio
- Tthe Endeca Presentation API, specifically:
  - `UrlGen` class
  - `ENEQueryToolkit` class
  - Guided Navigation classes, such as `DimVal`, `Dimension`, `DimLocation`, and so on

Conventions used in this guide

This guide uses the following typographical conventions:

Code examples, inline references to code elements, file names, and user input are set in `monospace` font. In the case of long lines of code, or when inline monospace text occurs at the end of a line, the following symbol is used to show that the content continues on to the next line: ¬

When copying and pasting such examples, ensure that any occurrences of the symbol and the corresponding line break are deleted and any remaining space is closed up.

Contacting Endeca Customer Support

The Endeca Support Center provides registered users with important information regarding Endeca software, implementation questions, product and solution help, training and professional services consultation as well as overall news and updates from Endeca.

You can contact Endeca Standard Customer Support through the Support section of the Endeca Developer Network (EDeN) at *http://eden.endeca.com*.

Chapter 1

Introduction

This section provides an introduction to the Sitemap Generator and its capabilities.

About sitemaps

A sitemap provides search engine spiders with information about all of the content available on the site, and allows them to access every specified page without crawling the site links.

Ensuring that site content is included in Web search indices, such as Google and Yahoo, is a common challenge facing websites. When the site in question contains a small set of dynamic or static pages, this is a straightforward process; however, the introduction of Endeca's Guided Navigation creates a combinatoric set of dynamic "pages" with a significant amount of overlapping content — where the same content (products, documents, etc.) is listed on pages that can be accessed by multiple distinct URLs. This creates a challenge for spiders to find and index the desired content simply by crawling link-by-link.

Providing an alternative starting point for spiders, such as a sitemap, can help the indexing process. Think of it as a special home page designed just for spiders, pointing to important content within the end-user site (but not accessible to the end user). Google, Yahoo!, Microsoft, and Ask.com have agreed to support the same XML Sitemap protocol. You can find out more about sitemaps and the sitemap protocol at *http://www.sitemaps.org*.

About the Endeca Sitemap Generator

The Sitemap Generator is a standalone Java application that builds a set of index pages containing links to all product detail pages as well as select navigation pages, static pages, and search results pages.

The Sitemap Generator retrieves the necessary record and dimension data by issuing a single bulk-export query against an MDEX engine. It then creates index pages using customizable templates to support different page formats (such as the Sitemap protocol XML format). In situations where one page can be referenced using different URLs, such as pages with multiple dynamic parameters in URL that can be transposed, the Sitemap Generator only includes one version of the URL in the sitemap. Doing so reduces the possibility of duplicate content in the search engine indices.

In order to encourage indexing of important site content in Web search indices, it is advantageous to provide links to pages resulting from common searches. The Sitemap Generator is capable of building

these links in a customized fashion. To use this functionality, you can supply a list of your site's most common search terms to the Sitemap Generator.

The Sitemap Generator also supports the creation of links to pre-existing static pages. This enables you to ensure that Web crawlers can access non-Endeca powered pages.

Chapter 2

Installing the Endeca Sitemap Generator

This section provides information about prerequisites, version compatibilities, and installation procedures.

System requirements

This section provides system requirements for installing the Endeca Sitemap Generator.

Endeca core requirements

To determine the compatibility of the Sitemap Generator with other Endeca installation packages, see the *Endeca InFront Compatibility Matrix* available on EDeN.

The following Endeca components must be installed on the same machine as the Sitemap Generator:

- **Platform Services.** The Sitemap Generator requires the `ENDECA_ROOT` variable to be set in order to run.
- **Application Controller Central Server and Agent** or **Application Controller Agent**. The Sitemap Generator scripts use the Servlet API installed with the Endeca platform, located in `%ENDECA_ROOT%\tools\server\common\lib\servlet-api.jar`.
- **Presentation API for Java**. The Sitemap Generator scripts use the Java Presentation API installed with the Endeca platform, located in `%ENDECA_ROOT%\lib\java\endeca_navigation.jar`.

These components are part of the Platform Services installation.

The sample configuration files provided with the Sitemap Generator are tailored for the standard wine data set. Endeca recommends that you have an MDEX Engine configured and running with this data set before you begin the installation procedure. Please see the *Endeca Getting Started Guide* for information on how to set up an MDEX Engine with this data set using the Deployment Template.

Endeca URL Optimization API

The Sitemap Generator produces links with search-engine optimized URLs. In order to ensure that the sitemap links resolve, Endeca recommends installing the URL Optimization API and integrating it with your application prior to installing the Sitemap Generator. The URL Optimization API can be downloaded from the Endeca Developer Network (EDeN).

Hardware and operating system requirements

The Endeca Sitemap Generator is supported for all hardware and operating system platforms that are supported by Endeca Platform Services.

Java requirements

The Sitemap Generator is a standalone Java application that requires Java 1.5 or later. By default, the Sitemap Generator scripts use the JDK installed with the Endeca platform, located in `%ENDECA_ROOT%\j2sdk\bin\java.exe`. If the `ENDECA_ROOT` environment variable is not found, then the Sitemap Generator uses the JDK specified in the `JAVA_HOME` environment variable.

If you have a large index and the Sitemap Generator runs into memory issues, Endeca recommends that you use a 64-bit JDK on a 64-bit operating system. Note that although a 32-bit JDK works on a 64-bit operating system, a 64-bit JDK does not work on a 32-bit operating system.

Installing the Sitemap Generator

The Sitemap Generator is distributed as a zip file (`sitemapGenerator-`*`version`*`.zip`) that is a self-contained tree.

The file can be unpacked at any location using WinZip, or any other compression utility that supports this format. Unpacking the file creates the subdirectory structure `Endeca/SEM/SitemapGenerator/`*`version`* .

To install the Sitemap Generator, extract the zip package using WinZip or an alternate decompression utility.

Endeca recommends that you install the Sitemap Generator to the same directory as your Endeca installation. For example:

Platform Services directory:	Extract ZIP archive to:
C:\Endeca\PlatformServices*version*\	C:\
/usr/local/endeca/PlatformServices/*version*/	/usr/local/

Package contents and directory structure

This section provides a reference list of the directories created by the Sitemap Generator.

The `SitemapGenerator/`*`version`*`/` directory contains the following subdirectories:

File/Directory	Purpose
`bin`	This directory contains the `.bat` and `.sh` scripts used to run the Sitemap Generator from a command line.
`conf`	This directory contains all files necessary to configure the Sitemap Generator before it is run. For details on the configuration files, see the section on Configuration.
`doc`	This directory contains the release notes and several sample sitemaps (in the `samples` subdirectory) that have been generated using different configuration settings.
`lib`	This file contains the Sitemap Generator classes packaged in `sitemapGenerator.jar`, which must be included in the Java classpath when running the Sitemap Generator.

Related Links

Configuration files on page 21

This section provides an overview of the configuration files used by the Sitemap Generator.

Chapter 3

Running the Sitemap Generator

This section describes the process for running the Sitemap Generator and provides information about the various output types.

Running the Sitemap Generator from the command line

The Sitemap Generator can be run from the command line using `\bin\RunSitemapGen.bat` or `/bin/RunSitemapGen.sh`.

These scripts take the location (absolute or relative) of the main configuration file, as a single argument. For example, when running on Windows:

```
cd C:\Endeca\SEM\SitemapGenerator\2.0.0\bin
RunSitemapGen.bat ..\conf\conf.xml
```

Note that the above example changes to the `\bin` directory before running the script. While this is convenient for specifying the `conf.xml` file via a relative path, the `bin` directory is not required as a working directory.

These scripts rely on the `ENDECA_ROOT` environment variable being specified in order to locate the appropriate Java binaries and Endeca Presentation API for Java. If necessary, alternate locations can be specified by modifying the `NAVIGATION_API` or `JAVA` variables in these scripts. The location of these scripts is also used to determine the location of the Sitemap Generator classes. If these scripts are moved from the `bin` directory, the `SITEMAP_GEN` variable must be modified as well.

Related Links

The main configuration file on page 21
The main configuration file for the Sitemap Generator is located in `conf/conf.xml`.

About validating sitemaps on page 37
Before deploying your sitemaps, you should ensure that all the generated links are valid.

Standard output

This section provides an example of a successful standard output.

A successful run of the Sitemap Generator should produce a standard output similar to the following:

```
Jul 29, 2009 11:31:36 AM com.endeca.soleng.sitemap.SitemapMain LoadConfigu¬
ration
```

```
INFO: Loading config file...
Jul 29, 2009 11:31:36 AM com.endeca.soleng.sitemap.SitemapMain LoadConfigu¬
ration
INFO: Clearing old sitemap files...
Jul 29, 2009 11:31:36 AM com.endeca.soleng.sitemap.SitemapMain LoadConfigu¬
ration
INFO: Loading templates...
Jul 29, 2009 11:31:36 AM com.endeca.soleng.sitemap.SitemapMain LoadUrlFor¬
matSettings
INFO: Load URL Formatting Settings...
Jul 29, 2009 11:31:36 AM org.springframework.beans.factory.xml.XmlBeanDefi¬
nitionReader loadBeanDefinitions
INFO: Loading XML bean definitions from file [C:\Endeca\SEM\SitemapGenera¬
tor\2.0.0\bin\..\conf\urlconfig.xml]
Jul 29, 2009 11:31:37 AM com.endeca.soleng.sitemap.SitemapMain runQueries
INFO: Querying engine...
Jul 29, 2009 11:31:43 AM com.endeca.soleng.sitemap.SitemapMain execute
INFO: Writing detail links...
Jul 29, 2009 11:31:53 AM com.endeca.soleng.sitemap.SitemapMain execute
INFO: Writing navigation links...
Jul 29, 2009 11:31:56 AM com.endeca.soleng.sitemap.SitemapMain execute
INFO: Writing search term links...
Jul 29, 2009 11:31:58 AM com.endeca.soleng.sitemap.SitemapMain execute
INFO: Writing static page links...
Jul 29, 2009 11:31:59 AM com.endeca.soleng.sitemap.SitemapMain execute
INFO: Writing index file...
Jul 29, 2009 11:31:59 AM com.endeca.soleng.sitemap.SitemapMain writeIndexFile
INFO: Sitemap files output here: C:\Endeca\SEM\SitemapGenera¬
tor\2.0.0\sitemap\*.xml
Script completed successfully.
```

File outputs

The following sections discuss each of the files that are output by the Sitemap Generator. By default, these files are saved in the `sitemap` directory, but this location can be configured in the main configuration file.

Index file

The Index file contains links to each of the other files that the Sitemap Generator produces.

For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
  <sitemapindex xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <sitemap>
    <loc>http://localhost:8888/urlformatter_jspref/detail0.xml</loc>
  </sitemap>
  <sitemap>
    <loc>http://localhost:8888/urlformatter_jspref/navigation0.xml</loc>
  </sitemap>
  <sitemap>
    <loc>http://localhost:8888/urlformatter_jspref/searchterm0.xml</loc>
  </sitemap>
  <sitemap>
    <loc>http://localhost:8888/urlformatter_jspref/staticpage0.xml</loc>
```

```
  </sitemap>
</sitemapindex>
```

Detail files

The detail files contain links to individual record detail pages. Based on the query specified in the main configuration file, a link to a record details page is generated for each record returned by that query.

The following is a sample detail file that might be generated using the default settings:

```
<?xml version="1.0" encoding="UTF-8" ?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
<url>
   <loc>http://localhost:8888/urlformatter_jspref/controller/Sonoma/Winery-
Lyeth/_/A-34699/An-0/Au-P_Winery</loc>
    <lastmod>2009-03-26</lastmod>
</url>

<url>
   <loc>http://localhost:8888/urlformatter_jspref/controller/Sonoma/Winery-
Benziger/_/A-34700/An-0/Au-P_Winery</loc>
    <lastmod>2009-03-26</lastmod>
</url>
...
</urlset>
```

Related Links

The main configuration file on page 21
The main configuration file for the Sitemap Generator is located in `conf/conf.xml`.

Navigation files

For each record returned by the query specified in the main configuration file, the Sitemap Generator creates navigation links based on the dimension values (i.e. 'Red', or 'Sonoma Valley') associated with that record, as well as dimension settings in the main configuration file.

When a user browses an Endeca application, they reach record pages via navigation pages. For example, if the Endeca record is a Red Wine from the Sonoma Valley, a user might reach this record by selecting 'Red Wines', and then selecting 'Sonoma Valley.' The 'Red Wines' page and the 'Sonoma Valley' page are navigation pages.

The following is a sample navigation file that might be generated using the default settings:

```
<?xml version="1.0" encoding="UTF-8" ?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
<url>
   <loc>http://localhost:8888/urlformatter_jspref/controller/Winery-Ol-Blue-
Jay/_/N-1z13wau/Nu-P_Winery</loc>
    <lastmod>2009-03-26</lastmod>
</url>

<url>
   <loc>http://localhost:8888/urlformatter_jspref/controller/Other-Califor¬
nia/Winery-Ol-Blue-Jay/_/N-1z141wnZ1z13wau/Nu-P_Winery</loc>
    <lastmod>2009-03-26</lastmod>
</url>
```

```
...
</urlset>
```

Related Links

The main configuration file on page 21

The main configuration file for the Sitemap Generator is located in `conf/conf.xml`.

Search term files

The Sitemap Generator creates links leading to search results pages for the most popular search terms as specified in the search terms configuration file.

The base portion of these links can be customized in the template configuration file, and the query string parameters can be customized in the URL formatting configuration file. Typically, you would generate the search terms configuration file based upon a list of the application's most commonly searched terms.

The following is a sample search terms file that might be generated using the default settings:

```
<?xml version="1.0" encoding="UTF-8" ?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
<url>
   <loc>http://localhost:8888/urlformatter_jspref/controller/Wine-Red/_/N-
66t/Ntt-merlot?Ntk=All</loc>
   <lastmod>2009-03-26</lastmod>
</url>
<url>
   <loc>http://localhost:8888/urlformatter_jspref/controller/_/N-/Ntt-
cabernet?Ntk=All</loc>
   <lastmod>2009-03-26</lastmod>
</url>
<url>
   <loc>http://localhost:8888/urlformatter_jspref/controller/_/N-/Ntt-shi¬
raz?Ntk=All</loc>
   <lastmod>2009-03-26</lastmod>
</url>
</urlset>
```

Related Links

The search terms configuration file on page 29

In order for the Sitemap Generator to create links to pages resulting from commonly searched terms, you must supply the common search parameters (including queries or navigations) in the search terms configuration file.

Static pages files

The static pages files contain links to pre-existing static pages, as specified in the static pages configuration file.

The base section of these URLs can be modified using the template configuration file.

The following is a sample static page that might be generated using the default settings:

```
<?xml version="1.0" encoding="UTF-8" ?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
<url>
   <loc>http://localhost:8888/urlformatter_jspref/contactus.html</loc>
   <lastmod>2009-03-26</lastmod>
```

```
</url>
<url>
   <loc>http://localhost:8888/urlformatter_jspref/aboutus.html</loc>
   <lastmod>2009-03-26</lastmod>
</url>

<url>
   <loc>http://localhost:8888/urlformatter_jspref/jobs.html</loc>
   <lastmod>2009-03-26</lastmod>
</url>
...
</urlset>
```

Related Links

The static pages configuration file on page 30

In order for the Sitemap Generator to create links to existing static pages, you must specify custom static URLs in the static pages configuration file.

Chapter 4

Configuring

This section provides a general overview of the Sitemap Generator's files and describes how to customize the tool to an application's needs. Before modifying any of the Configuration Files, Endeca recommends that you first make a backup of the original file.

Configuration files

This section provides an overview of the configuration files used by the Sitemap Generator.

The `conf` directory in the Sitemap Generator installation contains the following files:

File	Description
`conf.xml`	A sample main configuration file, which specifies settings to use when running the Sitemap Generator, such as which template configuration file to use, and where to generate output files.
`xml_tempate.xml`	A sample template configuration file that can be used to create Sitemap protocol XML files.
`html_template.xml`	A sample template configuration file that can be used to create a generic HTML sitemap.
`searchterms.xml`	A sample search terms configuration file. The Sitemap Generator creates a link for each of the search term parameters in the list.
`staticpages.txt`	A sample static pages configuration file. The Sitemap Generator creates a link for each of the pages in the list.
`urlconfig.xml`	A sample URL formatting configuration file, which specifies the settings for all URL formatting done by the Sitemap Generator.

The main configuration file

The main configuration file for the Sitemap Generator is located in `conf/conf.xml`.

This configuration file contains the following elements:

Element	Description
TEMPLATE_FILE	Specifies the template configuration file. The template configuration file customizes the formatting of each of the links generated. The location of this file can either be set using an absolute path, or a path relative to the location of this configuration file.
INDEX_FILE	Specifies the name of the index file that the Sitemap Generator creates. This setting is also used to determine the relative path and extension for other output files to be created. The location of this file can either be set using an absolute path, or a path relative to the location of the configuration file.
MDEX_ENGINES	Specifies the MDEX Engine or Engines to be queried by the Sitemap Generator. Endeca recommends that you use a dedicated staging index for generating sitemaps to minimize impact on production query performance. The <MDEX_ENGINES> tag allows for queries against multiple engines, as in an Agraph deployment.
QUERY_FIELD_LIST	Specifies dimension and property names available for tag replacements in the template configuration files. All dimensions and/or properties used in the urlconfig.xml file should be specified here. If no QUERY_FIELD_LIST is specified, properties and dimensions that are enabled for display with record list are available for tag replacement. However, for improved performance, Endeca recommends using the QUERY_FIELD_LIST configuration.
SEARCH_TERMS_FILE	(Optional) Specifies the search terms configuration file to use for building search term pages and links. The location of this file can be set by using either an absolute or relative path.
STATIC_PAGES_FILE	(Optional) Specifies the static pages configuration file to use for building static page links. The location of this file can be set by using either an absolute or relative path.
URL_FORMAT_FILE	(Optional) Specifies the URL formatting configuration file. The URL formatting configuration file customizes the text inserted each time the FORMATTED_URL tag is encountered in the template configuration file. The location of this file can be set by either using an absolute or relative path.

Element	Description
URLFORMATTER_COMPONENT	Specifies the top level component (or bean) in the URL formatting configuration file, which is used each time a FORMATTED_URL tag is encountered in the template configuration file.
LINKS_PER_FILE	Specifies the maximum number of links to include in a sitemap file before rolling to a new output file. Applies to all output files, including detail files, navigation files, search term files, and static pages files.
MAX_RECS	Specifies the maximum number of records to return for the bulk export query. This is useful for debugging and testing against large indices. ALL_RECS indicates that all records should be returned.
NAVIGATION_PAGE_SPEC_LIST	Specifies combinations of dimensions with which to create navigation page links.

MDEX Engine configuration

Each <MDEX_ENGINES> element in the main configuration file can have one or more <ENGINE> elements.

The <ENGINE> element contains the following children:

Element	Description
HOST	Specifies the host of an MDEX Engine to be queried by the Sitemap Generator.
PORT	Specifies the port of an MDEX Engine to be queried by the Sitemap Generator.
ROOT_QUERY	Specifies the Endeca query string for the Sitemap Generator to use when submitting the bulk export query against the specified index. Typically, this is left as a root query (N=0) to retrieve all records from an index. However, this element can also be modified to specify a subset of records to retrieve from the index. All queries should be entered in unencoded format (e.g. N=8021).
ROLLUP_KEY	(Optional) Specifies to the MDEX Engine the "column name" on which to aggregate record results. Results returned from the Engine are grouped according to this key. It is also possible to omit this parameter and specify the rollup key within the ROOT_QUERY value.

The following example shows the configuration of an MDEX Engine with a rollup key:

```
<MDEX_ENGINES>
  <ENGINE>
      <HOST>localhost</HOST>
      <PORT>15000</PORT>
      <ROOT_QUERY><![CDATA[N=0]]></ROOT_QUERY>
      <ROLLUP_KEY>P_Winery</ROLLUP_KEY>
  </ENGINE>
</MDEX_ENGINES>
```

The query field list

In order to include dimension and property names in tag replacements, you need to include them in the QUERY_FIELD_LIST in the Sitemap Generator's main configuration file.

If no QUERY_FIELD_LIST is specified, properties and dimensions that are enabled for display with record list are available for tag replacement. However, for improved performance, Endeca recommends using the QUERY_FIELD_LIST configuration.

The QUERY_FIELD_LIST contains a list of query fields. You add dimensions and properties as values on the QUERY_FIELD element. For example, the following configuration makes the P_Name property and Wine Type dimension information available for tag replacement in the template configuration files:

```
<QUERY_FIELD_LIST>
   <QUERY_FIELD>P_Name</QUERY_FIELD>
   <QUERY_FIELD>Wine Type</QUERY_FIELD>
</QUERY_FIELD_LIST>
```

Any properties or dimensions that are not included in the QUERY_FIELD_LIST are not available and cannot be used for tag replacement.

Note: The Sitemap Generator only accepts one QUERY_FIELD_LIST. If you create more than one, only the first QUERY_FIELD_LIST in the `conf.xml` file is read.

Related Links

About tag replacement on page 27
When outputting the content of each element, the Sitemap Generator replaces any **-enclosed text in the template configuration file with dynamic values.

The navigation page spec list

Navigation links are created by examining the dimension values tagged to each record processed.

For example, assume the following records are tagged with the following values from dimensions A, B, and C.

```
Rec1  A0    B1    C2
Rec2  A1    B2    C3
Rec3  A2    B2    C3
```

If a navigation page spec specified in the NAVIGATION_PAGE_SPEC_LIST element includes dimensions B and C, the following navigation links will be generated by iterating over all three records and creating a hash of all unique combinations of values from dimensions B and C.

```
B1+C2
B2+C3
```

Each dimension element in the NAVIGATION_PAGE_SPEC_LIST has the required attribute FULL_HIERARCHY. This attribute designates whether intermediate navigation page links should be generated for hierarchical dimensions. For instance, here is an example of hierarchical dimension values:

```
DimensionA
 DimensionVal1
  DimensionVal2
```

If the FULL_HIERARCHY attribute is set to "False", then values will be generated only for DimensionVal2. However, if the FULL_HIERARCHY attribute is set to "True", then a link for the intermediate value DimensionVal1 would also be generated.

For example, the following configuration:

```
<NAVIGATION_PAGE_SPEC_LIST>
 <NAVIGATION_PAGE_SPEC>
  <DIMENSION_NAME FULL_HIERARCHY=”True”>B</DIMENSION_NAME>
  <DIMENSION_NAME FULL_HIERARCHY=”False”>C</DIMENSION_NAME>
 </NAVIGATION_PAGE_SPEC>
 <NAVIGATION_PAGE_SPEC>
  <DIMENSION_NAME FULL_HIERARCHY=”False”>A</DIMENSION_NAME>
</NAVIGATION_PAGE_SPEC>
</NAVIGATION_PAGE_SPEC_LIST>
```

generates the following navigation pages when processing the sample record set at the top of this section:

```
A0
A1
A2
B1+C2
B2+C3
Any ancestor of B1+C2
Any ancestor of B2+C3
```

Important: The Sitemap Generator can only evaluate dimensions enabled for display (in Developer Studio) in the results list of an Endeca query. Dimensions that are not enabled for display in results lists cannot be used to create navigation page links.

Finally, it is important to realize that the creation of navigation page links behaves in a very combinatoric fashion. Endeca strongly recommends that only two or fewer dimensions be specified for any given <NAVIGATION_PAGE_SPEC>. Otherwise, millions of navigation links could easily be generated.

The template configuration file

The template configuration file is an XML file that defines the format of pages and links created by the Sitemap Generator.

A template must be specified when running the Sitemap Generator. This is done by setting the <TEMPLATE_FILE> tag in the main configuration file. Two sample templates are provided in the `conf` directory:

File name	Description
`xml_template.xml`	This file is used to create Sitemap protocol XML pages. See *http://www.sitemaps.org/* for more details.  **Note:** Google, Yahoo, Microsoft, and Ask.com have agreed to support the same Sitemap protocol. The `xml_template.xml` file follows that protocol.
`html_template.xml`	This file is used to create simple HTML sitemap pages. These pages include the appropriate "robots" meta tags that should allow most spiders to successfully crawl these pages.

The table below contains the standard elements of the template configuration file:

XML Element	Usage
`INDEX_LINK`	Specifies the link format in the index file.
`DETAIL_LINK`	Specifies the link format used in record detail pages.
`NAVIGATION_LINK`	Specifies the link format used in navigation pages.
`SEARCH_TERM_LINK`	Specifies the link format used in search term pages.
`STATIC_PAGE_LINK`	Specifies the link format used in static pages.
`INDEX_HEADER`	Specifies the header used in the index file.
`INDEX_FOOTER`	Specifies the footer used in the index file.
`PAGE_HEADER`	Specifies the header used in the detail, navigation, search terms, and static pages files.
`PAGE_FOOTER`	Specifies the footer that is used in the detail, navigation, search terms, and static pages files.

The Sitemap Generator uses the contents of the above elements to create pages. For example, to create a detail page, the Sitemap Generator uses the `PAGE_HEADER`, `DETAIL_LINK`, and `PAGE_FOOTER` XML elements.

About tag replacement

When outputting the content of each element, the Sitemap Generator replaces any **-enclosed text in the template configuration file with dynamic values.

For example, if the DETAIL_LINK is:

```
<DETAIL_LINK><![CDATA[
<url>
   <loc>http://localhost:8888/urlformatter_jspref/detail?ID=**RECID**</loc>

   <lastmod>**TIMESTAMP**</lastmod>
</url>
]]></DETAIL_LINK>
```

And if one of the records resulting from the query (set in the main configuration file) has ID 53, then one of the links in the detail page would be:

```
<url>
   <loc>http://localhost:8888/urlformatter_jspref/detail?ID=53</loc>
   <lastmod>2007-10-17</lastmod>
</url>
```

Related Links

The query field list on page 24
In order to include dimension and property names in tag replacements, you need to include them in the QUERY_FIELD_LIST in the Sitemap Generator's main configuration file.

Replacement tags for INDEX_LINK

The INDEX_LINK section of the template configuration file has a set of valid replacement tags.

The following table contains all the replacement parameters that can be used in the index link:

Parameter	Description	Example
FILE_NAME	Replaced with each of the file names created by the Sitemap Generator.	http://localhost:8888/url¬ format¬ ter_jspref/**FILE_NAME**

Replacement tags for DETAIL_LINK

The DETAIL_LINK section of the template configuration file has a set of valid replacement tags.

The following table contains all the replacement parameters that can be used in the detail link:

Parameter	Description	Example
FORMATTED_URL (Preferred)	Replaced with record-related information in search-engine optimized format, as specified in the URL formatting configuration file. This setting is applicable for both aggregate and non-aggregate queries.	http://localhost:8888/url¬ formatter_jspref/con¬ troller**FORMATTED_URL**

Parameter	Description	Example
RECID	Replaced with the record ID of each Endeca Record.	http://localhost:8888/url¬ formatter_jspref/con¬ troller?R=**RECID**
[Any Record Property]	Replaced with the value of the Property for each Record.  **Note:** The Sitemap Generator can use only properties enabled for display in the results list of an Endeca query. Properties that are enabled for use only on record detail requests are not be displayed.	localhost:8888/urlformat¬ ter_jspref/con¬ troller?R=**P_WineID**
ROLLUP_KEY	Replaced with rollup key of the aggregated record query.	localhost:8888/urlformat¬ ter_jspref/controller/ A=**RE¬ CID**&Au=**ROLLUP_KEY|Ur¬ lEncode**

Replacement tags for NAVIGATION_LINK

The NAVIGATION_LINK section of the template configuration file has a set of valid replacement tags.

The following table contains all the replacement parameters that can be used in the NAVIGATION_LINK:

Parameter	Description	Example
FORMAT¬ TED_URL (Preferred)	Replaced with the navigation information in search-engine optimized format, as specified in the URL formatting configuration file. This setting is applicable for both aggregate and non-aggregate queries.	http://localhost:8888/urlformat¬ ter_jspref/controller**FORMAT¬ TED_URL**
DIMVAL_IDS	Replaced with corresponding dimension value IDs of each navigation page.	http://localhost:8888/urlformat¬ ter_jspref/controller?N=**DIM¬ VAL_IDS**
DIM¬ VAL_NAMES	Replaced with dimension values of each navigation page. These are useful when tailoring sitemaps for HTML crawlers (using the HTML template, for example).	**DIMVAL_NAMES**
ROLLUP_KEY	Replaced with rollup key of the aggregated record query.	http://localhost:8888/urlformat¬ ter_jspref/controller?N= **DIM¬ VAL_IDS**&Nu=**ROLLUP_KEY **

Replacement tags for SEARCH_LINK

The SEARCH_LINK section of the template configuration file has a set of valid replacement tags.

The following table contains all the replacement parameters that can be used in the search link:

Parameter	Description	Example
FORMAT¬ TED_URL	Replaced with search parameters from the search terms configuration file, in Endeca URL format.	http://localhost:8888/urlformat¬ ter_jspref/controller**FORMAT¬ TED_URL**

Replacement tags for STATIC_PAGE_LINK

The STATIC_PAGE_LINK section of the template configuration file has a set of valid replacement tags.

The following table contains all the replacement parameters that can be used in the static page link:

Parameter	Description	Example
STAT¬ IC_PAGE	The name of the static page. Obtained from each word in the static pages configuration file.	http://localhost:8888/urlformat¬ ter_jspref/**STATIC_PAGE**

The search terms configuration file

In order for the Sitemap Generator to create links to pages resulting from commonly searched terms, you must supply the common search parameters (including queries or navigations) in the search terms configuration file.

The location of the search terms configuration file is specified in the main configuration file. At the top of the file, specify a default host, port, and query using <MDEXHOST>, <MDEXPORT>, and <DEFAULT¬
QUERY> tags. Below the default tags, each <URL> tag describes a unique search result link. Each tag contains a set of Endeca URL parameters in <PARAM> tags. An optional special parameter named <QUERY> overrides the default query for that search result link.

Here is an example of a search terms configuration file:

```
<xml version=”1.0” encoding=”UTF-8”>
<!--searchterms.xml
Configuration of terms from which URLs can be generated -->

<URLS>
  <MDEXHOST>localhost</MDEXHOST>
  <MDEXPORT>15000</MDEXPORT>
  <DEFAULTQUERY><![CDATA[N=0]]></DEFAULTQUERY>
  <URL>
   <PARAM NAME="QUERY"><![CDATA[N=8021]]></PARAM>
   <PARAM NAME="Ntt">merlot</PARAM>
 <PARAM NAME="Ntk">All</PARAM>
  </URL>
    <URL>
     <PARAM NAME="Ntt">cabernet</PARAM>
```

```
   <PARAM NAME="Ntk">All</PARAM>
</URLS>
```

The Sitemap Generator creates a link based on the parameters within each <URL> tag. The formatting of the base portion of the link (e.g. "localhost:8888/urlformatter_jspref") is configured in the template configuration file. The formatting of the parameters is configured in the URL formatting configuration file.

For example, if the SEARCH_TERM_LINK were defined as localhost:8888/urlformatter_jspref in the template configuration file, the Sitemap Generator might generate the following link from the first URL in the above example: http://localhost:8888/urlformatter_jspref/con¬troller/Wine-Red/_/N-66t/Ntt-merlot?Ntk=All.

Note that the un-encoded N values should be specified for <QUERY> values (e.g. N=8021).

Related Links

The URL formatting configuration file on page 31
The URL formatting configuration file controls the format of the URL parameters that are substituted for the **FORMATTED_URL** tag specified in the template configuration file.

The template configuration file on page 26
The template configuration file is an XML file that defines the format of pages and links created by the Sitemap Generator.

The static pages configuration file

In order for the Sitemap Generator to create links to existing static pages, you must specify custom static URLs in the static pages configuration file.

The location of the static pages configuration file is specified in the main configuration file. The static pages configuration file is carriage-return delimited and plain text, where each line designates a separate page. You can specify either absolute paths or relative paths. For example:

```
AboutUs.HTML
ContactUs.HTML
Index.HTM
Products/
```

The Sitemap Generator creates a link for each URL using the formatting specified in the template configuration file.

For example, if the STATIC_PAGE_LINK were defined as localhost:8888/urlformat¬ter_jspref/ in the template configuration file, the Sitemap Generator would generate the following link from the first URL in the above example: localhost:8888/urlformatter_jspref/Abou¬tUs.html.

Related Links

The template configuration file on page 26
The template configuration file is an XML file that defines the format of pages and links created by the Sitemap Generator.

The URL formatting configuration file

The URL formatting configuration file controls the format of the URL parameters that are substituted for the `**FORMATTED_URL**` tag specified in the template configuration file.

The Sitemap Generator uses the settings specified in the URL formatting configuration file in conjunction with the Endeca URL Optimization API to produce search-engine optimized URLs. The Sitemap Generator is essentially reproducing the process by which your URL-optimized application creates URLs for any given link.

Important: To ensure that the URLs in the sitemap are consistent with the URLs in your application, you must integrate the URL Optimization API with your application and configure it with the same formatting options that are specified in the Sitemap Generator's URL formatting configuration file. For more information about the capabilities of the URL Optimization API, its configuration, and integration with the Sitemap Generator, please refer to the appropriate *URL Optimization API Developer's Guide*.

The URL formatting configuration file uses Spring Framework syntax. If you need further information about this format, please consult the documentation provided with the Spring Framework. The Sitemap Generator includes a sample configuration file (`conf/urlconfig.xml`), which is tailored for the standard wine data set.

Chapter 5

Troubleshooting and Performance

This section provides information about performance and troubleshooting common problems encountered while running and configuring the Sitemap Generator.

Commonly encountered errors

This section addresses the most commonly ecountered errors that users run into while running the Sitemap Generator.

Error message or problem	Description and suggested solution
`Unable to Locate Config File Speci¬ fied`	This error is generated by the `.sh` or `.bat` scripts if they cannot locate the main configuration file specified as an input parameter. Make sure to specify either an absolute path, or a path relative to the directory from which you are running the script.
`ENDECA_ROOT Is Not Set`	This error is generated by the `.sh` or `.bat` scripts if the `ENDECA_ROOT` environment variable is not set. These scripts use `ENDECA_ROOT` to locate `endeca_navigation.jar`, and `servlet-api.jar`, which need to be included in the classpath when running the Sitemap Generator.
`Fatal Error in Running Queries: Error Establishing Connection to Retrieve Navigation Engine`	This generally means you are trying to connect to an MDEX Engine that is not running. Check the main configuration file (`conf.xml`) and make sure that the host and port of the MDEX Engine are correct.
`Unable to Locate Sitemap Generator Library`	This error is generated by the `.sh` or `.bat` scripts if they are unable to locate the `/lib/siteMapGenerator.jar` file included in the Sitemap Generator installation. The location

Error message or problem	Description and suggested solution
	of this file is computed from the location of the `.sh` or `.bat` scripts being run, with the assumption that those scripts have not been moved from their default locations. This file also needs to be included in the classpath when running the Sitemap Generator.
`Unable to Locate Template Configura¬tion File`	The Sitemap Generator throws this exception if it is unable to find the template configuration file specified in the main configuration file. Remember to use either an absolute path, or a path relative to the location of the main configuration file, when specifying the template configuration file to use.
Can't Find Output Files	Remember to use either an absolute path or a path relative to the location of the main configuration file, when specifying the target location for output files. If you are still unable to find the output files, the last line of the standard output should indicate the location where the files were written: `INFO: Sitemap files output here: C:\Endeca\SEM\SitemapGenera¬tor\2.0.0\sitemap\*.xml`
Sitemap Files Missing Values / Name of the Tag Output	If sitemap files that are output are missing values that were specified in the template configuration files, or contain the Name of the tag used in the template configuration files, such as: <a `href="http://localhost:8888/urlformat¬ter_jspref/detail?id=">...</a>` Or: <a `href="http://localhost:8888/urlformat¬ter_jspref/detail?id=id">...</a>`, those keys did not have corresponding values for that given record. In the above example, there is no property "id" specified in the `QUERY_FIELD_LIST`. Make sure such a property exists, and that it is included in the `QUERY_FIELD_LIST` in the main configuration file.
No Navigation Pages Created	If no navigation pages were created, then the dimensions listed in the `NAVIGA¬TION_PAGE_SPEC_LIST` parameter in the main configuration file may not be included in the `QUERY_FIELD_LIST`. Make sure to add the dimensions to the `QUERY_FIELD_LIST` in the main configuration file. Another possibility is that no records exist with the combination of dimensions that you have specified.

Error message or problem	Description and suggested solution
Errors Reading Search Term Queries	If errors occurred running search term queries, the problem may be that the search terms configuration file is not configured appropriately. Please check your: • `MDEXHOST` • `MDEXPORT` • `DEFAULTQUERY` as well as the parameters for each query that failed.

Related Links

The main configuration file on page 21
The main configuration file for the Sitemap Generator is located in `conf/conf.xml`.

The template configuration file on page 26
The template configuration file is an XML file that defines the format of pages and links created by the Sitemap Generator.

The search terms configuration file on page 29
In order for the Sitemap Generator to create links to pages resulting from commonly searched terms, you must supply the common search parameters (including queries or navigations) in the search terms configuration file.

Performance information

Depending on your application, using the Sitemap Generator may have performance implications. This section addresses those possible implications and provides options for resolving them.

Runtime performance

When run locally on the same server as the MDEX Engine being queried, the Sitemap Generator is able to create more than 50,000 detail and navigation links in less than 30 seconds. When run remotely, performance varies based on network congestion.

Production vs. dedicated MDEX Engine

Since the bulk-export query can be expensive, it is advisable for sites with high throughput requirements to either schedule the Sitemap Generator for off-peak hours, or even run the Sitemap Generator against a dedicated staging index. When running the Sitemap Generator against a dedicated index, you can gain an additional performance advantage by using the `QUERY_FIELD_LIST` to specify the properties and dimensions that are available for display in URLs rather than enabling the properties and dimensions to display with record list.

Combinatoric navigation pages

Due to the method in which navigation links are defined, it is important to avoid creating complex navigation page specs that would result in an overwhelming number of unique links. Such a scenario may cause excessive memory usage by the Sitemap Generator, as well as long runtimes to write all of the specified links. Endeca strongly recommends that only two or fewer dimensions be specified in

the main configuration file for any given <NAVIGATION_PAGE_SPEC>. Otherwise, millions of navigation links could easily be generated.

Related Links

The navigation page spec list on page 24

Navigation links are created by examining the dimension values tagged to each record processed.

Chapter 6

Implementing

This section describes the steps necessary to move your Sitemap Generator files into production and update search engines.

About validating sitemaps

Before deploying your sitemaps, you should ensure that all the generated links are valid.

Endeca recommends that you write a script to test each URL in the sitemap against your application server and confirm that each link returns an HTTP 200 (OK) code. Broken links may indicate a mismatch between the URL formatting configuration of the Sitemap Generator and the configuration of the URL Optimization API that the application uses to generate its links.

If you are using the URL Optimization API for the RAD Toolkit for ASP.NET, be aware that the .NET Framework imposes a limit of 260 characters on the physical path in URLs. Any attempt to access a link that violates this limit returns an HTTP 400 (Bad Request) code. Your application code should detect and shorten such URLs, and your sitemap validation script should also detect and shorten these URLs accordingly.

Related Links

The URL formatting configuration file on page 31

The URL formatting configuration file controls the format of the URL parameters that are substituted for the `**FORMATTED_URL**` tag specified in the template configuration file.

About moving sitemap files to production

Once you have created and validated the sitemap files, you should move them to the production Web servers. You can either do this manually, or via operations scripts.

It is also important to note that when the Sitemap Generator runs, it deletes any previously existing files from the target output directory. Therefore, if you want to do any sitemap archiving, you must handle this separately.

Notifying search engines of a sitemap location

Endeca recommends that you actively register your sitemap files with various Web search engines.

For more information on sitemap submission programs for each search engine please visit:

- Google: *http://www.google.com/webmasters/sitemaps/*. For more information on Google's Sitemaps Program, please visit *http://www.google.com/support/webmasters/bin/topic.py?topic=8465*
- Yahoo! Search: *https://siteexplorer.search.yahoo.com/submit*
- Ask.com: *http://about.ask.com/en/docs/about/webmasters.shtml#22*
- Bing (MSN): *http://www.bing.com/webmaster*

Another method of notifying the search engines of the sitemap location is through the `robots.txt` file. To do so you need to add a line to your site's `robots.txt` file similar to the following example:

```
Sitemap: http://localhost:8888/urlformatter_jspref/sitemap.xml
```

Here the `localhost:8888/urlformatter_jspref` portion of the URL should be replaced with the absolute URL for your sitemap index file. For more information on this notification method please visit *http://www.sitemaps.org/protocol.php#informing*.

Notifying search engines of an updated sitemap

You can ping search engines to let them know that your sitemap has been updated. For more information and instructions for each search engine, please visit:

- Google: *http://www.google.com/support/webmasters/bin/answer.py?answer=34609*.
- Yahoo! Search: *http://developer.yahoo.com/search/siteexplorer/V1/ping.html*
- Ask.com: *http://about.ask.com/en/docs/about/webmasters.shtml#22*
- Bing (MSN): *http://www.bing.com/webmaster*