

Google Search Appliance

Administering Crawl

Google Search Appliance software version 7.4



Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
www.google.com

GSA-ADM_200.02
March 2015

© Copyright 2015 Google, Inc. All rights reserved.

Google and the Google logo are, registered trademarks or service marks of Google, Inc. All other trademarks are the property of their respective owners.

Use of any Google solution is governed by the license agreement included in your original contract. Any intellectual property rights relating to the Google services are and shall remain the exclusive property of Google, Inc. and/or its subsidiaries ("Google"). You may not attempt to decipher, decompile, or develop source code for any Google product or service offering, or knowingly allow others to do so.

Google documentation may not be sold, resold, licensed or sublicensed and may not be transferred without the prior written consent of Google. Your right to copy this manual is limited by copyright law. Making copies, adaptations, or compilation works, without prior written authorization of Google, is prohibited by law and constitutes a punishable violation of the law. No part of this manual may be reproduced in whole or in part without the express written consent of Google. Copyright © by Google, Inc.

Contents

Chapter 1	Introduction	7
	Deprecation Notices	7
	On-Board File System Crawling	7
	On-Board Database Crawler	7
	What Is Search Appliance Crawling?	7
	Crawl Modes	8
	What Content Can Be Crawled?	9
	Public Web Content	9
	Secure Web Content	9
	Content from Network File Shares	10
	Databases	10
	Compressed Files	10
	What Content Is Not Crawled?	10
	Content Prohibited by Crawl Patterns	11
	Content Prohibited by a robots.txt File	11
	Content Excluded by the nofollow Robots META Tag	11
	Links within the area Tag	12
	Unlinked URLs	12
	Configuring the Crawl Path and Preparing the Content	12
	How Does the Search Appliance Crawl?	12
	About the Diagrams in this Section	13
	Crawl Overview	13
	Starting the Crawl and Populating the Crawl Queue	14
	Attempting to Fetch a URL and Indexing the Document	16
	Following Links within the Document	19
	When Does Crawling End?	21
	When Is New Content Available in Search Results?	21
	How Are URLs Scheduled for Recrawl?	21
	How Are Network Connectivity Issues Handled?	22
	What Is the Search Appliance License Limit?	22
	Google Search Appliance License Limit	22
	When Is a Document Counted as Part of the License Limit?	23
	License Expiration and Grace Period	24
	How Many URLs Can Be Crawled?	24
	How Are Document Dates Handled?	24
	Are Documents Removed From the Index?	25
	Document Removal Process	26
	What Happens When Documents Are Removed from Content Servers?	26

Chapter 2	Preparing for a Crawl	28
	Preparing Data for a Crawl	28
	Using robots.txt to Control Access to a Content Server	28
	Using Robots meta Tags to Control Access to a Web Page	30
	Using X-Robots-Tag to Control Access to Non-HTML Documents	31
	Excluding Unwanted Text from the Index	31
	Using no-crawl Directories to Control Access to Files and Subdirectories	33
	Preparing Shared Folders in File Systems	33
	Ensuring that Unlinked URLs Are Crawled	33
	Configuring a Crawl	34
	Start URLs	34
	Follow Patterns	35
	Do Not Follow Patterns	36
	Crawling and Indexing Compressed Files	37
	Testing Your URL Patterns	37
	Using Google Regular Expressions as Crawl Patterns	37
	Configuring Database Crawl	38
	About SMB URLs	38
	Unsupported SMB URLs	39
	SMB URLs for Non-file Objects	40
	Hostname Resolution	40
	Setting Up the Crawler's Access to Secure Content	40
	Configuring Searchable Dates	40
	Defining Document Date Rules	41
Chapter 3	Running a Crawl	42
	Selecting a Crawl Mode	42
	Scheduling a Crawl	42
	Stopping, Pausing, or Resuming a Crawl	43
	Submitting a URL to Be Recrawled	43
	Starting a Database Crawl	44
Chapter 4	Monitoring and Troubleshooting Crawls	45
	Using the Admin Console to Monitor a Crawl	45
	Crawl Status Messages	47
	Network Connectivity Test of Start URLs Failed	47
	Slow Crawl Rate	48
	Non-HTML Content	48
	Complex Content	48
	Host Load	48
	Network Problems	49
	Slow Web Servers	49
	Query Load	49
	Wait Times	50
	Errors from Web Servers	50
	URL Moved Permanently Redirect (301)	51
	URL Moved Temporarily Redirect (302)	51
	Authentication Required (401) or Document Not Found (404) for SMB File	
	Share Crawls	52
	Cyclic Redirects	53

URL Rewrite Rules	53
BroadVision Web Server	53
Sun Java System Web Server	54
Microsoft Commerce Server	54
Servers that Run Java Servlet Containers	54
Lotus Domino Enterprise Server	54
ColdFusion Application Server	56
Index Pages	56
Chapter 5	
Advanced Topics	57
Identifying the User Agent	57
User Agent Name	57
User Agent Email Address	57
Coverage Tuning	58
Freshness Tuning	58
Changing the Amount of Each Document that Is Indexed	59
Configuring Metadata Indexing	59
Including or Excluding Metadata Names	59
Specifying Multivalued Separators	60
Specifying a Date Format for Metadata Date Fields	60
Crawling over Proxy Servers	60
Preventing Crawling of Duplicate Hosts	61
Enabling Infinite Space Detection	61
Configuring Web Server Host Load Schedules	61
Removing Documents from the Index	62
Using Collections	62
Default Collection	62
Changing URL Patterns in a Collection	62
JavaScript Crawling	63
Logical Redirects by Assignments to window.location	63
Links and Content Added by document.write and document.writeln Functions	63
Links that are Generated by Event Handlers	64
Links that are JavaScript Pseudo-URLs	64
Links with an onclick Return Value	65
Indexing Content Added by document.write/writeln Calls	65
Discovering and Indexing Entities	65
Creating Dictionaries and Composite Entities	66
Setting Up Entity Recognition	66
Use Case: Matching URLs for Dynamic Navigation	66
Use Case: Testing Entity Recognition for Non-HTML Documents	69
Wildcard Indexing	71
Chapter 6	
Database Crawling and Serving	72
Database Crawler Deprecation Notice	72
Introduction	72
Supported Databases	73
Overview of Database Crawling and Serving	73
Synchronizing a Database	74
Processing a Database Feed	75
Serving Database Content	75
Configuring Database Crawling and Serving	76
Providing Database Data Source Information	76
Setting URL Patterns to Enable Database Crawl	81
Starting Database Synchronization	82
Monitoring a Feed	82

	Troubleshooting	82
	Frequently Asked Questions	83
Chapter 7	Constructing URL Patterns	86
	Introduction	86
	Rules for Valid URL Patterns	87
	Comments in URL Patterns	88
	Case Sensitivity	88
	Simple URL Patterns	89
	Matching domains	89
	Matching directories	89
	Matching files	90
	Matching protocols	91
	Matching ports	91
	Using the prefix option	91
	Using the suffix option	92
	Matching specific URLs	92
	Matching specified strings	93
	SMB URL Patterns	93
	Exception Patterns	94
	Google Regular Expressions	94
	Using Backreferences with Do Not Follow Patterns	96
	Controlling the Depth of a Crawl with URL Patterns	96
Chapter 8	Crawl Quick Reference	97
	Crawling and Indexing Features	97
	Crawling and Indexing Administration Tasks	99
	Admin Console Basic Crawl Pages	101
	Index	102

Chapter 1

Introduction

Crawling is the process where the Google Search Appliance discovers enterprise content to index. This chapter provides an overview of how the Google Search Appliance crawls public content.

For information about specific feature limitations, see [Specifications and Usage Limits](#).

Deprecation Notices

On-Board File System Crawling

In GSA release 7.4, on-board file system crawling (File System Gateway) is deprecated. It will be removed in a future release. If you have configured on-board file system crawling for your GSA, install and configure the Google Connector for File Systems 4.0.4 or later instead. For more information, see “Deploying the Connector for File Systems,” available from the [Connector Documentation page](#).

On-Board Database Crawler

In GSA release 7.4, the on-board database crawler is deprecated. It will be removed in a future release. If you have configured on-board database crawling for your GSA, install and configure the Google Connector for Databases 4.0.4 or later instead. For more information, see “Deploying the Connector for Databases,” available from the [Connector Documentation page](#).

What Is Search Appliance Crawling?

Before anyone can use the Google Search Appliance to search your enterprise content, the search appliance must build the search index, which enables search queries to be quickly matched to results. To build the search index, the search appliance must browse, or “crawl” your enterprise content, as illustrated in the following example.

The administration at Missitucky University plans to offer its staff, faculty, and students simple, fast, and secure search across all their content using the Google Search Appliance. To achieve this goal, the search appliance must crawl their content, starting at the Missitucky University Web site’s home page.

Missitucky University has a Web site that provides categories of information such as Admissions, Class Schedules, Events, and News Stories. The Web site's home page lists hyperlinks to other URLs for pages in each of these categories. For example, the News Stories hyperlink on the home page points to a URL for a page that contains hyperlinks to all recent news stories. Similarly, each news story contains hyperlinks that point to other URLs.

The relations among the hyperlinks within the Missitucky University Web site constitute a virtual web, or pathway that connects the URLs to each other. Starting at the home page and following this pathway, the search appliance can crawl from URL to URL, browsing content as it goes.

Crawling Missitucky University's content actually begins with a list of URLs ("start URLs") where the search appliance should start browsing; in this example, the first start URL is the Missitucky University home page.

The search appliance visits the Missitucky University home page, then it:

1. Identifies all the hyperlinks on the page. These hyperlinks are known as "newly-discovered URLs."
2. Adds the hyperlinks to a list of URLs to visit. The list is known as the "crawl queue."
3. Visits the next URL in the crawl queue.

By repeating these steps for each URL in the crawl queue, the search appliance can crawl all of Missitucky University's content. As a result, the search appliance gathers the information that it needs to build the search index, and ultimately, to serve search results to end users.

Because Missitucky University's content changes constantly, the search appliance continuously crawls it to keep the search index and the search results up-to-date.

Crawl Modes

The Google Search Appliance supports two modes of crawling:

- Continuous crawl
- Scheduled crawl

For information about choosing a crawl mode and starting a crawl, see "Selecting a Crawl Mode" on page 42.

Continuous Crawl

In continuous crawl mode, the search appliance is crawling your enterprise content at all times, ensuring that newly added or updated content is added to the index as quickly as possible. After the Google Search Appliance is installed, it defaults to continuous crawl mode and establishes the default collection (see "Default Collection" on page 62).

The search appliance does not recrawl any URLs until all new URLs have been discovered or the license limit has been reached (see "What Is the Search Appliance License Limit?" on page 22). A URL in the index is recrawled even if there are no longer any links to that URL from other pages in the index.

Scheduled Crawl

In scheduled crawl mode, the Google Search Appliance crawls your enterprise content at a scheduled time.

What Content Can Be Crawled?

The Google Search Appliance can crawl and index content that is stored in the following types of sources:

- Public Web servers
- Secure Web servers
- Network file shares
- Databases
- Compressed files

Crawling FTP is not supported on the Google Search Appliance.

Public Web Content

Public Web content is available to all users. The Google Search Appliance can crawl and index both public and secure enterprise content that resides on a variety of Web servers, including these:

- Apache HTTP server
- BroadVision Web server
- Sun Java System Web server
- Microsoft Commerce server
- Lotus Domino Enterprise server
- IBM WebSphere server
- BEA WebLogic server
- Oracle server

Secure Web Content

Secure Web content is protected by authentication mechanisms and is available only to users who are members of certain authorized groups. The Google Search Appliance can crawl and index secure content protected by:

- Basic authentication
- NTLM authentication

The search appliance can crawl and index content protected by forms-based single sign-on systems.

For HTTPS websites, the Google Search Appliance uses a serving certificate as a client certificate when crawling. You can upload a new serving certificate using the Admin Console. Some Web servers do not accept client certificates unless they are signed by trusted Certificate Authorities.

Content from Network File Shares

In GSA release 7.4, on-board file system crawling (File System Gateway) is deprecated. For more information, see [Deprecation Notices](#).

The Google Search Appliance can also crawl several file formats, including Microsoft Word, Excel, and Adobe PDF that reside on network file shares. The crawler can access content over Server Message Block (SMB) protocol (the standard network file share protocol on Microsoft Windows, supported by the SAMBA server software and numerous storage devices).

For a complete list of supported file formats, refer to *Indexable File Formats*.

Databases

In GSA release 7.4, the on-board database crawler is deprecated. For more information, see [Deprecation Notices](#).

The Google Search Appliance can crawl databases directly. To access content in a database, the Google Search Appliance sends SQL (Structured Query Language) queries using JDBC (Java Database Connectivity) adapters provided by each database company.

For information about crawling databases, refer to “Database Crawling and Serving” on page 72.

Compressed Files

The Google Search Appliance supports crawling and indexing compressed files in the following formats: .zip, .tar, .tar.gz, and .tgz.

For more information, refer to “Crawling and Indexing Compressed Files” on page 37.

What Content Is Not Crawled?

The Google Search Appliance does not crawl or index enterprise content that is excluded by these mechanisms:

- Crawl patterns
- robots.txt
- nofollow Robots META tag

Also the Google Search Appliance cannot:

- Follow any links that appear within an HTML area tag.
- Discover unlinked URLs. However, you can enable them for crawling.
- Crawl any content residing in 192.168.255 subnet, because this subnet is used for internal configuration.

The following sections describe all these exclusions.

Content Prohibited by Crawl Patterns

A Google Search Appliance administrator can prohibit the crawler from following and indexing particular URLs. For example, any URL that should not appear in search results or be counted as part of the search appliance license limit should be excluded from crawling. For more information, refer to "Configuring a Crawl" on page 34.

Content Prohibited by a robots.txt File

To prohibit any crawler from accessing all or some of the content on an HTTP or HTTPS site, a content server administrator or webmaster typically adds a robots.txt file to the root directory of the content server or Web site. This file tells the crawlers to ignore all or some files and directories on the server or site. Documents crawled using other protocols, such as SMB, are not affected by the restrictions of robots.txt. For the Google Search Appliance to be able to access the robot.txt file, the file must be public. For examples of robots.txt files, see "Using robots.txt to Control Access to a Content Server" on page 28.

The Google Search Appliance crawler always obeys the rules in robots.txt. You cannot override this feature. Before crawling HTTP or HTTPS URLs on a host, a Google Search Appliance fetches the robots.txt file. For example, before crawling any URLs on <http://www.mycompany.com/> or <https://www.mycompany.com/>, the search appliance fetches <http://www.mycompany.com/robots.txt>.

When the search appliance requests the robots.txt file, the host returns an HTTP response that determines whether or not the search appliance can crawl the site. The following table lists HTTP responses and how the Google Search Appliance crawler responds to them.

HTTP Response	File Returned?	Google Search Appliance Crawler Response
200 OK	Yes	The search appliance crawler obeys exclusions specified by robots.txt when fetching URLs on the site.
404 Not Found	No	The search appliance crawler assumes that there are no exclusions to crawling the site and proceeds to fetch URLs.
Other responses		The search appliance crawler assumes that it is not permitted to crawl the site and does not fetch URLs.

When crawling, the search appliance caches robots.txt files and refetches a robots.txt file if 30 minutes have passed since the previous fetch. If changes to a robots.txt file prohibit access to documents that have already been indexed, those documents are removed from the index. If the search appliance can no longer access robots.txt on a particular site, all the URLs on that site are removed from the index.

For detailed information about HTTP status codes, visit http://en.wikipedia.org/wiki/List_of_HTTP_status_codes.

Content Excluded by the nofollow Robots META Tag

The Google Search Appliance does not crawl a Web page if it has been marked with the nofollow Robots META tag (see "Using Robots meta Tags to Control Access to a Web Page" on page 30).

Links within the area Tag

The Google Search Appliance does not crawl links that are embedded within an area tag. The HTML area tag is used to define a mouse-sensitive region on a page, which can contain a hyperlink. When the user moves the pointer into a region defined by an area tag, the arrow pointer changes to a hand and the URL of the associated hyperlink appears at the bottom of the window.

For example, the following HTML defines an region that contains a link:

```
<map name="n5BDE56.Body.1.4A70">
  <area shape="rect" coords="0,116,311,138" id="TechInfoCenter"
    href="http://www.bbb.com/main/help/ourcampaign/ourcampaign.htm" alt="">
</map>
```

When the search appliance crawler follows newly discovered links in URLs, it does not follow the link (<http://www.bbb.com/main/help/ourcampaign/ourcampaign.htm>) within this area tag.

Unlinked URLs

Because the Google Search Appliance crawler discovers new content by following links within documents, it cannot find a URL that is not linked from another document through this process.

You can enable the search appliance crawler to discover any unlinked URLs in your enterprise content by:

- Adding unlinked URLs to the crawl path.
- Using a jump page (see “Ensuring that Unlinked URLs Are Crawled” on page 33), which is a page that can provide links to pages that are not linked to from any other pages. List unlinked URLs on a jump page and add the URL of the jump page to the crawl path.

Configuring the Crawl Path and Preparing the Content

Before crawling starts, the Google Search Appliance administrator configures the crawl path (see “Configuring a Crawl” on page 34), which includes URLs where crawling should start, as well as URL patterns that the crawler should follow and should not follow. Other information that webmasters, content owners, and search appliance administrators typically prepare before crawling starts includes:







- Robots exclusion protocol (robots.txt) for each content server that it crawls
- Robots META tags embedded in the header of an HTML document
- googleon/googleoff tags embedded in the body of an HTML document
- Jump pages

How Does the Search Appliance Crawl?

This section describes how the Google Search Appliance crawls Web and network file share content as it applies to both scheduled crawl and continuous crawl modes.

About the Diagrams in this Section

This section contains data flow diagrams, used to illustrate how the Google Search Appliance crawls enterprise content. The following table describes the symbols used in these diagrams.

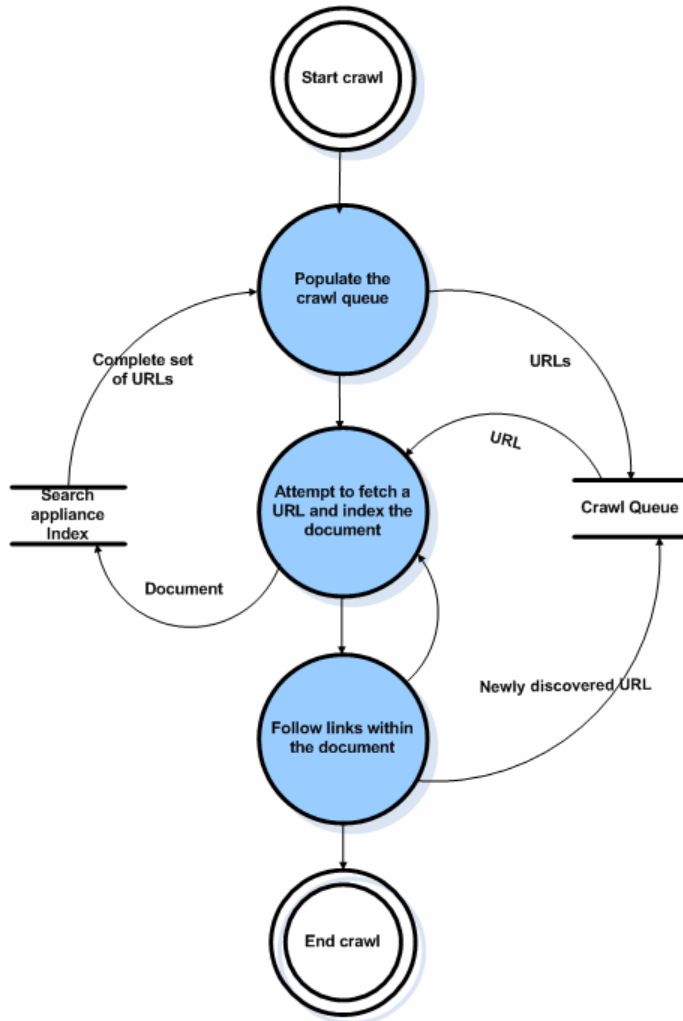
Symbol	Definition	Example
	Start state or Stop state	Start crawl, end crawl
	Process	Follow links within the document
	Data store, which can be a database, file system, or any other type of data store	Crawl queue
	Data flow among processes, data stores, and external interactors	URLs
	External input or terminator, which can be a process in another diagram	Delete URL
	Callout to a diagram element	

Crawl Overview

The following diagram provides an overview of the following major crawling processes:

- Starting the crawl and populating the crawl queue
- Attempting to fetch a URL and index the document
- Following links within the document

The sections following the diagram provide details about each of the these major processes.



Starting the Crawl and Populating the Crawl Queue

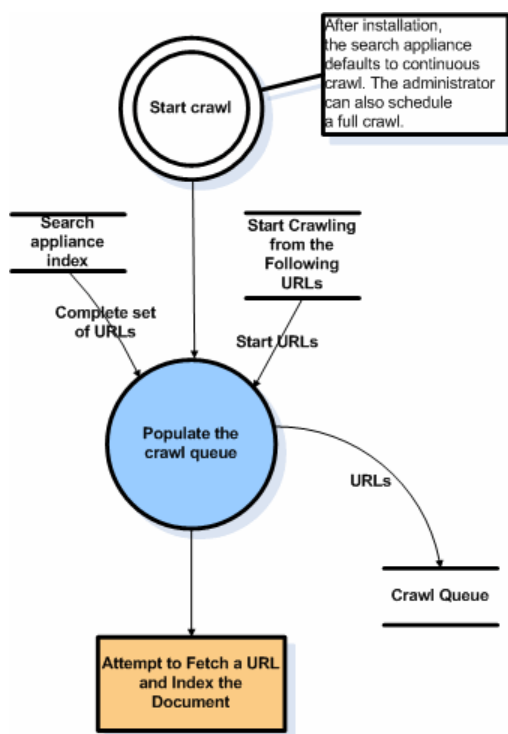
The crawl queue is a list of URLs that the Google Search Appliance will crawl. The search appliance associates each URL in the crawl queue with a priority, typically based on estimated Enterprise PageRank. Enterprise PageRank is a measure of the relative importance of a Web page within the set of your enterprise content. It is calculated using a link-analysis algorithm similar to the one used to calculate PageRank on google.com.

The order in which the Google Search Appliance crawls URLs is determined by the crawl queue. The following table gives an overview of the priorities assigned to URLs in the crawl queue.

Source of URL	Basis for Priority
Start URLs (highest)	Fixed priority
New URLs that have never been crawled	Estimated Enterprise PageRank
Newly discovered URLs	For a new crawl, estimated Enterprise PageRank For a recrawl, estimated Enterprise PageRank and a factor that ensures that new documents are crawled before previously indexed content
URLs that are already in the index (lowest)	Enterprise PageRank, the last time it was crawled, and estimated change frequency

By crawling URLs in this priority, the search appliance ensures that the freshest, most relevant enterprise content appears in the index.

After configuring the crawl path and preparing content for crawling, the search appliance administrator starts a continuous or scheduled crawl (see “Selecting a Crawl Mode” on page 42). The following diagram provides an overview of starting the crawl and populating the crawl queue.

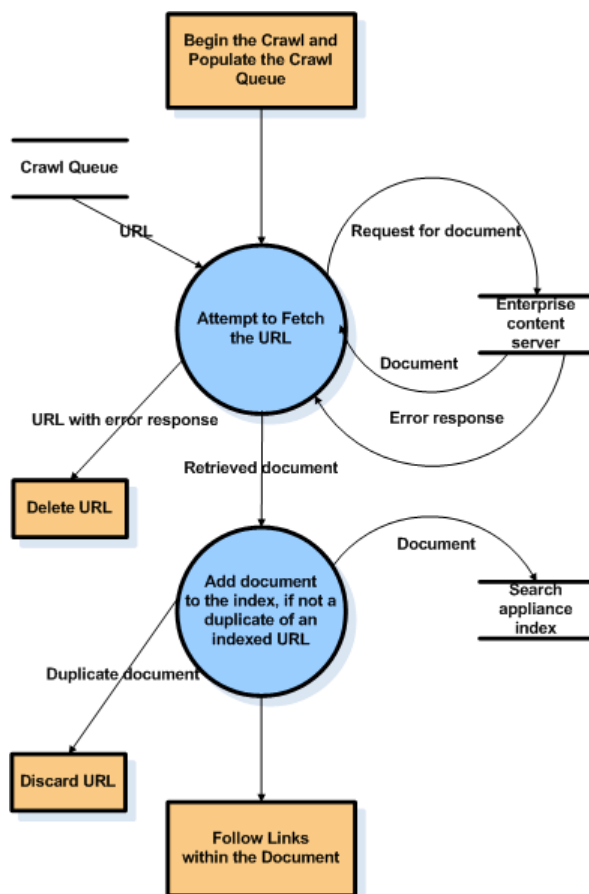


When crawling begins, the search appliance populates the crawl queue with URLs. The following table lists the contents of the crawl queue for a new crawl and a recrawl.

Type of Crawl	Crawl Queue Contents
New crawl	The start URLs that the search appliance administrator has configured.
Recrawl	The start URLs that the search appliance administrator has configured and the complete set of URLs contained in the current index.

Attempting to Fetch a URL and Indexing the Document

The Google Search Appliance crawler attempts to fetch the URL with the highest priority in the crawl queue. The following diagram provides an overview of this process.



If the search appliance successfully fetches a URL, it downloads the document. If you have enabled and configured infinite space detection, the search appliance uses the checksum to test if there are already 20 documents with the same checksum in the index (20 is the default value, but you can change it when you configure infinite space detection). If there are 20 documents with the same checksum in the index, the document is considered a duplicate and discarded (in Index Diagnostics, the document is shown as “Considered Duplicate”). If there are fewer than 20 documents with the same checksum in the index, the search appliance caches the document for indexing. For more information, refer to “Enabling Infinite Space Detection” on page 61.

Generally, if the search appliance fails to fetch a URL, it deletes the URL from the crawl queue. Depending on several factors, the search appliance may take further action when it fails to fetch a URL.

When fetching documents from a slow server, the search appliance paces the process so that it does not cause server problems. The search appliance administrator can also adjust the number of concurrent connections to a server by configuring the web server host load schedule (see “Configuring Web Server Host Load Schedules” on page 61).

Determining Document Changes with If-Modified-Since Headers and the Content Checksum

During the recrawl of an indexed document, the Google Search Appliance sends the If-Modified-Since header based on the last crawl date of the document. If the web server returns a 304 Not Modified response, the appliance does not further process the document. If the web server returns content, the Google Search Appliance uses the Last-Modified header, if present, to detect change. If the Last-Modified header is not present, the search appliance computes the checksum of the newly downloaded content and compares it to the checksum of the previous content. If the checksum is the same, then the appliance does not further process the document.

To detect changes to cached documents when recrawling it, the search appliance:

1. Downloads the document.
2. Computes a checksum of the file.
3. Compares the checksum to the checksum that was stored in the index the last time the document was indexed.
4. If the checksum has not changed, the search appliance stops processing the document and retains the cached document.

If the checksum has changed since the last modification time, the search appliance determines the size of the file (see “File Type and Size” on page 18), modifies the file as necessary, follows newly discovered links within the document (see “Following Links within the Document” on page 19), and indexes the document.

Fetching URLs from File Shares

In GSA release 7.4, on-board file system crawling (File System Gateway) is deprecated. For more information, see [Deprecation Notices](#).

When the Google Search Appliance fetches a URL from a file share, the object that it actually retrieves and the method of processing it depends on the type of object that is requested. For each type of object requested, the following table provides an overview of the process that the search appliance follows. For information on how these objects are counted as part of the search appliance license limit, refer to “When Is a Document Counted as Part of the License Limit?” on page 23.

Requested Object	Google Search Appliance Process Overview
Document	<ol style="list-style-type: none"> 1. Retrieve the document. 2. Detect document changes. 3. Index the document.
Directory	<ol style="list-style-type: none"> 1. Retrieve a list of files and subdirectories contained within the directory. 2. Create a directory listings page. This page contains links to files and subdirectories within the directory. 3. Index the directory listings page.
Share	<ol style="list-style-type: none"> 1. Retrieve a list of files and directories in the top-level directory of the share. 2. Create a directory listings page. 3. Index the directory listings page.
Host	<ol style="list-style-type: none"> 1. Retrieve a list of shares on the host. 2. Create a share listings page. This page is similar to a directory listings page, but with links to the shares on the host instead of files and subdirectories. 3. Index the share listing page. Because of limitations of the share listing process, a share name is not returned if it uses non-ASCII characters or exceeds 12 characters in length. To work around this limitation, you can specify the share itself in Start URLs on the Content Sources > Web Crawl > Start and Block URLs page in the Admin Console.

File Type and Size

When the Google Search Appliance fetches a document, it determines the type and size of the file. The search appliance attempts to determine the type of the file by first examining the Content-Type header. Provided that the Content-Type header is present at crawl time, the search appliance crawls and indexes files where the content type does not match the file extension. For example, an HTML file saved with a PDF extension is correctly crawled and indexed as an HTML file.

If the search appliance cannot determine the content type from the Content-Type header, it examines the file extension by parsing the URL.

As a search appliance administrator, you can change the maximum file size for the downloader to use when crawling documents. By default, the maximum file sizes are:

- 20MB for text or HTML documents
- 100MB for all other document types

To change the maximum file size, enter new values on the **Content Sources > Web Crawl > Host Load Schedule** page. For more information about setting the maximum file size to download, click **Admin Console Help > Content Sources > Web Crawl > Host Load Schedule**.

If the document is:

- A text or HTML document that is larger than the maximum file size, the search appliance truncates the file and discards the remainder of the file
- Any other type of document that does not exceed the maximum file size, the search appliance converts the document to HTML
- Any other type of document that is larger than the maximum file size, the search appliance discards it completely

By default, the search appliance indexes up to 2.5MB of each text or HTML document, including documents that have been truncated or converted to HTML. You can change the default by entering an new amount of up to 10MB. For more information, refer to “Changing the Amount of Each Document that Is Indexed” on page 59.

Compressed document types, such as Microsoft Office 2007, might not be converted properly if the uncompressed file size is greater than the maximum file size. In these cases, you see a conversion error message on the **Index > Diagnostics > Index Diagnostics** page.

LINK Tags in HTML Headers

The search appliance indexes `LINK` tags in HTML headers. However, it strips these headers from cached HTML pages to avoid cross-site scripting (XSS) attacks.

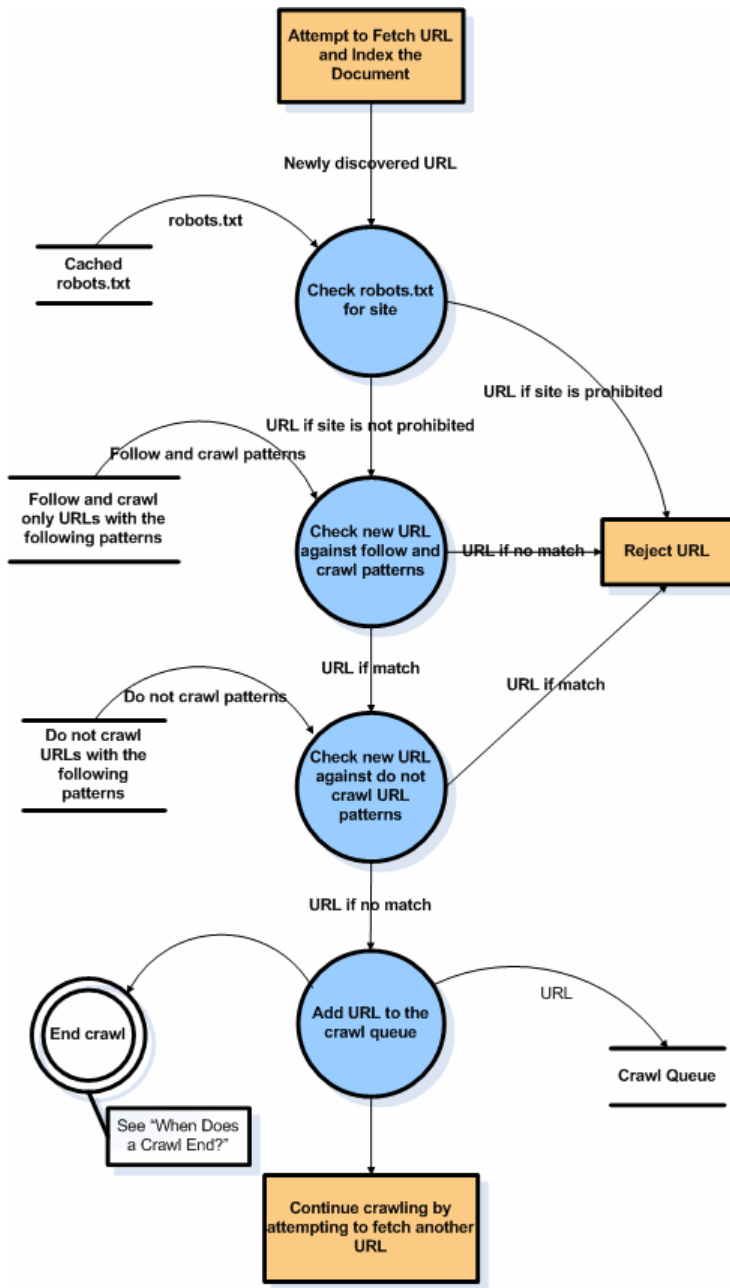
Following Links within the Document

For each document that it indexes, the Google Search Appliance follows newly discovered URLs (HTML links) within that document. When following URLs, the search appliance observes the index limit that is set on the **Index > Index Settings** page in the Admin Console. For example, if the index limit is 5MB, the search appliance only follows URLs within the first 5MB of a document. There is no limit to the number of URLs that can be followed from one document.

Before following a newly discovered link, the search appliance checks the URL against:

- The robots.txt file for the site
- Follow and crawl URL patterns
- Do not crawl URL patterns

If the URL passes these checks, the search appliance adds the URL to the crawl queue, and eventually crawls it. If the URL does not pass these checks, the search appliance deletes it from the crawl queue. The following diagram provides an overview of this process.



The search appliance crawler only follows HTML links in the following format:

```
<a href="/page2.html">link to page 2</a>
```

It follows HTML links in PDF files, Word documents, and Shockwave documents. The search appliance also supports JavaScript crawling (see “JavaScript Crawling” on page 63) and can detect links and content generated dynamically through JavaScript execution.

When Does Crawling End?

The Google Search Appliance administrator can end a continuous crawl by pausing it (see “Stopping, Pausing, or Resuming a Crawl” on page 43).

The search appliance administrator can configure a scheduled crawl to end at a specified time. A scheduled crawl also ends when the license limit is reached (see “What Is the Search Appliance License Limit?” on page 22). The following table provides more details about the conditions that cause a scheduled crawl to end.

Condition	Description
Scheduled end time	Crawling stops at its scheduled end time.
Crawl to completion	There are no more URLs in the crawl queue. The search appliance crawler has discovered and attempted to fetch all reachable content that matches the configured URL patterns.
The license limit is reached	The search appliance license limits the maximum number of URLs in the index. When the search appliance reaches this limit, it stops crawling new URLs. The search appliance removes the excess URLs (see “Are Documents Removed From the Index?” on page 25) from the crawl queue.

When Is New Content Available in Search Results?

For both scheduled crawls and continuous crawls, documents usually appear in search results approximately 30 minutes after they are crawled. This period can increase if the system is under a heavy load, or if there are many non-HTML documents (see “Non-HTML Content” on page 48).

For a recrawl, if an older version of a document is cached in the index from a previous crawl, the search results refer to the cached document until the new version is available.

How Are URLs Scheduled for Recrawl?

The search appliance determines the priority of URLs for recrawl using the following rules, listed in order from highest to lowest priority:

1. URLs that are designated for recrawl by the administrator- for example, when you request a certain URL pattern to be crawled by using the **Content Sources > Web Crawl > Start and Block URLs, Content Sources > Web Crawl > Freshness Tuning** or **Index > Diagnostics > Index Diagnostics** page in the Admin Console or sent in web feeds where the `crawl-immediately` attribute for the record is set to `true`.
2. URLs that are set to crawl frequently on the **Content Sources > Web Crawl > Freshness Tuning** page and have not been crawled in the last 23 hours.
3. URLs that have not been crawled yet.
4. URLs that have already been crawled. Crawled URLs’ priority is mostly based the number of links from a start URL. The last crawl date and frequency with which the URL changes also contribute to the priority of crawled URLs. URLs with a crawl date further in the past and that change more frequently also get higher priority.

There are some other factors that also contribute to whether a URL is recrawled, for example how fast a host can respond will also play a factor, or whether it received an error on the last crawl attempt.

If you need to give URLs high priority, you can do a few things to change their priority:

- You can submit a recrawl request by using the **Content Sources > Web Crawl > Start and Block URLs, Content Sources > Web Crawl > Freshness Tuning** or **Index > Diagnostics > Index Diagnostics** pages, which gives the URLs the highest priority possible.
- You can submit a web feed, which makes the URL's priority identical to an uncrawled URL's priority.
- You can add a URL to the **Crawl Frequently** list on the **Content Sources > Web Crawl > Freshness Tuning** page, which ensures that the URL gets crawled about every 24 hours.

To see how often a URL has been recrawled in the past, as well as the status of the URL, you can view the crawl history of a single URL by using the **Index > Diagnostics > Index Diagnostics** page in the Admin Console.

How Are Network Connectivity Issues Handled?

When crawling, the Google Search Appliance tests network connectivity by attempting to fetch every start URL every 30 minutes. If approximately 10% of the start URLs return HTTP 200 (OK) responses, the search appliance assumes that there are no network connectivity issues. If less than 10% return OK responses, the search appliance assumes that there are network connectivity issues with a content server and slows down or stops.

During a temporary network outage, slowing or stopping a crawl prevents the search appliance from removing URLs that it cannot reach from the index. The crawl speeds up or restarts when the start URL connectivity test returns an HTTP 200 response.

What Is the Search Appliance License Limit?

Your Google Search Appliance license determines the number of documents that can appear in your index, as listed in the following table.

Search Appliance Model	Maximum License Limit
GB-7007	10 million
GB-9009	30 million
G100	20 million
G500	100 million

Google Search Appliance License Limit

For a Google Search Appliance, between 500,000 and 100 million documents can appear in the index, depending on your model and license.

For example, if the license limit is 10 million, the search appliance crawler attempts to put the 10 million documents in the index. During a recrawl, when the crawler discovers a new URL, it must decide whether to crawl the document.

When the search appliance reaches its limit, it stops crawling new URLs, and removes documents from the index to bring the total number of documents to the license limit.

Google recommends managing crawl patterns on the **Content Sources > Web Crawl > Start and Block URLs** page in the Admin Console to ensure that the total number of URLs that match the crawl patterns remains at or below the license limit.

When Is a Document Counted as Part of the License Limit?

Generally, when the Google Search Appliance successfully fetches a document, it is counted as part of the license limit. If the search appliance does not successfully fetch a document, it is not counted as part of the license limit. The following table provides an overview of the conditions that determine whether or not a document is counted as part of the license limit.

Condition	Counted as Part of the License Limit?
The search appliance fetches a URL without errors. This includes HTTP responses 200 (success), 302 (redirect, URL moved temporarily), and 304 (not modified)	The URL is counted as part of the license limit.
The search appliance receives a 301 (redirect, URL moved permanently) when it attempts to fetch a document, and then fetches the URL without error at its destination.	The destination URL is counted as part of the license limit, but not the source URL, which is excluded.
The search appliance cannot fetch a URL. Instead, the search appliance receives an HTTP error response, such as 404 (document not found) or 500 (temporary server error).	The URL is not counted as part of the license limit.
The search appliance fetches two URLs that contain exactly the same content without errors.	Both URLs are counted as part of the license limit, but the one with the lower Enterprise PageRank is automatically filtered out of search results. It is not possible to override this automatic filtering.
The search appliance fetches a document from a file share.	The document is counted as part of the license limit.
The search appliance retrieves a list of files and subdirectories and in a file share and converts it to a directory listings page.	Each directory in the list is counted as part of the license limit, even if the directory is empty.
The search appliance retrieves a list of file shares on a host and converts it to a share listings page.	Each share in the list is counted as part of the license limit.
The SharePoint connector indexes a folder.	Each folder is indexed as a document and counted as part of the license limit.

If there are one or more robots meta tags embedded in the head of a document, they can affect whether the document is counted as part of the license limit. For more information about this topic, see “Using Robots meta Tags to Control Access to a Web Page” on page 30.

To view license information for your Google Search Appliance, use the **Administration > License** page. For more information about this page, click **Admin Console Help > Administration > License** in the Admin Console.

License Expiration and Grace Period

Google Search Appliance licensing has a grace period, which starts when the license expires and lasts for 30 days. During the 30-day grace period, the search appliance continues to crawl, index, and serve documents. At the end of the grace period, it stops crawling, indexing, and serving.

If you have configured your search appliance to receive email notifications, you will receive daily emails during the grace period. The emails notify you that your search appliance license has expired and it will stop crawling, indexing and serving in n days, where n is the number of days left in your grace period.

At the end of the grace period, the search appliance will send one email stating that the license has completely expired, the grace period has ended, and the software has stopped crawling,

indexing, and serving. The Admin Console on the search appliance will still be accessible at the end of the grace period.

To configure your search appliance to receive email notifications, use the **Administration > System Settings** page. For more information about this page, click **Admin Console Help > Administration > System Settings** in the Admin Console.

How Many URLs Can Be Crawled?

The Google Search Appliance crawler stores a maximum number of URLs that can be crawled. The maximum number depends on the search appliance model and license limit, as listed in the following table.

Search Appliance Model	Maximum License Limit	Maximum Number of URLs that Match Crawl Patterns
GB-7007	10 million	~ 13.6 million
GB-9009	30 million	~ 40 million
G100	20 million	~133 million
G500	100 million	~666 million

If the Google Search Appliance has reached the maximum number of URLs that can be crawled, this number appears in **URLs Found That Match Crawl Patterns** on the **Content Sources > Diagnostics > Crawl Status** page in the Admin Console.

Once the maximum number is reached, a new URL is considered for crawling only if it has a higher priority than the least important known URL. In this instance, the higher priority URL is crawled and the lower priority URL is discarded.

For an overview of the priorities assigned to URLs in the crawl queue, see “Starting the Crawl and Populating the Crawl Queue” on page 14.

How Are Document Dates Handled?

To enable search results to be sorted and presented based on dates, the Google Search Appliance extracts dates from documents according to rules configured by the search appliance administrator (see “Defining Document Date Rules” on page 41).

In Google Search Appliance software version 4.4.68 and later, document dates are extracted from Web pages when the document is indexed.

The search appliance extracts the first date for a document with a matching URL pattern that fits the date format associated with the rule. If a date is written in an ambiguous format, the search appliance assumes that it matches the most common format among URLs that match each rule for each domain that is crawled. For this purpose, a domain is one level above the top level. For example, `mycompany.com` is a domain, but `intranet.mycompany.com` is not a domain.

The search appliance periodically runs a process that calculates which of the supported date formats is the most common for a rule and a domain. After calculating the statistics for each rule and domain, the process may modify the dates in the index. The process first runs 12 hours after the search appliance is installed, and thereafter, every seven days. The process also runs each time you change the document date rules.

The search appliance will not change which date is most common for a rule until after the process has run. Regardless of how often the process runs, the search appliance will not change the date format more than once a day. The search appliance will not change the date format unless 5,000 documents have been crawled since the process last ran.

If you import a configuration file with new document dates after the process has first run, then you may have to wait at least seven days for the dates to be extracted correctly. The reason is that the date formats associated with the new rules are not calculated until the process runs. If no dates were found the first time the process ran, then no dates are extracted until the process runs again.

If no date is found, the search appliance indexes the document without a date.

Normally, document dates appear in search results about 30 minutes after they are extracted. In larger indexes, the process can several hours to complete because the process may have to look at the contents of every document.

The search appliance can extract date information from SMB/CIFS servers by using values from the file system attributes. To verify the date that is assigned to a document, use one of the following methods:

- Find the file by using Windows Explorer and check the entry in the Date Modified column.
- At the Windows command prompt, enter `dir filepath`.

Are Documents Removed From the Index?

The Google Search Appliance index includes all the documents it has crawled. These documents remain in the index and the search appliance continues to crawl them until either one of the following conditions is true:

- The search appliance administrator resets the index.
- The search appliance removes the document from the index during the document removal process.

The search appliance administrator can also remove documents from the index (see “Removing Documents from the Index” on page 62) manually.

Removing all links to a document in the index does not remove the document from the index.

Document Removal Process

The following table describes the conditions that cause documents to be removed from the index.

Condition	Description
The license limit is exceeded	The limit on the number of URLs in the index is the value of Maximum number of pages overall on the Administration > License page.
The crawl pattern is changed	<p>To determine which content should be included in the index, the search appliance uses the start urls, follow patterns, and do not follow URL patterns specified on the Content Sources > Web Crawl > Start and Block URLs page. If these URL patterns are modified, the search appliance examines each document in the index to determine whether it should be retained or removed.</p> <p>If the URL does not match any follow and crawl patterns, or if it matches any do not crawl patterns, it is removed from the index. Document URLs disappear from search results between 15 minutes and six hours after the pattern changes, depending on system load.</p>
The robots.txt file is changed	If the robots.txt file for a content server or web site has changed to prohibit search appliance crawler access, URLs for the server or site are removed from the index.
Authentication failure (401)	If the search appliance receives three successive 401 (authentication failure) errors from the Web server when attempting to fetch a document, the document is removed from the index after the third failed attempt.
Document is not found (404)	If the search appliance receives a 404 (Document not found) error from the Web server when attempting to fetch a document, the document is removed from the index.
Document is indexed, but removed from the content server.	See "What Happens When Documents Are Removed from Content Servers?" on page 26.

Note: Search appliance software versions prior to 4.6 include a process called the "remove doc ripper." This process removes documents from the index every six hours. If the appliance has crawled more documents than its license limit, the ripper removes documents that are below the Enterprise PageRank threshold. The ripper also removes documents that don't match any follow patterns or that do match exclude patterns. If you want to remove documents from search results, use the Remove URLs feature on the **Search > Search Features > Front Ends > Remove URLs** page. When the remove doc ripper has run with your changes to the crawl patterns, you should delete all Remove URL patterns. The Remove URL patterns are checked at search query time and are expensive to process. A large number of Remove URLs patterns affects search query speed.

What Happens When Documents Are Removed from Content Servers?

During the recrawl of an indexed document, the search appliance sends an If-Modified-Since header based on the last crawl date of the document. Even if a document has been removed from a content server, the search appliance makes several attempts to recrawl the URL before removing the document from the index.

When a document is removed from the index, it disappears from the search results. However, the search appliance maintains the document in its internal status table. For this reason, the URL might still appear in Index Diagnostics.

The following table lists the timing of recrawl attempts and removal of documents from the index based on different scenarios.

Scenario	Recrawl Attempts	Document Removal from the Index
The search appliance encounters an error during crawling that could be a server timeout error (500 error code) or forbidden (403 errors).	First recrawl attempt: 1 Day Second recrawl attempt: 3 Days Third recrawl attempt: 1 Week Fourth recrawl attempt: 3 Weeks	The document is removed if the search appliance encounters the error for the fourth time.
The search appliance encounters an unreachable message during crawling, which might be caused by network issues, such as DNS server issues.	First recrawl attempt: 5 hours Second recrawl attempt: 1 Day Third recrawl attempt: 5 Days Fourth recrawl attempt: 3 Weeks	The document is removed if the search appliance encounters the error for the fourth time.
The search appliance encounters issues caused by robots meta-tag setup, for example the search appliance is blocked by a robots meta-tag.	First recrawl attempt: 5 Days Second recrawl attempt: 15 Days Third recrawl attempt: 1 Month	The document is removed if the search appliance encounters the error for the third time.
The search appliance encounters garbage data, that is data that is similar to other documents, but which is not marked as considered duplicate	First recrawl attempt: 1 day Second recrawl attempt: 1 week Third recrawl attempt: 1 month Fourth recrawl attempt: 3 months	The document is removed if the search appliance encounters the error for the fourth time.

Chapter 2

Preparing for a Crawl

Crawling is the process where the Google Search Appliance discovers enterprise content to index. This chapter tells search appliance administrators and content owners how to prepare enterprise content for crawling.

Preparing Data for a Crawl

Before the Google Search Appliance crawls your enterprise content, people in various roles may want to prepare the content to meet the objectives described in the following table.

Objective	Role
Control access to a content server	Content server administrator, webmaster
Control access to a Web page	Search appliance administrator, webmaster, content owner, and/or content server administrator
Control indexing of parts of a Web page	
Control access to files and subdirectories	
Ensure that the search appliance can crawl a file system	

Using robots.txt to Control Access to a Content Server

The Google Search Appliance always obeys the rules in robots.txt (see “Content Prohibited by a robots.txt File” on page 11) and it is not possible to override this feature. However, this type of file is not mandatory. When a robots.txt file is present, it is located in the Web server’s root directory. For the search appliance to be able to access the robot.txt file, the file must be public.

Before the search appliance crawls any content servers in your environment, check with the content server administrator or webmaster to ensure that robots.txt allows the search appliance user agent access to the appropriate content (see “Identifying the User Agent” on page 57). For the search appliance to be able to access to the robot.txt file, the file must be public.

If any hosts require authentication before serving robots.txt, you must configure authentication credentials using the **Content Sources > Web Crawl > Secure Crawl > Crawler Access** page in the Admin Console.

A robots.txt file identifies a crawler as the `User-Agent`, and includes one or more `Disallow:` or `Allow:` (see “Using the Allow Directive” on page 29) directives, which inform the crawler of the content to be ignored. The following example shows a robots.txt file:

```
User-agent: gsa-crawler
Disallow: /personal_records/
```

`User-agent: gsa-crawler` identifies the Google Search Appliance crawler. `Disallow:` tells the crawler not to crawl and index content in the `/personal_records/` path.

To tell the search appliance crawler to ignore all of the content in a site, use the following syntax:

```
User-agent: gsa-crawler
Disallow: /
```

To allow the search appliance crawler to crawl and index all of the content in a site, use `Disallow:` without a value, as shown in the following example:

```
User-agent: gsa-crawler
Disallow:
```

Using the Allow Directive

In Google Search Appliance software versions 4.6.4.G.44 and later, the search appliance user agent (`gsa-crawler`, see “Identifying the User Agent” on page 57) obeys an extension to the robots.txt standard called “Allow.” This extension may not be recognized by all other search engine crawlers, so check with other search engines you’re interested in finding out. The `Allow:` directive works exactly like the `Disallow:` directive. Simply list a directory or page you want to allow.

You may want to use `Disallow:` and `Allow:` together. For example, to block access to all pages in a subdirectory except one, use the following entries:

```
User-Agent: gsa-crawler
Disallow: /folder1/
Allow: /folder1/myfile.html
```

This blocks all pages inside the `folder1` directory except for `myfile.html`.

Caching robots.txt

The Google Search Appliance caches robots.txt file for 30 minutes. You can clear the robots.txt file from cache and refresh it by changing the DNS Servers settings and then restoring them.

To clear the robots.txt file from cache and refresh it:

1. Choose **Administration > Network Settings**.
2. Change the **DNS Servers** settings.
3. Click **Update Settings and Perform Diagnostics**.
4. Restore the original **DNS Servers** settings.
5. Click **Update Settings and Perform Diagnostics**.

Using Robots meta Tags to Control Access to a Web Page

To prevent the search appliance crawler (as well as other crawlers) from indexing or following links in a specific HTML document, embed a robots meta tag in the head of the document. The search appliance crawler obeys the noindex, nofollow, noarchive, and none keywords in meta tags. Refer to the following table for details about Robots meta tags, including examples.

Keyword	Description	Example
noindex	The search appliance crawler does not archive the document in the search appliance cache or index it. The document is not counted as part of the license limit.	<code><meta name="robots" content="noindex"/></code>
nofollow	The search appliance crawler retrieves and archives the document in the search appliance cache, but does not follow links on the Web page to other documents. The document is counted as part of the license limit.	<code><meta name="robots" content="nofollow"/></code>
noarchive	The search appliance crawler retrieves and indexes the document, but does not archive it in its cache. The document is counted as part of the license limit.	<code><meta name="robots" content="noarchive"/></code>
none	The none tag is equal to <code><meta name="robots" content="noindex, nofollow, noarchive"/></code> .	<code><meta name="robots" content="none"/></code>

You can combine any or all of the keywords in a single meta tag, for example:

```
<meta name="robots" content="noarchive, nofollow"/>
```

Even if a robots meta tag contains words other than noindex, nofollow, noarchive, and none, if the keywords appear between separators, such as commas, the search appliance is able to extract that keyword correctly.

Also, you can include `rel="nofollow"` in an anchor tag, which causes the search appliance to ignore the link. For example:

```
<a href="test1.html" rel="nofollow">no follow</a>
```

Currently, it is not possible to set `name="gsa-crawler"` to limit these restrictions to the search appliance.

If the search encounters a robots meta tag when fetching a URL, it schedules a retry after a certain time interval. For URLs excluded by robots meta tags, the maximum retry interval is one month.

Using X-Robots-Tag to Control Access to Non-HTML Documents

While the robots meta tag gives you control over HTML pages, the X-Robots-Tag directive in an HTTP header response gives you control of other types of documents, such as PDF files.

For example, the following HTTP response with an X-Robots-Tag instructs the crawler not to index a page:

```
HTTP/1.1 200 OK
Date: Tue, 25 May 2010 21:42:43 GMT
(...)
X-Robots-Tag: noindex
(...)
```

The Google Search Appliance supports the X-Robots-Tag directives listed in the following table.

Directive	Description	Example
noindex	Do not show this page in search results and do not show a “Cached” link in search results.	X-Robots-Tag: noindex
nofollow	Do not follow the links on this page.	X-Robots-Tag: nofollow
noarchive	Do not show a “Cached” link in search results.	X-Robots-Tag: noarchive

Excluding Unwanted Text from the Index

There may be Web pages that you want to suppress from search results when users search on certain words or phrases. For example, if a Web page consists of the text “the user conference page will be completed as soon as Jim returns from medical leave,” you might not want this page to appear in the results of a search on the terms “user conference.”

You can prevent this content from being indexed by using googleoff/googleon tags. By embedding googleon/googleoff tags with their flags in HTML documents, you can disable:

- The indexing of a word or portion of a Web page
- The indexing of anchor text
- The use of text to create a snippet in search results

For details about each googleon/googleoff flag, refer to the following table.

Flag	Description	Example	Results
index	Words between the tags are not indexed as occurring on the current page.	<pre>fish <!--googleoff: index-->shark <!--googleon: index-- >mackerel</pre>	<p>The words fish and mackerel are indexed for this page, but the occurrence of shark is not indexed.</p> <p>This page could appear in search results for the term shark only if the word appears elsewhere on the page or in anchor text for links to the page.</p> <p>But the word shark could appear in a result snippet.</p> <p>Hyperlinks that appear within these tags are followed.</p>
anchor	Anchor text that appears between the tags and in links to other pages is not indexed. This prevents the index from using the hyperlink to associate the link text with the target page in search results.	<pre><!--googleoff: anchor-- > shark <!-- googleon: anchor--></pre>	<p>The word shark is not associated with the page sharks_rugby.html. Otherwise this hyperlink would cause the page sharks_rugby.html to appear in the search results for the term shark. Hyperlinks that appear within these tags are followed, so sharks_rugby.html is still crawled and indexed.</p>
snippet	Text between the tags is not used to create snippets for search results.	<pre><!--googleoff: snippet-- >Come to the fair! shark <!--googleon: snippet-- ></pre>	<p>The text ("Come to the fair!" and "shark") does not appear in snippets with the search results, but the words will still be indexed and searchable. Also, the link sharks_rugby.html will still be followed. The URL sharks_rugby.html will also appear in the search results for the term shark.</p>
all	Turns off all the attributes. Text between the tags is not indexed, is not associated with anchor text, or used for a snippet.	<pre><!--googleoff: all-- >Come to the fair! <!--googleon: all--></pre>	<p>The text Come to the fair! is not indexed, is not associated with anchor text, and does not appear in snippets with the search results.</p>

There must be a space or newline before the googleon tag.

If URL1 appears on page URL2 within googleoff and googleon tags, the search appliance still extracts the URL and adds it to the link structure. For example, the query `link:URL2` still contains URL1 in the result set, but depending on which googleoff option you use, you do not see URL1 when viewing the cached version, searching using the anchor text, and so on. If you want the search appliance not to follow the links and ignore the link structure, follow the instructions in "Using Robots meta Tags to Control Access to a Web Page" on page 30.

Using no-crawl Directories to Control Access to Files and Subdirectories

The Google Search Appliance does not crawl any directories named “no_crawl.” You can prevent the search appliance from crawling files and directories by:

1. Creating a directory called “no_crawl.”
2. Putting the files and subdirectories you do not want crawled under the no_crawl directory.

This method blocks the search appliance from crawling everything in the no_crawl directory, but it does not provide directory security or block people from accessing the directory.

End users can also use no_crawl directories on their local computers to prevent personal files and directories from being crawled.

Preparing Shared Folders in File Systems

In GSA release 7.4, on-board file system crawling (File System Gateway) is deprecated. For more information, see [Deprecation Notices](#).

In a Windows network file system, folders and drives can be shared. A shared folder or drive is available for any person, device, or process on the network to use. To enable the Google Search Appliance to crawl your file system, do the following:

1. Set the properties of appropriate folders and drives to “Share this folder.”
2. Check that the content to be crawled is in the appropriate folders and drives.

Ensuring that Unlinked URLs Are Crawled

The Google Search Appliance crawls content by following newly discovered links in pages that it crawls. If your enterprise content includes unlinked URLs that are not listed in the follow and crawl patterns, the search appliance crawler will not find them on its own. In addition to adding unlinked URLs to follow and crawl patterns, you can force unlinked URLs into a crawl by using a jump page, which lists any URLs and links that you want the search appliance crawl to discover.

A jump page allows users or crawlers to navigate all the pages within a Web site. To include a jump page in the crawl, add the URL for the page to the crawl path.

Configuring a Crawl

Before starting a crawl, you must configure the crawl path so that it only includes information that your organization wants to make available in search results. To configure the crawl, use the **Content Sources > Web Crawl > Start and Block URLs** page in the Admin Console to enter URLs and URL patterns in the following boxes:

- **Start URLs**
- **Follow Patterns**
- **Do Not Follow Patterns**

Note: URLs are case-sensitive.

If the search appliance should never crawl outside of your intranet site, then Google recommends that you take one or more of the following actions:

- Configure your network to disallow search appliance connectivity outside of your intranet.

If you want to make sure that the search appliance never crawls outside of your intranet, then a person in your IT/IS group needs to specifically block the search appliance IP addresses from leaving your intranet.
- Make sure all patterns in the field Follow Patterns specify `yourcompany.com` as the domain name.

Note: Some content servers do not respond correctly to crawl requests from the search appliance. When this happens, the URL state on the Admin Console may appear as:

```
Error: Malformed HTTP header: empty content.
```

To crawl documents when this happens, you can add a header on the **Content Sources > Web Crawl > HTTP Headers** page of the Admin Console. In the **Additional HTTP Headers for Crawler** field, add:

```
Accept-Encoding: identity
```

For complete information about the **Content Sources > Web Crawl > Start and Block URLs** page, click **Admin Console Help > Content Sources > Web Crawl > Start and Block URLs** in the Admin Console.

Start URLs

Start URLs control where the Google Search Appliance begins crawling your content. The search appliance should be able to reach all content that you want to include in a particular crawl by following the links from one or more of the start URLs. Start URLs are required.

Start URLs must be fully qualified URLs in the following format:

```
<protocol>://<host>{:port}/{path}
```

The information in the curly brackets is optional. The forward slash "/" after `<host>{:port}` is required.

Typically, start URLs include your company's home site, as shown in the following example:

```
http://mycompany.com/
```

The following example shows a valid start URL:

```
http://www.example.com/help/
```

The following table contains examples of invalid URLs

Invalid examples	Reason:
<code>http://www/</code>	Invalid because the hostname is not fully qualified. A fully qualified hostname includes the local hostname and the full domain name. For example: <code>mail.corp.company.com</code> .
<code>www.example.com/</code>	Invalid because the protocol information is missing.
<code>http://www.example.com</code>	The <code>"/</code> after <code><host>[:port]</code> is required.

The search appliance attempts to resolve incomplete path information entered, using the information entered on the **Administration > Network Settings** page in the **DNS Suffix (DNS Search Path)** section. However, if it cannot be successfully resolved, the following error message displays in red on the page:

```
You have entered one or more invalid start URLs. Please check your edits.
```

The crawler will retry several times to crawl URLs that are temporarily unreachable.

These URLs are *only* the starting point(s) for the crawl. They tell the crawler where to begin crawling. However, links from the start URLs will be followed and indexed only if they match a pattern in **Follow Patterns**. For example, if you specify a starting URL of `http://mycompany.com/` in this section and a pattern `www.mycompany.com/` in the **Follow Patterns** section, the crawler will discover links in the `http://www.mycompany.com/` web page, but will only crawl and index URLs that match the pattern `www.mycompany.com/`.

Enter start URLs in the **Start URLs** section on the **Content Sources > Web Crawl > Start and Block URLs** page in the Admin Console. To crawl content from multiple websites, add start URLs for them.

Follow Patterns

Follow and crawl URL patterns control which URLs are crawled and included in the index. Before crawling any URLs, the Google Search Appliance checks them against follow and crawl URL patterns. Only URLs that match these URL patterns are crawled and indexed. You must include all start URLs in follow and crawl URL patterns.

The following example shows a follow and crawl URL pattern:

```
http://www.example.com/help/
```

Given this follow and crawl URL pattern, the search appliance crawls the following URLs because each one matches it:

```
http://www.example.com/help/two.html  
http://www.example.com/help/three.html
```

However, the search appliance does not crawl the following URL because it does not match the follow and crawl pattern:

```
http://www.example.com/us/three.html
```

The following table provides examples of how to use follow and crawl URL patterns to match sites, directories, and specific URLs.

To Match	Expression Format	Example
A site	<site>/	www.mycompany.com/
URLs from all sites in the same domain	<domain>/	mycompany.com/
URLs that are in a specific directory or in one of its subdirectories	<site>/<directory>/	sales.mycompany.com/products/
A specific file	<site>/<directory>/<file>	www.mycompany.com/products/index.html

For more information about writing URL patterns, see “Constructing URL Patterns” on page 86.

Enter follow and start URL patterns in the **Follow Patterns** section on the **Content Sources > Web Crawl > Start and Block URLs** page in the Admin Console.

Do Not Follow Patterns

Do not follow patterns exclude URLs from being crawled and included in the index. If a URL contains a do not crawl pattern, the Google Search Appliance does not crawl it. Do not crawl patterns are optional.

Enter do not crawl URL patterns in the **Do Not Follow Patterns** section on the **Content Sources > Web Crawl > Start and Block URLs** page in the Admin Console.

To prevent specific file types, directories, or other sets of pages from being crawled, enter the appropriate URLs in this section. Using this section, you can:

- Prevent certain URLs, such as email links, from consuming your license limit.
- Protect files that you do not want people to see.
- Save time while crawling by eliminating searches for objects such as MP3 files.

For your convenience, this section is prepopulated with many URL patterns and file types, some of which you may not want the search appliance to index. To make a pattern or file type unavailable to the search appliance crawler, remove the # (comment) mark in the line containing the file type. For example, to make Excel files on your servers unavailable to the crawler, change the line

```
#.xls$
```

to

```
.xls$
```

Crawling and Indexing Compressed Files

The search appliance supports crawling and indexing compressed files in the following formats: .zip, .tar, .tar.gz, and .tgz.

To enable the search appliance to crawl these types of compressed files, use the **Do Not Follow Patterns** section on the **Content Sources > Web Crawl > Start and Block URLs** page in the Admin Console. Put a "#" in front of the following patterns:

- .tar\$
- .zip\$
- .tar .gz\$
- .tgz\$
- `regexIgnoreCase:([^.]. |[^p]. |[^s])[.]gz$`

Testing Your URL Patterns

To confirm that URLs can be crawled, you can use the **Pattern Tester Utility** page. This page finds which URLs will be matched by the patterns you have entered for:

- **Follow Patterns**
- **Do Not Follow Patterns**

To use the **Pattern Tester Utility** page, click **Test these patterns** on the **Content Sources > Web Crawl > Start and Block URLs** page. For complete information about the **Pattern Tester Utility** page, click **Admin Console Help > Content Sources > Web Crawl > Start and Block URLs** in the Admin Console.

Using Google Regular Expressions as Crawl Patterns

The search appliance's Admin Console accepts Google regular expressions (similar to GNU regular expressions) as crawl patterns, but not all of these are valid in the Robots Exclusion Protocol. Therefore, the Admin Console does not accept Robots Exclusion Protocol patterns that are not valid Google regular expressions. Similarly, Google or GNU regular expressions cannot be used in robots.txt unless they are valid under the Robots Exclusion Protocol.

Here are some examples:

- The asterisk (*) is a valid wildcard character in both GNU regular expressions and the Robots Exclusion Protocol, and can be used in the Admin Console or in robots.txt.
- The \$ and ^ characters indicate the end or beginning of a string, respectively, in GNU regular expressions, and can be used in the Admin Console. They are not valid delimiters for a string in the Robots Exclusions Protocol, however, and cannot be used as anchors in robots.txt.
- The "Disallow" directive is used in robots.txt to indicate that a resource should not be visited by web crawlers. However, "Disallow" is not a valid directive in Google or GNU regular expressions, and cannot be used in the Admin Console.

Configuring Database Crawl

In GSA release 7.4, the on-board database crawler is deprecated. For more information, see [Deprecation Notices](#).

To configure a database crawl, provide database data source information by using the **Create New Database Source** section on the **Content Sources > Databases** page in the Admin Console.

For information about configuring a database crawl, refer to “Providing Database Data Source Information” on page 76.

About SMB URLs

In GSA release 7.4, on-board file system crawling (File System Gateway) is deprecated. For more information, see [Deprecation Notices](#).

As when crawling HTTP or HTTPS web-based content, the Google Search Appliance uses URLs to refer to individual objects that are available on SMB-based file systems, including files, directories, shares, hosts.

Use the following format for an SMB URL:

```
smb://string1/string2/...
```

When the crawler sees a URL in this format, it treats string1 as the hostname and string2 as the share name, with the remainder as the path within the share. Do not enter a workgroup in an SMB URL.

The following example shows a valid SMB URL for crawl:

```
smb://fileserver.mycompany.com/mysharemydir/mydoc.txt
```

The following table describes all of the required parts of a URL that are used to identify an SMB-based document.

URL Component	Description	Example
Protocol	Indicates the network protocol that is used to access the object.	smb://
Hostname	Specifies the DNS host name. A hostname can be one of the following:	
	A fully qualified domain name	fileserver.mycompany.com
	An unqualified hostname	fileserver
	An IP Address	10.0.0.100
Share name	Specifies the name of the share to use. A share is tied to a particular host, so two shares with the same name on different hosts do not necessarily contain the same content.	myshare
File path	Specifies the path to the document, relative to the root share.	If myshare on myhost.mycompany.com shares all the documents under the C:\myshare directory, the file C:\myshare\mydir\mydoc.txt is retrieved by the following: smb://myhost.mycompany.com/myshare/mydir/mydoc.txt
Forward slash	SMB URLs use forward slashes only. Some environments, such as Microsoft Windows systems, use backslashes (“\”) to separate file path components. Even if you are referring to documents in such an environment, use forward slashes for this purpose.	Microsoft Windows style: C:\myshare\ SMB URL: smb://myhost.mycompany.com/myshare/

In addition, ensure that the file server accepts inbound TCP connections on ports 139, 445. Port 139 is used to send NETBIOS requests for SMB crawling and port 445 is used to send Microsoft CIFS requests for SMB crawling. These ports on the file server need to be accessible by the search appliance. For information about checking the accessibility of these ports on the file server, see “Authentication Required (401) or Document Not Found (404) for SMB File Share Crawls” on page 52.

Unsupported SMB URLs

Some SMB file share implementations allow:

- URLs that omit the hostname
- URLs with workgroup identifiers in place of hostnames

The file system crawler does not support these URL schemes.

SMB URLs for Non-file Objects

SMB URLs can refer to objects other than files, including directories, shares, and hosts. The file system gateway, which interacts with the network file shares, treats these non-document objects like documents that do not have any content, but do have links to certain other objects. The following table describes the correspondence between objects that the URLs can refer to and what they actually link to.

URL Refers To	URL Links To	Example
Directory	Files and subdirectories contained within the directory	<code>smb://fileserver.mycompany.com/myshare/mydir/</code>
Share	Files and subdirectories contained within the share's top-level directory	<code>smb://fileserver.mycompany.com/myshare/</code>

Hostname Resolution

Hostname resolution is the process of associating a symbolic hostname with a numeric address that is used for network routing. For example, the symbolic hostname `www.google.com` resolves to the numeric address `10.0.0.100`.

File system crawling supports Domain Name Services (DNS), the standard name resolution method used by the Internet; it may not cover an internal network. During setup, the search appliance requires that at least one DNS server be specified. When crawling a host a search appliance will perform a DNS request if 30 minutes have passed since the previous request.

Setting Up the Crawler's Access to Secure Content

The information in this document describes crawling public content. For information about setting up the crawler's access to secure content, see the "Overview" in *Managing Search for Controlled-Access Content*.

Configuring Searchable Dates

For dates to be properly indexed and searchable by date range, they must be in ISO 8601 format:

```
YYYY-MM-DD
```

The following example shows a date in ISO 8601 format:

```
2007-07-11
```

For a date in a meta tag to be indexed, not only must it be in ISO 8601 format, it must also be the only value in the content. For example, the date in the following meta tag can be indexed:

```
<meta name="date" content="2007-07-11">
```

The date in the following meta tag cannot be indexed because there is additional content:

```
<meta name="date" content="2007-07-11 is a date">
```


Defining Document Date Rules

Documents can have dates explicitly stated in these places:

- URL
- Title
- Body of the document
- meta tags of the document
- Last-modified date from the HTTP response

To define a rule that the search appliance crawler should use to locate document dates (see “How Are Document Dates Handled?” on page 24) in documents for a particular URL, use the **Index > Document Dates** page in the Admin Console. If you define more than one document date rule for a URL, the search appliance finds all the matching dates from document and uses the first matching rule (from top to bottom) as its document date.

To configure document dates:

1. Choose **Index > Document Dates**. The **Document Dates** page appears.
2. In the **Host or URL Pattern** box, enter the host or URL pattern for which you want to set the rule.
3. Use the **Locate Date In** drop-down list to select the location of the date for the document in the specified URL pattern.
4. If you select **Meta Tag**, specify the name of the tag in the **Meta Tag Name** box. Make sure that you find a meta tag in your HTML. For example, for the tag `<meta name="publication_date">`, enter “publication_date” in the **Meta Tag Name** box.
5. To add another date rule, click **Add More Lines**, and add the rule.
6. Click **Save**. This triggers the Documents Dates process to run.

For complete information about the **Document Dates** page, click **Admin Console Help > Index > Document Dates** in the Admin Console.

Chapter 3

Running a Crawl

Crawling is the process where the Google Search Appliance discovers enterprise content to index. This chapter tells search appliance administrators how to start a crawl.

Selecting a Crawl Mode

Before crawling starts, you must use the **Content Sources > Web Crawl > Crawl Schedule** page in the Admin Console to select one of the following the crawl modes:

- Continuous crawl mode (see “Continuous Crawl” on page 8)
- Scheduled crawl mode (see “Scheduled Crawl” on page 8)

If you select **scheduled crawl**, you must schedule a time for crawling to start and a duration for the crawl (see “Scheduling a Crawl” on page 42). If you select and save **Continuous crawl** mode, crawling starts and a link to the **Freshness Tuning** page appears (see “Freshness Tuning” on page 58).

For complete information about the **Content Sources > Web Crawl > Crawl Schedule** page, click **Admin Console Help > Content Sources > Web Crawl > Crawl Schedule** in the Admin Console.

Scheduling a Crawl

The search appliance starts crawling in scheduled crawl mode according to a schedule that you can specify using the **Content Sources > Web Crawl > Crawl Schedule** page in the Admin Console. Using this page, you can specify:

- The day, hour, and minute when crawling should start
- Maximum duration for crawling

Stopping, Pausing, or Resuming a Crawl

Using the **Content Sources > Diagnostics > Crawl Status** page in the Admin Console, you can:

- Stop crawling (scheduled crawl mode)
- Pause crawling (continuous crawl mode)
- Resume crawling (continuous crawl mode)

When you stop crawling:

- The documents that were crawled remain in the index
- The index contains some old documents and some newly crawled documents

When you pause crawling, the Google Search Appliance only stops crawling documents in the index. Connectivity tests still run every 30 minutes for Start URLs. You may notice this activity in access logs.

For complete information about the **Content Sources > Diagnostics > Crawl Status** page, click **Admin Console Help > Content Sources > Diagnostics > Crawl Status** in the Admin Console.

Submitting a URL to Be Recrawled

Occasionally, there may be a recently changed URL that you want to be recrawled sooner than the Google Search Appliance has it scheduled for recrawling (see “How Are URLs Scheduled for Recrawl?” on page 21). Provided that the URL has been previously crawled, you can submit it for immediate recrawling from the Admin Console using one of the following methods:

- Selecting **Recrawl** from the **Actions** menu for a start URL or follow pattern on the **Content Sources > Web Crawl > Start and Block URLs** page in the Admin Console.
- Using the **Recrawl these URL Patterns** box on the **Content Sources > Web Crawl > Freshness Tuning** page in the Admin Console (see “Freshness Tuning” on page 58)
- Clicking **Recrawl this URL** in a detail view of a URL on the **Index > Diagnostics > Index Diagnostics** page in the Admin Console (see “Using the Admin Console to Monitor a Crawl” on page 45)

URLs that you submit for recrawling are treated the same way as new, uncrawled URLs in the crawl queue. They are scheduled to be crawled in order of Enterprise PageRank, and before any URLs that the search appliance has automatically scheduled for recrawling.

How quickly the search appliance can actually crawl these URLs depends on multiple other factors, such as network latency, content server responsiveness, and existing documents already queued up. A good place to check is the **Content Sources > Diagnostics > Crawl Queue** page (see “Using the Admin Console to Monitor a Crawl” on page 45), where you can observe the crawler backlog to ensure there isn't a content server acting as a bottleneck in the crawl progress.

Starting a Database Crawl

In GSA release 7.4, the on-board database crawler is deprecated. For more information, see [Deprecation Notices](#).

The process of crawling a database is called “synchronizing” a database. After you configure database crawling (see “Configuring Database Crawl” on page 38), you can start synchronizing a database by using the **Content Sources > Databases** page in the Admin Console.

To synchronize a database:

1. Click **Content Sources > Databases**.
2. In the **Current Databases** section of the page, click the **Sync** link next to the database that you want to synchronize.

The database synchronization runs until it is complete.

For more information about starting a database crawl, refer to “Database Crawling and Serving” on page 72.

Chapter 4

Monitoring and Troubleshooting Crawls

Crawling is the process where the Google Search Appliance discovers enterprise content to index. This chapter tells search appliance administrators how to monitor a crawl. It also describes how to troubleshoot some common problems that may occur during a crawl.

Using the Admin Console to Monitor a Crawl

The Admin console provides **Reports** pages that enable you to monitor crawling. The following table describes monitoring tasks that you can perform using these pages.

Task	Admin Console Page	Comments
Monitor crawling status	Content Sources > Diagnostics > Crawl Status	<p>While the Google Search Appliance is crawling, you can view summary information about events of the past 24 hours using the Content Sources > Diagnostics > Crawl Status page.</p> <p>You can also use this page to stop a scheduled crawl, or to pause or restart a continuous crawl (see “Stopping, Pausing, or Resuming a Crawl” on page 43).</p>
Monitor crawling crawl	Index > Diagnostics > Index Diagnostics	<p>While the Google Search Appliance is crawling, you can view its history using the Index > Diagnostics > Index Diagnostics page. Index diagnostics, as well as search logs and search reports, are organized by collection (see “Using Collections” on page 62).</p> <p>When the Index > Diagnostics > Index Diagnostics page first appears, it shows the crawl history for the current domain. It shows each URL that has been fetched and timestamps for the last 10 fetches. If the fetch was not successful, an error message is also listed.</p> <p>From the domain level, you can navigate to lower levels that show the history for a particular host, directory, or URL. At each level, the Index > Diagnostics > Index Diagnostics page displays information that is pertinent to the selected level.</p> <p>At the URL level, the Index > Diagnostics > Index Diagnostics page shows summary information as well as a detailed Crawl History.</p> <p>You can also use this page to submit a URL for recrawl (see “Submitting a URL to Be Recrawled” on page 43).</p>
Take a snapshot of the crawl queue	Content Sources > Diagnostics > Crawl Queue	<p>Any time while the Google Search Appliance is crawling, you can define and view a snapshot of the queue using the Content Sources > Diagnostics > Crawl Queue page. A crawl queue snapshot displays URLs that are waiting to be crawled, as of the moment of the snapshot.</p> <p>For each URL, the snapshot shows:</p> <ul style="list-style-type: none"> • Enterprise PageRank • Last crawled time • Next scheduled crawl time • Change interval
View information about crawled files	Index > Diagnostics > Content Statistics	<p>At any time while the Google Search Appliance is crawling, you can view summary information about files that have been crawled using the Index > Diagnostics > Content Statistics page. You can also use this page to export the summary information to a comma-separated values file.</p>

Crawl Status Messages

In the **Crawl History** for a specific URL on the **Index > Diagnostics > Index Diagnostics** page, the **Crawl Status** column lists various messages, as described in the following table.

Crawl Status Message	Description
Crawled: New Document	The Google Search Appliance successfully fetched this URL.
Crawled: Cached Version	The Google Search Appliance crawled the cached version of the document. The search appliance sent an if-modified-since field in the HTTP header in its request and received a 304 response, indicating that the document is unchanged since the last crawl.
Retrying URL: Connection Timed Out	The Google Search Appliance set up a connection to the Web server and sent its request, but the Web server did not respond within three minutes or the HTTP transaction didn't complete after 3 minutes.
Retrying URL: Host Unreachable while trying to fetch robots.txt	The Google Search Appliance could not connect to a Web server when trying to fetch robots.txt.
Retrying URL: Network unreachable during fetch	The Google Search Appliance could not connect to a Web server due to networking issue.
Retrying URL: Received 500 server error	The Google Search Appliance received a 500 status message from the Web server, indicating that there was an internal error on the server.
Excluded: Document not found (404)	The Google Search Appliance did not successfully fetch this URL. The Web server responded with a 404 status, which indicates that the document was not found. If a URL gets a status 404 when it is recrawled, it is removed from the index within 30 minutes.
Cookie Server Failed	The Google Search Appliance did not successfully fetch a cookie using the cookie rule. Before crawling any Web pages that match patterns defined for Forms Authentication, the search appliance executes the cookie rules.
Error: Permanent DNS failure	<p>The Google Search Appliance cannot resolve the host. Possible reasons can be a change in your DNS servers while the appliance still tries to access the previously cached IP.</p> <p>The crawler caches the results of DNS queries for a long time regardless of the TTL values specified in the DNS response. A workaround is to save and then revert a pattern change on the Content Sources > Web Crawl > Proxy Servers page. Saving changes here causes internal processes to restart and flush out the DNS cache.</p>

Network Connectivity Test of Start URLs Failed

When crawling, the Google Search Appliance tests network connectivity by attempting to fetch every start URL every 30 minutes. If less than 10% return OK responses, the search appliance assumes that there are network connectivity issues with a content server and slows down or stops and displays the following message: "Crawl has stopped because network connectivity test of Start URLs failed." The crawl restarts when the start URL connectivity test returns an HTTP 200 response.

Slow Crawl Rate

The **Content Sources > Diagnostics > Crawl Status** page in the Admin Console displays the **Current Crawling Rate**, which is the number of URLs being crawled per second. Slow crawling may be caused by the following factors:

- “Non-HTML Content” on page 48
- “Complex Content” on page 48
- “Host Load” on page 48
- “Network Problems” on page 49
- “Slow Web Servers” on page 49
- “Query Load” on page 49

These factors are described in the following sections.

Non-HTML Content

The Google Search Appliance converts non-HTML documents, such as PDF files and Microsoft Office documents, to HTML before indexing them. This is a CPU-intensive process that can take up to five seconds per document. If more than 100 documents are queued up for conversion to HTML, the search appliance stops fetching more URLs.

You can see the HTML that is produced by this process by clicking the cached link for a document in the search results.

If the search appliance is crawling a single UNIX/Linux Web server, you can run the tail command-line utility on the server access logs to see what was recently crawled. The tail utility copies the last part of a file. You can also run the `tcpdump` command to create a dump of network traffic that you can use to analyze a crawl.

If the search appliance is crawling multiple Web servers, it can crawl through a proxy.

Complex Content

Crawling many complex documents can cause a slow crawl rate.

To ensure that static complex documents are not recrawled as often as dynamic documents, add the URL patterns to the **Crawl Infrequently** URLs on the **Content Sources > Web Crawl > Freshness Tuning** page (see “Freshness Tuning” on page 58).

Host Load

If the Google Search Appliance crawler receives many temporary server errors (500 status codes) when crawling a host, crawling slows down.

To speed up crawling, you may need to increase the value of concurrent connections to the Web server by using the **Content Sources > Web Crawl > Host Load Schedule** page (see “Configuring Web Server Host Load Schedules” on page 61).

Network Problems

Network problems, such as latency, packet loss, or reduced bandwidth can be caused by several factors, including:

- Hardware errors on a network device
- A switch port set to a wrong speed or duplex
- A saturated CPU on a network device

To find out what is causing a network problem, you can run tests from a device on the same network as the search appliance.

Use the `wget` program (available on most operating systems) to retrieve some large files from the Web server, with both crawling running and crawling paused. If it takes significantly longer with crawling running, you may have network problems.

Run the `tracert` network tool from a device on the same network as the search appliance and the Web server. If your network does not permit Internet Control Message Protocol (ICMP), then you can use `tcptracert`. You should run the `tracert` with both crawling running and crawling paused. If it takes significantly longer with crawling running, you may have network performance problems.

Packet loss is another indicator of a problem. You can narrow down the network hop that is causing the problem by seeing if there is a jump in the times taken at one point on the route.

Slow Web Servers

If response times are slow, you may have a slow Web server. To find out if your Web server is slow, use the `wget` command to retrieve some large files from the Web server. If it takes approximately the same time using `wget` as it does while crawling, you may have a slow Web server.

You can also log in to a Web server to determine whether there are any internal bottlenecks.

If you have a slow host, the search appliance crawler fetches lower-priority URLs from other hosts while continuing to crawl the slower host.

Query Load

The crawl processes on the search appliance are run at a lower priority than the processes that serve results. If the search appliance is heavily loaded serving search queries, the crawl rate drops.

Wait Times

During continuous crawling, you may find that the Google Search Appliance is not recrawling URLs as quickly as specified by scheduled crawl times in the crawl queue snapshot. The amount of time that a URL has been in the crawl queue past its scheduled recrawl time is the URL's "wait time."

Wait times can occur when your enterprise content includes:

- Large numbers of documents
- Large PDF files or Microsoft Office documents
- Many frequently changing URLs
- New content with high Enterprise PageRank

If the search appliance crawler needs four hours to catch up to the URLs in the crawl queue whose scheduled crawl time has already passed, the wait time for crawling the URLs is four hours. In extreme cases, wait times can be several days. The search appliance cannot recrawl a URL more frequently than the wait time.

It is not possible for an administrator to view the maximum wait time for URLs in the crawl queue or to view the number of URLs in the queue whose scheduled crawl time has passed. However, you can use the **Content Sources > Diagnostics > Crawl Queue** page to create a crawl queue snapshot, which shows:

- Last time a URL was crawled
- Next scheduled crawl time for a URL

Errors from Web Servers

If the Google Search Appliance receives an error when fetching a URL, it records the error in **Index > Diagnostics > Index Diagnostics**. By default, the search appliance takes action based on whether the error is permanent or temporary:

- **Permanent errors**—Permanent errors occur when the document is no longer reachable using the URL. When the search appliance encounters a permanent error, it removes the document from the crawl queue; however, the URL is not removed from the index.
- **Temporary errors**—Temporary errors occur when the URL is unavailable because of a temporary move or a temporary user or server error. When the search appliance encounters a temporary error, it retains the document in the crawl queue and the index, and schedules a series of retries after certain time intervals, known as "backoff" intervals, before removing the URL from the index. The search appliance maintains an error count for each URL, and the time interval between retries, increases as the error count rises. The maximum backoff interval is three weeks.

You can either use the search appliance default settings for index removal and backoff intervals, or configure the following options for the selected error state:

- **Immediate Index Removal**—Select this option to immediately remove the URL from the index
- **Number of Failures for Index Removal**—Use this option to specify the number of times the search appliance is to retry fetching a URL
- **Successive Backoff Intervals (hours)**—Use this option to specify the number of hours between backoff intervals

To configure settings, use the options in the **Configure Backoff Retries and Remove Index Information** section of the **Content Sources > Web Crawl > Crawl Schedule** page in the Admin Console. For more information about configuring settings, click **Admin Console Help > Content Sources > Web Crawl > Crawl Schedule**.

The following table lists permanent and temporary Web server errors. For detailed information about HTTP status codes, see http://en.wikipedia.org/wiki/List_of_HTTP_status_codes.

Error	Type	Description
301	Permanent	Redirect, URL moved permanently.
302	Temporary	Redirect, URL moved temporarily.
401	Temporary	Authentication required.
404	Temporary	Document not found. URLs that get a 404 status response when they are recrawled are removed from the index within 30 minutes.
500	Temporary	Temporary server error.
501	Permanent	Not implemented.

In addition, the search appliance crawler refrains from visiting Web pages that have noindex and nofollow Robots META tags. For URLs excluded by Robots META tags, the maximum retry interval is one month.

You can view errors for a specific URL in the **Crawl Status** column on the **Index > Diagnostics > Index Diagnostics** page.

URL Moved Permanently Redirect (301)

When the Google Search Appliance crawls a URL that has moved permanently, the Web server returns a 301 status. For example, the search appliance crawls the old address, `http://myserver.com/301-source.html`, and is redirected to the new address, `http://myserver.com/301-destination.html`. On the **Index > Diagnostics > Index Diagnostics** page, the **Crawl Status** of the URL displays “Source page of permanent redirect” for the source URL and “Crawled: New Document” for the destination URL.

In search results, the URL of the 301 redirect appears as the URL of the destination page.

For example, if a user searches for `info:http://myserver.com/301-<source>.html`, the results display `http://myserver.com/301-<destination>.html`.

To enable search results to display a 301 redirect, ensure that start and follow URL patterns on the **Content Sources > Web Crawl > Start and Block URLs** page match both the source page and the destination page.

URL Moved Temporarily Redirect (302)

When the Google Search Appliance crawls a URL that has moved temporarily, the Web server returns a 302 status. On the **Index > Diagnostics > Index Diagnostics** page, the **Crawl Status** of the URL shows the following value for the source page:

- Crawled: New Document

There is no entry for the destination page in a 302 redirect.

In search results, the URL of the 302 redirect appears as the URL of the source page.

If the redirect destination URL does not match a Follow pattern, or matches a Do Not Follow Pattern, on the **Content Sources > Web Crawl > Start and Block URLs** page, the document does not display in search results. On the **Index > Diagnostics > Index Diagnostics** page, the **Crawl Status** of the URL shows the following value for the source page:

- Excluded: In "Do Not Crawl" URLs.

A META tag that specifies `http-equiv="refresh"` is handled as a 302 redirect.

Authentication Required (401) or Document Not Found (404) for SMB File Share Crawls

When the Google Search Appliance attempts to crawl content on SMB-based file systems, the web server might return 401 or 404 status. If this happens, take the following actions:

- Ensure that the URL patterns entered on the **Content Sources > Web Crawl > Start and Block URLs** page are in the format `smb://.//`
- Ensure that you have entered the appropriate patterns for authentication on the **Content Sources > Web Crawl > Secure Crawl > Crawler Access** page.
- If the document that returns the error requires authentication, ensure that:
 - The authentication rule is appropriately configured with a URL pattern for this document or set of documents
 - You have provided the proper user name, domain and password for the document
 - There are no special characters in the password. If the password includes special characters, you might try to set one without special characters to see if it resolves the issue

On the file share server, ensure that the directories or files you have configured for crawling are not empty. Also, on the file share server (in the configuration panel), verify that:

- The file share is not part of a Distributed File System (DFS) configuration
- Basic Authentication or NTLM is used as the authentication protocol
- Permissions are set properly (read access for user on this share, allow various permission sets including listings of files in the Share's settings)
- For a Windows file share, the read permissions are set specifically for the configured user in the Security tab in Share properties dialog

Also, ensure that the file server accepts inbound TCP connections on ports 139, 445. These ports on the file share need to be accessible by the search appliance. You can verify whether the ports are open by using the `nmap` command on a machine on the same subnet as the search appliance. Run the following command:

```
nmap <fileshare host> -p 139,445
```

The response needs to be "open" for both. If the `nmap` command is not available on the machine you are using, you can use the `telnet` command for each of the ports individually. Run the following commands:

```
telnet <fileshare-host> 139
telnet <fileshare-host> 445
```

A connection should be established rather than refused.

If the search appliance is crawling a Windows file share, verify that NTLMv2 is enabled on the Windows file share by following section 10 in Microsoft Support's document (<http://support.microsoft.com/kb/823659>). Take note that NTLMv1 is very insecure and is not supported.

Take note that you can also use a script on the Google Search Appliance Admin Toolkit project page for additional diagnostics outside the search appliance. To access the script, visit <http://gsa-admin-toolkit.googlecode.com/svn/trunk/smbcrawler.py>.

Cyclic Redirects

A cyclic redirect is a request for a URL in which the response is a redirect back to the same URL with a new cookie. The search appliance detects cyclic redirects and sets the appropriate cookie.

URL Rewrite Rules

In certain cases, you may notice URLs in the Admin Console that differ slightly from the URLs in your environment. The reason for this is that the Google Search Appliance automatically rewrites or rejects a URL if the URL matches certain patterns. The search appliance rewrites the URL for the following reasons:

- To avoid crawling duplicate content
- To avoid crawling URLs that cause a state change (such as changing or deleting a value) in the Web server
- To reject URLs that are binary files

Before rewriting a URL, the search appliance crawler attempts to match it against each of the patterns described for:

- "BroadVision Web Server" on page 53
- "Sun Java System Web Server" on page 54
- "Microsoft Commerce Server" on page 54
- "Servers that Run Java Servlet Containers" on page 54
- "Lotus Domino Enterprise Server" on page 54
- "ColdFusion Application Server" on page 56
- "Index Pages" on page 56

If the URL matches one of the patterns, it is rewritten or rejected before it is fetched.

BroadVision Web Server

In URLs for BroadVision Web server, the Google Search Appliance removes the BV_SessionID and BV_EngineID parameters before fetching URLs.

For example, before the rewrite, this is the URL:

```
http://www.broadvision.com/OneToOne/SessionMgr  
/  
home_page.jsp?BV_SessionID=NNNN0974886399.1076010447NNNN&BV_EngineID=ccceadcjdhd  
felgcefe4ecefedghdfjk.0
```

After the rewrite, this is the URL:

```
http://www.broadvision.com/OneToOne/SessionMgr/home_page.jsp
```

Sun Java System Web Server

In URLs for Sun Java System Web Server, the Google Search Appliance removes the `GXHC_qx_session_id` parameter before fetching URLs.

Microsoft Commerce Server

In URLs for Microsoft Commerce Server, the Google Search Appliance removes the `shopperID` parameter before fetching URLs.

For example, before the rewrite, this is the URL:

```
http://www.shoprogers.com/homeen.asp?shopperID=PBA1XEW6H5458NRV2VGQ909
```

After the rewrite, this is the URL:

```
http://www.shoprogers.com/homeen.asp
```

Servers that Run Java Servlet Containers

In URLs for servers that run Java servlet containers, the Google Search Appliance removes `jsessionId`, `$sessionId$`, and `$sessionId$` parameters before fetching URLs.

Lotus Domino Enterprise Server

Lotus Domino Enterprise URLs patterns are case-sensitive and are normally recognized by the presence of `.nsf` in the URL along with a well-known command such as "OpenDocument" or "ReadForm." If your Lotus Domino Enterprise URL does not match any of the cases below, then it does not trigger the rewrite or reject rules.

The Google Search Appliance rejects URL patterns that contain:

- The Collapse parameter
- SearchView, SearchSite, or SearchDomain
- The Navigate parameter and either `To=Prev` or `To=Next`
- ExpandSection or ExpandOutline parameters, unless they represent a single-section expansion
- `$OLEOBJINFO`, or `FieldElemFormat`
- CreateDocument, DeleteDocument, SaveDocument, or EditDocument

- OpenAgent, OpenHelp, OpenAbout, or OpenIcon
- ReadViewEntries

The search appliance rewrites:

- OpenDocument URLs
- URLs with the suffix #
- Multiple versions of the same URL

The following sections provide details about search appliance rewrite rules for Lotus Domino Enterprise server.

OpenDocument URLs

The Google Search Appliance rewrites OpenDocument URLs to substitute a 0 for the view name. This is a method for accessing the document regardless of view, and stops the search appliance crawler from fetching multiple views of the same document.

The syntax for this type of URL is `http://Host/Database/View/DocumentID?OpenDocument`. The search appliance rewrites this as `http://Host /Database/0/DocumentID?OpenDocument`

For example, before the rewrite, this is the URL:

```
http://www12.lotus.com/idd/doc/domino_notes/5.0.1/readme.nsf
/8d7955daacc5bdbd852567a1005ae562/c8dac6f3fef2f475852567a6005fb38f
```

After the rewrite, this is the URL:

```
http://www12.lotus.com/idd/doc/domino_notes/5.0.1/readme.nsf/0/
c8dac6f3fef2f475852567a6005fb38f?OpenDocument
```

URLs with # Suffixes

The Google Search Appliance removes suffixes that begin with # from URLs that have no parameters.

Multiple Versions of the Same URL

The Google Search Appliance converts a URL that has multiple possible representations into one standard, or canonical URL. The search appliance does this conversion so that it does not fetch multiple versions of the same URL with differing order of parameters. The search appliance's canonical URL has the following syntax for the parameters that follow the question mark:

- `?Command&Start=&Count=&Expand&...`
- `?Command&Start=&Count=&ExpandView&...`

To convert a URL to a canonical URL, the search appliance makes the following changes:

- Rewrites the "!" character that is used to mark the beginning of the parameters to "?"
- Rewrites `Expand=parameter` to `ExpandView`. If there is not a number argument to expand, it is not modified.
- Rejects URLs with more than one `Expand` parameter.
- Places parameters in the following order: `Start`, `Count`, `Expand`, followed by any other parameters.

- If the URL contains a Start parameter, but no Count parameter, adds Count=1000.
- If the URL contains Count=1000, but no Start parameter, adds Start=1.
- If the URL contains the ExpandView parameter, and has a Start parameter but no Count parameter, sets Start=1&Count=1000.
- Removes additional parameters after a command except Expand/ExpandView, Count, or Start.

For example, before the rewrite, this is the URL:

```
http://www-12.lotus.com/1dd/doc/domino_notes/5.0.1/  
readme.nsf?OpenDatabase&Count=30&Expand=3
```

After the rewrite, this is the URL:

```
http://www12.lotus.com/1dd/doc/domino_notes/5.0.1/  
readme.nsf?OpenDatabase&Start=1&Count=1000&ExpandView
```

ColdFusion Application Server

In URLs for ColdFusion application server, the Google Search Appliance removes CFID and CFTOKEN parameters before fetching URLs.

Index Pages

In URLs for index pages, the Google Search Appliance removes index.htm or index.html from the end of URLs before fetching them. It also automatically removes them from Start URLs that you enter on the **Content Sources > Web Crawl > Start and Block URLs** page in the Admin Console.

For example, before the rewrite, this is the URL:

```
http://www.google.com/index.html
```

After the rewrite, this is the URL:

```
http://www.google.com/
```


Chapter 5

Advanced Topics

Crawling is the process where the Google Search Appliance discovers enterprise content to index. The information in this chapter extends beyond basic crawl.

Identifying the User Agent

Web servers see various client applications, including Web browsers and the Google Search Appliance crawler, as “user agents.” When the search appliance crawler visits a Web server, the crawler identifies itself to the server by its User-Agent identifier, which is sent as part of the HTTP request.

The User-Agent identifier includes all of the following elements:

- A unique identifier that is assigned for each search appliance
- A user agent name
- An email address that is associated with the search appliance

User Agent Name

The default user agent name for the Google Search Appliance is “gsa-crawler.” In a Web server’s logs, the server administrator can identify each visit by the search appliance crawler to a Web server by this user agent name.

You can view or change the User-Agent name or enter additional HTTP headers for the search appliance crawler to use with the **Content Sources > Web Crawl > HTTP Headers** page in the Admin Console.

User Agent Email Address

Including an email address in the User-Agent identifier enables a webmaster to contact the Google Search Appliance administrator in case the site is adversely affected by crawling that is too rapid, or if the webmaster does not want certain pages crawled at all. The email address is a required element of the search appliance User-Agent identifier.

For complete information about the **Content Sources > Web Crawl > HTTP Headers** page, click **Admin Console Help > Content Sources > Web Crawl > HTTP Headers** in the Admin Console.

Coverage Tuning

You can control the number of URLs the search appliance crawls for a site by using the **Content Sources > Web Crawl > Coverage Tuning** page in the Admin Console. To tune crawl coverage, a URL pattern and setting the maximum number of URLs to crawl for it. The URL patterns you provide must conform to the “Rules for Valid URL Patterns” on page 87 in “Administering Crawl.”

For complete information about the **Content Sources > Web Crawl > Coverage Tuning** page, click **Admin Console Help > Content Sources > Web Crawl > Coverage Tuning** in the Admin Console.

Freshness Tuning

You can improve the performance of a continuous crawl using URL patterns on the **Content Sources > Web Crawl > Freshness Tuning** page in the Admin Console. The **Content Sources > Web Crawl > Freshness Tuning** page provides four categories of crawl behaviors, as described in the following table. To apply a crawl behavior, specify URL patterns for the behavior.

Behavior	Description
Crawl Frequently	<p>Use Crawl Frequently patterns for URLs that are dynamic and change frequently. You can use the Crawl Frequently patterns to give hints to the search appliance crawler during the early stages of crawling, before the search appliance has a history of how frequently URLs actually change.</p> <p>Any URL that matches one of the Crawl Frequently patterns is scheduled to be recrawled at least once every day. The minimum wait time (see “Wait Times” on page 50) is 15 minutes, but if you have too many URLs in Crawl Frequently patterns, wait time increases.</p>
Crawl Infrequently	<p>Use Crawl Infrequently Patterns for URLs that are relatively static and do not change frequently. Any URL that matches one of the Crawl Infrequently patterns is not crawled more than once every 90 days, regardless of its Enterprise PageRank or how frequently it changes. You can use this feature for Web pages that do not change and do not need to be recrawled. You can also use it for Web pages where a small part of their content changes frequently, but the important parts of their content does not change.</p>
Always Force Recrawl	<p>Use Always Force Recrawl patterns to prevent the search appliance from crawling a URL from cache (see “Determining Document Changes with If-Modified-Since Headers and the Content Checksum” on page 17).</p>
Recrawl these URL Patterns	<p>Use Recrawl these URL Patterns to submit a URL to be recrawled. URLs that you enter here are recrawled as soon as possible.</p>

For complete information about the **Content Sources > Web Crawl > Freshness Tuning** page, click **Admin Console Help > Content Sources > Web Crawl > Freshness Tuning** in the Admin Console.

Changing the Amount of Each Document that Is Indexed

By default, the search appliance indexes up to 2.5MB of each text or HTML document, including documents that have been truncated or converted to HTML. After indexing, the search appliance caches the indexed portion of the document and discards the rest.

You can change the default by entering a new amount of up to 10MB in **Index Limits** on the **Index > Index Settings** page.

For complete information about changing index settings on this page, click **Admin Console Help > Index > Index Settings** in the Admin Console.

Configuring Metadata Indexing

The search appliance has default settings for indexing metadata, including which metadata names are to be indexed, as well as how to handle multivalued metadata and date fields. You can customize the default settings or add an indexing configuration for a specific attribute by using the **Index > Index Settings** page. By using this page you can perform the following tasks:

- Including or excluding metadata names in dynamic navigation
- Specifying multivalued separators
- Specifying a date format for metadata date fields

For complete information about configuring metadata indexing, click **Admin Console Help > Index > Index Settings** in the Admin Console.

Including or Excluding Metadata Names

You might know which indexed metadata names you want to use in dynamic navigation. In this case, you can create a whitelist of names to be used by entering an RE2 regular expression that includes those names in **Regular Expression** and checking **Include**.

If you know which indexed metadata names you do not want to use in dynamic navigation, you can create a blacklist of names by entering an RE2 regular expression that includes those names in **Regular Expression** and selecting **Exclude**. Although blacklisted names do not appear in dynamic navigation options, these names are still indexed and can be searched by using the `inmeta`, `requiredfields`, and `partialfields` query parameters.

This option is required for dynamic navigation. For information about dynamic navigation, click **Admin Console Help > Search > Search Features > Dynamic Navigation**.

By default, the regular expression is `".*"` and **Include** is selected, that is, index all metadata names and use all the names in dynamic navigation.

For complete information about creating a whitelist or blacklist of metadata names, click **Admin Console Help > Index > Index Settings** in the Admin Console.

Specifying Multivalued Separators

A metadata attribute can have multiple values, indicated either by multiple meta tags or by multiple values within a single meta tag, as shown in the following example:

```
<meta name="authors" content="S. Jones, A. Garcia">
```

In this example, the two values (S. Jones, A. Garcia) are separated by a comma.

By using the **Multivalued Separator** options, you can specify multivalued separators for the default metadata indexing configuration or for a specific metadata name. Any string except an empty string is a valid multivalued separator. An empty string causes the multiple values to be treated as a single value.

For complete information about specifying multivalued separators, click **Admin Console Help > Index > Index Settings** in the Admin Console.

Specifying a Date Format for Metadata Date Fields

By using the **Date Format** menus, you can specify a date format for metadata date fields. The following example shows a date field:

```
<meta name="releasedOn" content="20120714">
```

To specify a date format for either the default metadata indexing configuration or for a specific metadata name, select a value from the menu.

The search appliance tries to parse dates that it discovers according to the format that you select for a specific configuration or, in case you do not add a specific configuration, the default date format. If the date that the search appliance discovers in the metadata isn't of the selected format, the search appliance determines if it can parse it as any date format.

For complete information about specifying a date format, click **Admin Console Help > Index > Index Settings** in the Admin Console.

Crawling over Proxy Servers

If you want the Google Search Appliance to crawl outside your internal network and include the crawled data in your index, use the **Content Sources > Web Crawl > Proxy Servers** page in the Admin Console. For complete information about the **Content Sources > Web Crawl > Proxy Servers** page, click **Admin Console Help > Content Sources > Web Crawl > Proxy Servers** in the Admin Console.

Preventing Crawling of Duplicate Hosts

Many organizations have mirrored servers or duplicate hosts for such purposes as production, testing, and load balancing. Mirrored servers are also the case where multiple aliases are used or a Web site has changed names, which usually occurs when companies or departments merge.

Disadvantages of allowing the Google Search Appliance to recrawl content on mirrored servers include:

- Increasing the time it takes for the search appliance to crawl content.
- Indexing the same content twice, because both versions count towards the license limit.
- Decreasing the relevance of search results, because the search appliance cannot discover accurate information about the link structure of crawled documents.

To prevent crawling of duplicate hosts, you can specify one or more “canonical,” or standard, hosts using the **Content Sources > Web Crawl > Duplicate Hosts** page.

For complete information about the **Content Sources > Web Crawl > Duplicate Hosts** page, click **Admin Console Help > Content Sources > Web Crawl > Duplicate Hosts** in the Admin Console.

Enabling Infinite Space Detection

In “infinite space,” the search appliance repeatedly crawls similar URLs with the same content while useful content goes uncrawled. For example, the search appliance might start crawling infinite space if a page that it fetches contains a link back to itself with a different URL. The search appliance keeps crawling this page because, each time, the URL contains progressively more query parameters or a longer path. When a URL is in infinite space, the search appliance does not crawl links in the content.

By enabling infinite space detection, you can prevent crawling of duplicate content to avoid infinite space indexing.

To enable infinite space detection, use the **Content Sources > Web Crawl > Duplicate Hosts** page.

For complete information about the **Content Sources > Web Crawl > Duplicate Hosts** page, click **Admin Console Help > Content Sources > Web Crawl > Duplicate Hosts** in the Admin Console.

Configuring Web Server Host Load Schedules

A Web server can handle several concurrent requests from the search appliance. The number of concurrent requests is known as the Web server’s “host load.” If the Google Search Appliance is crawling through a proxy, the host load limits the maximum number of concurrent connections that can be made through the proxy. The default number of concurrent requests is 4.0.

Increasing the host load can speed up the crawl rate, but it also puts more load on your Web servers. It is recommended that you experiment with the host load settings at off-peak time or in controlled environments so that you can monitor the effect it has on your Web servers.

To configure a **Web Server Host Load** schedule, use the **Content Sources > Web Crawl > Host Load Schedule** page. You can also use this page to configure exceptions to the web server host load.

Regarding file system crawling: if you've configured the search appliance to crawl documents from a SMB file system, it only follows the configurable default value of **Web Server Host Load** (default to 4.0), it does not follow the **Exceptions to Web Server Host Load** specifically for the SMB host. Due to design constraint, the default **Web Server Host Load** value can only be set to 8.0 or below, or it may effect the performance of your file system crawling.

For complete information about the **Content Sources > Web Crawl > Host Load Schedule** page, click **Admin Console Help > Content Sources > Web Crawl > Host Load Schedule** in the Admin Console.

Removing Documents from the Index

To remove a document from the index, add the full URL of the document to **Do Not Follow Patterns** on the **Content Sources > Web Crawl > Start and Block URLs** page in the Admin Console.

Using Collections

Collections are subsets of the index used to serve different search results to different users. For example, a collection can be organized by geography, product, job function, and so on. Collections can overlap, so one document can be relevant to several different collections, depending on its content. Collections also allow users to search targeted content more quickly and efficiently than searching the entire index.

For information about using the **Index > Collections** page to create and manage collections, click **Admin Console Help > Index > Collections** in the Admin Console.

Default Collection

During initial crawling, the Google Search Appliance establishes the default_collection, which contains all crawled content. You can redefine the default_collection but it is not advisable to do this because index diagnostics are organized by collection. Troubleshooting using the **Index > Diagnostics > Index Diagnostics** page becomes much harder if you cannot see all URLs crawled.

Changing URL Patterns in a Collection

Documents that are added to the index receive a tag for each collection whose URL patterns they match. If you change the URL patterns for a collection, the search appliance immediately starts a process that runs across all the crawled URLs and retags them according to the change in the URL patterns. This process usually completes in a few minutes but can take up to an hour for heavily-loaded appliances. Search results for the collection are corrected after the process finishes.

JavaScript Crawling

The search appliance supports JavaScript crawling and can detect links and content generated dynamically through JavaScript execution. The search appliance supports dynamic link and content detection in the following situations:

- “Logical Redirects by Assignments to `window.location`” on page 63
- “Links and Content Added by `document.write` and `document.writeln` Functions” on page 63
- “Links that are Generated by Event Handlers” on page 64
- “Links that are JavaScript Pseudo-URLs” on page 64
- “Links with an onclick Return Value” on page 65

If your enterprise content relies on URLs generated by JavaScript that is not covered by any of these situations, use jump pages or basic HTML site maps to force crawling of such URLs in JavaScript.

The search appliance only executes scripts embedded inside a document. The search appliance does not support:

- DOM tracking to support calls, such as `document.getElementById`
- External scripts execution
- AJAX execution

Also, if the search appliance finds an error while parsing JavaScript, or if the JavaScript contains an error, the search appliance might fail to find links that require functions below the error. In this instance, anything below the error might be discarded.

Logical Redirects by Assignments to `window.location`

The search appliance crawls links specified by a logical redirect by assignment to `window.location`, which makes the web browser load a new document by using a specific URL.

The following code example shows a logical redirect by assignment to `window.location`.

```
<HTML>
  <HEAD>
    <SCRIPT type='text/javascript'>
      var hostName = window.location.hostname;
      var u = "http://" + hostName + "/links" + "/link1.html";
      window.location.replace(u);
    </SCRIPT>
  </HEAD>
  <BODY></BODY>
</HTML>
```

Links and Content Added by `document.write` and `document.writeln` Functions

The search appliance crawls links and indexes content that is added to a document by `document.write` and `document.writeln` functions. These functions generate document content while the document is being parsed by the browser.

The following code example shows links added to a document by `document.write`.

```
<HTML>
  <HEAD>
    <SCRIPT type='text/javascript'>
      document.write('<a href="http://foo.google.com/links/'
        + 'link2.html">link2</a>');
      document.write(
        '<script>document.write('<a href="http://foo.google.com/links/'
          + 'link3.html">script within a script</a>\') ;</script>');
    </SCRIPT>
  </HEAD>
</BODY></BODY>
</HTML>
```

Links that are Generated by Event Handlers

The search appliance crawls links that are generated by event handlers, such as `onclick` and `onsubmit`.

The following code example shows links generated by event handlers in an anchor and a `div` tag.

```
<HTML>
  <HEAD>
    <SCRIPT type='text/javascript'>
      function openlink(id) {
        window.location.href = "/links/link" + id + ".html";
      }
    </SCRIPT>
  </HEAD>
  <BODY>
    <a onclick="openlink('4');" href="#">attribute anchor 1</a>
    <div onclick="openlink('5');">attribute anchor 2</div>
  </BODY>
</HTML>
```

Links that are JavaScript Pseudo-URLs

The search appliance crawls links that include JavaScript code and use the `javascript:` pseudoprotocol specifier.

The following code example shows a link that is JavaScript pseudo-URL.

```
<HTML>
  <HEAD>
    <SCRIPT type='text/javascript'>
      function openlink(id) {
        window.location.href = "/links/link" + id + ".html";
      }
    </SCRIPT>
  </HEAD>
  <BODY>
    <a href="javascript:openlink('6')">JavaScript URL</a>
  </BODY>
</HTML>
```


Links with an onclick Return Value

The search appliance crawls links with an onclick return value other than false. If onclick script returns false, then the URL will not be crawled. The following code example shows both situations.

```
<HTML>
  <HEAD></HEAD>
  <BODY>
    <a href="http://bad.com" onclick="return false;">This link will not be
crawled</a>
    <a href="http://good.com" onclick="return true;">This link will be crawled</a>
  </BODY>
</HTML>
```

Indexing Content Added by document.write/writeln Calls

Any content added to the document by `document.write/writeln` calls (as shown in the following example) will be indexed as a part of the original document.

```
<HTML>
  <HEAD>
    <SCRIPT type='text/javascript'>
      document.write('<P>This text will be indexed.</P>');
    </SCRIPT>
  </HEAD>
  <BODY></BODY>
</HTML>
```

Discovering and Indexing Entities

Entity recognition is a feature that enables the Google Search Appliance to discover interesting entities in documents with missing or poor metadata and store these entities in the search index.

For example, suppose that your search appliance crawls and indexes multiple content sources, but only one of these sources has robust metadata. By using entity recognition, you can enrich the metadata-poor content sources with discovered entities and discover new, interesting entities in the source with robust metadata.

After you configure and enable entity recognition, the search appliance automatically discovers specific entities in your content sources during indexing, annotates them, and stores them in the index. Once the entities are indexed, you can enhance keyword search by adding the entities in dynamic navigation, which uses metadata in documents and entities discovered by entity recognition to enable users to browse search results by using specific attributes. To add the entities to dynamic navigation, use the **Search > Search Features > Dynamic Navigation** page.

Additionally, by default, entity recognition extracts and stores full URLs in the index. This includes both document URLs and plain text URLs that appear in documents. So you can match specific URLs with entity recognition and add them to dynamic navigation, enabling users to browse search results by full or partial URL. For details about this scenario, see "Use Case: Matching URLs for Dynamic Navigation" on page 66.

The **Index > Entity Recognition** page enables you to specify the entities that you want the search appliance to discover in your documents. If you want to identify terms that should not be stored in the index, you can upload the terms in an entity blacklist file.

Creating Dictionaries and Composite Entities

Before you can specify entities on the **Index > Entity Recognition** page, you must define each entity by creating dictionaries of terms and regular expressions. Dictionaries for terms are required for entity recognition. Dictionaries enable entity recognition to annotate entities, that is, to discover specific entities in the content and annotate them as entities.

Generally, with dictionaries, you define an entity with lists of terms and regular expressions. For example, the entity "Capital" might be defined by a dictionary that contains a list of country capitals: Abu Dhabi, Abuja, Accra, Addis Ababa, and so on. After you create a dictionary, you can upload it to the search appliance.

Entity recognition accepts dictionaries in either TXT or XML format.

Optionally, you can also create composite entities that run on the annotated terms. Like dictionaries, composite entities define entities, but composite entities enable the search appliance to discover more complex terms. In a composite entity, you can define an entity with a sequence of terms. Because composite entities run on annotated terms, all the words in a sequence must be tagged with an entity and so depend on dictionaries.

For example, suppose that you want to define a composite entity that detects full names, that is, combinations of titles, names, middlenames, and surnames. First, you need to define four dictionary-based entities, Title, Name, Middlename, and Surname, and provide a dictionary for each one. Then you define the composite entity, FullName, which detects full names.

A composite entity is written as an LL1 grammar.

The search appliance provides sample dictionaries and composite entities, as shown on the **Index > Entity Recognition** page.

Setting Up Entity Recognition

Google recommends that you perform these tasks for setting up entity recognition, in the following order:

1. Creating dictionaries and, optionally, composite entities.
2. Adding new entities by adding dictionaries and, optionally, composite entities.
3. Enabling entity recognition.

For complete information about setting up entity recognition, click **Admin Console Help > Index > Entity Recognition** in the Admin Console.

Use Case: Matching URLs for Dynamic Navigation

This use case describes matching URLs with entity recognition and using them to enrich dynamic navigation options. It also shows you how to define the name of the dynamic navigation options that display, either by explicitly specifying the name or by capturing the name from the URL.

This use case assumes you have already enabled entity recognition on your GSA and added entities to dynamic navigation. Having seen how easy this feature makes browsing results, your users also want to be able to browse by URLs. These URLs include:

```
http://www.mycompany.com/services/...  
http://www.mycompany.com/policies/...  
http://www.mycompany.com/history/...
```

They want dynamic navigation results to include just the domains "services," "policies," and so on. You can achieve this goal by performing the following steps:

1. Creating an XML dictionary that defines the entity
2. Adding the entity and dictionary to entity recognition
3. Adding the entity to dynamic navigation

Creating an XML Dictionary that Defines an Entity for Matching URLs

The following example shows an XML dictionary for entity recognition that matches URLs. In this example, the names displayed for the dynamic navigation options are defined using the name element:

```
<?xml version="1.0"?>
<instances>
  <instance>
    <name>services</name>
    <pattern>http://.*/services.*</pattern>
    <store_regex_or_name>name</store_regex_or_name>
  </instance>
  <instance>
    <name>policies</name>
    <pattern>http://.*/policies/*</pattern>
    <store_regex_or_name>name</store_regex_or_name>
  </instance>
  <instance>
    <name>history</name>
    <pattern>http://.*/history/*</pattern>
    <store_regex_or_name>name</store_regex_or_name>
  </instance>
</instances>
```

Note: You must create an instance for each type of URL that you want to match.

Creating an XML Dictionary that Defines an Entity for Capturing the Name from the URL

The following example shows an XML dictionary that matches URLs and captures the name of the dynamic navigation options by using the group term in the regular expression pattern:

```
<?xml version="1.0"?>
<instances>
  <instance>
    <name> Anything - will not be used </name>
    <pattern> http://www.mycompany.com/(\\w+)/[^\\s]+ </pattern>
    <store_regex_or_name> regex_tagged_as_first_group </store_regex_or_name>
  </instance>
</instances>
```

There are two important things to note about this example:

- The regular expression has a (\\w+) term. The term is in parenthesis, which defines a capturing group. The \\w means that this expression will capture any word characters (? [0-9A-Za-z_]).
- The <store_regex_or_name> is set to regex_tagged_as_first_group. This indicates that if the pattern has a match, the text matched by the capturing group will be used as the name for the entity.

Adding the Entity to Entity Recognition

Add a new entity, which is defined by the dictionary:

1. Click **Index > Entity Recognition > Simple Entities**.
2. On the **Simple Entities** tab, enter the name of the entity in the **Entity name** field, for example "type-of-doc."
3. Click **Choose File** to navigate to the dictionary file in its location and select it.
4. Under **Case sensitive?**, click **Yes**.
5. Under **Transient?**, click **No**.
6. Click **Upload**.
7. (Optional) Click **Entity Diagnostics** to test that everything works.

Adding the Entity to Dynamic Navigation

To show URLs as dynamic navigation options, add the entity:

1. Click **Search > Search Features > Dynamic Navigation**.
2. Under **Existing Configurations**, click **Add**.
3. In the **Name** box, type a name for the new configuration, for example "domains."
4. Under **Attributes**, click **Add Entity**.
5. In the **Display Label** box, enter the name you want to appear in the search results, for example "TypeOfUrl." This name can be different from the name of the entity.
6. From the **Attribute Name** drop-down menu, select the name of the entity that you created, for example "type-of-doc."
7. From the **Type** drop-down menu, select STRING.
8. Select options for sorting entities in the dynamic navigation panel.
9. Click **OK**.

Viewing URLs in the Search Results

After you perform the steps described in the preceding sections, your users will be able to view URLs in the dynamic navigation options, as shown in the following figure.



Navigate	All results
TypeOfUrl	Content of the service page 3
history (3)	Content of the service page 3
policies (3)	- 1k - 2013-02-14 - Cached
services (3)	Content of the policy page 3
	Content of the policy page 3
	- 1k - 2013-02-14 - Cached
	Content of the service page 2
	Content of the service page 2
	- 1k - 2013-02-14 - Cached
	Content of the policy page 2
	Content of the policy page 2
	- 1k - 2013-02-14 - Cached
	Content of the history page 1
	Content of the history page 1
	- 1k - 2013-02-15 - Cached
	Content of the service page 1
	Content of the service page 1
	- 1k - 2013-02-14 - Cached
	Content of the history page 3
	Content of the history page 3
	- 1k - 2013-02-15 - Cached
	Content of the policy page 1
	Content of the policy page 1
	- 1k - 2013-02-14 - Cached
	Content of the history page 2
	Content of the history page 2
	- 1k - 2013-02-15 - Cached

Note that dynamic navigation only displays the entities of the documents in the result set (the first 30K documents). If documents that contain entities are not in the result set, their entities are not displayed.

However, take note that entity recognition only runs on documents that are added to the index after you enable entity recognition. Documents already in the index are not affected. To run entity recognition on documents already in the index, force the search appliance to recrawl URL patterns by using the **Index > Diagnostics > Index Diagnostics** page.

Use Case: Testing Entity Recognition for Non-HTML Documents

This use case describes how you can test your entity recognition configuration on an indexed document that is not in HTML format. To run entity diagnostics on HTML documents, use the **Index > Entity Recognition > Entity Diagnostics** page in the Admin Console.

Testing Entity Recognition on a Cached Non-HTML Document

Note: This procedure does not affect the crawl and indexing of a URL

1. Click **Index > Diagnostics > Index Diagnostics**.
2. Click **List format**.
3. Under **All hosts**, click the URL of the document that you want to test.

4. Under **More information about this page**, click **Open in entity diagnostics**, as shown below.

Search Appliance Index > Diagnostics > Index Diagnostics

Content Sources

Index

Index Settings

Document Dates

Entity Recognition

Alerts

Collections

Corpus Phrases

Composite Collections

Diagnostics

Index Diagnostics

Content Statistics

Export URLs

Reset Index

Search

Reports

GSA Unification

GSA'n

Administration

Index Diagnostics (Help)

Specify the URLs for which you want diagnostics:

URLs starting with:

URL Status: Include Exclude

All hosts > http://pcaorios2.mtv.corp.google.com/features/doc_types/USA.pdf

More information about this page

- [URI After Redirects: http://pcaorios2.mtv.corp.google.com/features/doc_types/USA.pdf](#)
- [Link to this page](#)
- [Cached version \(Open in entity diagnostics \)](#)
- [PageRank:](#)
- [Last Modified: 04 Mar 2014](#)
- [Authentication Method at Crawl Time: None](#)
- [Security at Serve Time: Public](#)
- [Number of links on this page to crawled pages: 0](#)
- [View list of public crawled pages that link to this page](#)
- [View list of all crawled pages that link to this page](#)
- [Crawl Frequency: normal](#)
- [Download time in ms: 126](#)
- [Content Type: text/pdf](#)
 - [Preview Status: READY](#)
- [Content Size: 16643](#)
- [Language: ENGLISH](#)
- [Encoding: UTF8](#)
- [Currently Inflight: no](#)
- [This page is in the following collections:](#)
 - [Default](#)
 - [default_collection](#)

Note: The **Open in entity diagnostics** link is only available for public documents.

Entity diagnostics runs on the cached version of the document and displays the entities found in the document, as shown below.

Search Appliance Index > Entity Recognition > Entity Diagnostics

Simple Entities Composite Entities Blacklist **Entity Diagnostics** Adjustments

Entity Diagnostics (Help)

This page allows checking the entities that will be extracted from a given document. The URL or local document provided must be an HTML file.

Recognize entities for URL:

Recognize entities for local html file (< 1MB): No file chosen

Entities recognized in the text

Below, the entities discovered and stored for the uploaded document are colored in the text. In addition, entities discovered but not stored since they are transient are highlighted in grey. Note that this does not report the entities discovered in the document URL since it is not displayed. A full list of entities is reported in the right-hand table.

Page 1

United States - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/United_States
 [en]

United States of America

Flag
 Great Seal

Motto:
 "In God we trust" (official)[[2]]
 "E pluribus unum" (Latin) (traditional)
 "Out of many, one"

Anthem: "The StarSpangled Banner"

Capital
Washington, D.C.
 [37°N, 77°W]

Largest city
New York City

Official languages
 None at federal level[a]

National language
 English[b]

Demonym
 American

Government
 Federal presidential
 constitutional republic
 President

Entity name	Entity value
Country	United States
Country	Mexico
Country	United States of America
Date	3/3/2014
City	Washington, DC
City	New York City

Wildcard Indexing

Wildcard search enables your users to enter queries that contain substitution patterns rather than exact spellings of terms. Wildcard *indexing* makes words in your content available for wildcard search.

To disable or enable wildcard indexing or the change the type of wildcard indexing, use the **Index > Index Settings** page in the Admin Console. For more information about wildcard indexing, click **Admin Console Help > Index > Index Settings**.

By default, wildcard search is enabled for each front end of the search appliance. You can disable or enable wildcard search for one or more front ends by using the **Filters** tab of the **Search > Search Features > Front Ends** page. Take note that wildcard search is not supported with Chinese, Japanese, Korean, or Thai. For more information about wildcard search, click **Admin Console Help > Search > Search Features > Front Ends > Filters**.

Chapter 6

Database Crawling and Serving

Crawling is the process where the Google Search Appliance discovers enterprise content to index. This chapter tells search appliance administrators how to configure database crawling and serving.

Database Crawler Deprecation Notice

In GSA release 7.4, the on-board database crawler is deprecated. It will be removed in a future release. If you have configured on-board database crawling for your GSA, install and configure the Google Connector for Databases 4.0.4 or later instead. For more information, see “Deploying the Connector for Databases,” available from the [Connector Documentation page](#).

Introduction

This chapter describes how the Google Search Appliance crawls databases and serves results from them. This document is intended for search appliance administrators who need to understand:

- How to configure a database crawl
- How to start a database crawl
- How to troubleshoot a database crawl

The following table lists the major sections in this chapter.

Section	Describes
"Supported Databases" on page 73	The relational databases that the Google Search Appliance can crawl
"Overview of Database Crawling and Serving" on page 73	How the Google Search Appliance crawls a database, uploads database content for inclusion in the index, and serves and displays search results from a database
"Configuring Database Crawling and Serving" on page 76	How to configure a Google Search Appliance for database crawling and serving
"Starting Database Synchronization" on page 82	How to start the process of crawling a database and uploading content for inclusion in the search index
"Troubleshooting" on page 82	How to troubleshoot a database crawl
"Frequently Asked Questions" on page 83	Questions about database crawling and serving

Supported Databases

The Google Search Appliance provides access to data stored in relational databases by crawling the content directly from the database and serving the content. To access content in a database, the Google Search Appliance sends SQL (Structured Query Language) queries using JDBC (Java Database Connectivity) adapters provided by each database company.

The following table lists databases and JDBC adapter versions that the Google Search Appliance supports.

Database	JDBC Adapters
DB2®	IBM®DB2 Universal Database (UDB) 8.1.0.64
MySQL®	MySQL Connector/J 3.1.13
Oracle®	Oracle Database 10g Release 2, 10.1.0.2.0 driver
SQL Server™	Microsoft® SQL Server™ 2008 JDBC™ Driver 2.0
Sybase®	jConnect™ for JDBC™ 5.5 Build 25137

Overview of Database Crawling and Serving

The Google Search Appliance has two built-in components for crawling and serving content from databases:

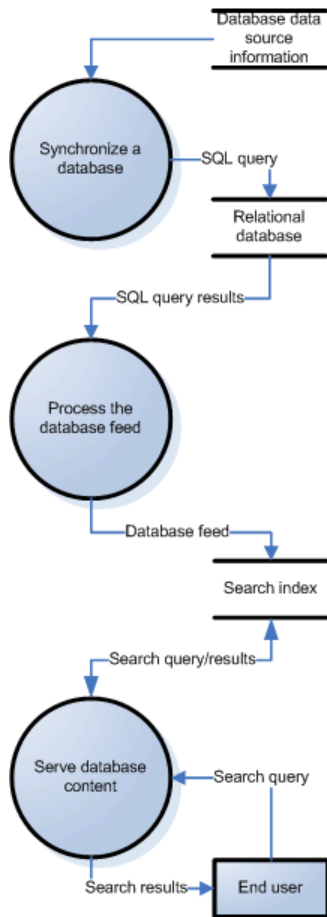
- **TableCrawler**—A custom connector used for pushing records from a database into the appliance's index using feeds.
- **TableServer**—A component used for serving search results.

TableServer connects to the database when a serve query (see "Serve Queries" on page 77) is defined and the user clicks on a search result from the database.

The following diagram provides an overview of the major database crawl and serve processes:

- “Synchronizing a Database” on page 74
- “Processing a Database Feed” on page 75
- “Serving Database Content” on page 75

See the following sections in this document for descriptions of each major process. For an explanation of the symbols used in the diagram, refer to “About the Diagrams in this Section” on page 13.



Synchronizing a Database

The process of crawling a database is called synchronizing a database. Full database synchronizations are always manual and you must start one by using the **Content Sources > Databases** page. The Google Search Appliance does not currently support scheduled database synchronization.

After you start database synchronization (see “Starting Database Synchronization” on page 82), the TableCrawler and TableServer use JDBC adapters to connect to the specified database. They connect by using information that you provide when you configure database crawling and serving (see “Configuring Database Crawling and Serving” on page 76).

When you start database synchronization, the TableCrawler component of the Google Search Appliance performs the following steps:

1. Connects to a relational database.
2. Crawls the contents of the database.
3. Pushes records from a database into the appliance's index using feeds.

Specifically, the TableCrawler sends the database the SQL query that you entered in the **Crawl Query** field, using the JDBC database client libraries.

The results are wrapped in Feed XML (eXtensible Markup Language) syntax (see “The Feed XML and Stylesheet” on page 78), and include a record for each row of the database crawl query results. This database feed file is presented to the Feeder system as soon as the crawl is complete.

To prevent the Google Search Appliance from deleting database content from the index when its PageRank™ is low and the index has reached its license limit, click the **Lock documents** checkbox under **Create** (see “Providing Database Data Source Information” on page 76).

Processing a Database Feed

Once synchronization is complete, the database feed is automatically uploaded to the search appliance for inclusion in the index. The crawl query (see “Crawl Queries” on page 77) is used to produce a feed description. All feeds, including database feeds, share the same namespace. Database source names should not match existing feed names.

Unique Generated URLs

The database synchronization process generates the URL attribute. Note that the IP address of the Google Search Appliance is used rather than the name in the URL.

Pages created from the database being indexed all have the form shown in the following example.

```
googledb://<database-hostname>/<DB_SOURCE_NAME>/
```

Therefore, you need to enter this pattern in **Follow Patterns** on the **Content Sources > Web Crawl > Start and Block URLs** page in the Admin Console. For more details see Setting the URL Patterns to Enable Database Crawl.

The Feeds connector generates a URL from either the Primary Key columns or from the URL column as specified in **Serve URL** field on the **Content Sources > Databases** page. A unique hash value is generated from the primary key to form part of the URL.

These generated URLs have the form shown in the following example.

```
http://<appliance_hostname>/db/<database-hostname>/<DB_SOURCE_NAME>/  
<result_of_hash>
```

Serving Database Content

Once the search appliance index has been updated with the Feed XML file, the content is available for serving in approximately 30 minutes. The following sections describe how the Google Search Appliance performs the following actions:

- Generating search results for database content (see “Generating Search Results” on page 76)
- Displaying search results for database content (see “Displaying Search Results” on page 76)

Generating Search Results

The linked content from search results is generated by the serve query at serve time. If a user's query returns results that were originally retrieved from a database query, each result links to content that is queried from the database at serve time. The associated snippets are generated from the index.

If the database has changed since the last database synchronization, the resulting page may not relate to the original user search.

If the database serve query has changed since the database was last synchronized, the search results may produce pages that do not match the user's query.

Displaying Search Results

The TableServer displays search results for a query from the index. The search is made over the indexed content and shows URLs and snippet information from the crawl query results. The result links direct to URLs that are either:

- Generated on the search appliance (when **Serve Query** is used). The URLs are rewritten by the XSLT to be served through the search appliance by using the following format: `http://<appliance_hostname>/db/<database-hostname>/<DB_SOURCE_NAME>/<result_of_hash>`

or

- Obtained from the database (when **Serve URL Field** is used).

The TableServer is not used if **Serve URL Field** is selected. **Serve URL Field** indicates the column in the database that contains a URL to display for each row, when the user clicks on the result of a search.

When **Serve URL Field** is selected, the database stylesheet is only used to format the database data for indexing and for the snippets. It is not used by the referenced URL from **Serve URL Field**.

Configuring Database Crawling and Serving

Before you can start database synchronization (see "Starting Database Synchronization" on page 82), you must configure database crawling and serving by performing the following tasks:

- "Providing Database Data Source Information" on page 76
- "Setting URL Patterns to Enable Database Crawl" on page 81

Providing Database Data Source Information

This information enables the crawler to access content stored in the database and to format search results. Database data source information includes the following items:

- Source name—Name of the data source
- Database type—Choose from IBM DB2, Oracle, MySQL, MySQL Server, or Sybase
- Hostname—Name of the database server in fully-qualified domain name format (for example, db.mycompany.com) resolvable by the DNS server on the appliance (an IP address can also be used)
- Port—The port number that is open to the database that JDBC adapter should connect to

- Database Name—The name given to the database
- Username—User name to access the database
- Password—Password for the database
- Lock documents—If selected, documents won't be removed from index when license limit is reached. This is equivalent to using "lock" attribute on a feed record.
- Crawl query (see "Crawl Queries" on page 77)—A SQL query for the database that returns all rows to be indexed
- Serve query (see "Serve Queries" on page 77)—A SQL statement that returns a row from a table or joined tables which matches a search query.
- Data Display/Usage (see "The Feed XML and Stylesheet" on page 78)—The stylesheet used to format the content of the Feed XML document
- Advanced Settings (see "Configuring Advanced Settings" on page 80)—Incremental crawl query, BLOB fields

You provide database data source information by using the **Create New Database Source** section on the **Content Sources > Databases** page in the Admin Console. To navigate to this page, click **Content Sources > Databases**.

For complete information about the **Create New Database Source** section, click **Admin Console Help > Content Sources > Databases** in the Admin Console.

Crawl Queries

An SQL crawl query must be in the form shown in the following example.

```
SELECT <table.column> [, <table.column>, ...]
FROM <table> [, <table>, ...
[WHERE some_condition_or_join]
```

Each row result corresponds to a separate document. The information retrieved from the crawl query provides the data for the indexing.

Serve Queries

An SQL serve query is used when a user clicks on a search result link, to retrieve and display the desired document data from the database.

A serve query displays result data using the '?' in the `WHERE` clause to allow for particular row selection and display. The Primary Key Fields must provide the column names for the field to substitute with the '?'.

Primary Key Fields are column heading names (separated by commas) such as Last_Name, First_Name, SSN (Social Security Number), Birth_Date. The Primary Key field must provide a unique identifier for a database query result. This may be a combination of column names which produce a unique permutation from the corresponding values. The Primary Key allows each result row from a database query to be reliably identified by the serve query. Primary keys must be listed in exactly the same order as they appear in the `WHERE` clause.

Crawl and Serve Query Examples

This section shows example crawl and serve queries for an employee database with these fields:

```
employee_id, first_name, last_name, email, dept
```

The following example shows the crawl query.

```
SELECT employee_id, first_name, last_name, email, dept
FROM employee
```

The following example shows the serve query.

```
SELECT employee_id, first_name, last_name, email, dept
FROM employee
WHERE employee_id = ?
```

The Primary Key field for this case must be `employee_id`. The '?' signifies that this value is provided at serve time, from the search result that the user clicks.

For a table with multiple column Primary Keys, if the combination of `employee_id`, `dept` is unique, you can use multiple bind variables. The crawl query for this example is the same as shown in this section. The following example shows the serve query.

```
SELECT employee_id, first_name, last_name, email, dept
FROM employee
WHERE employee_id = ? AND dept = ?
```

Note:

- SQL keywords are in uppercase by convention. Uppercase is not required.
- The '?' is substituted with a real column value to identify a particular record to be displayed when a user clicks on a database search result.
- The URL accessed by the user is partly generated from the Primary Keys; the database query is made based on the serve query and the substituted Primary Key values. The possible values for this column are obtained from those listed in the results from the crawl query.
- The column names specified in the `WHERE` clause must be included in the same order in the Primary Key Fields.

The Feed XML and Stylesheet

You specify a stylesheet for formatting the content of the Feed XML document by using the stylesheet specified in the **Data Display/Usage** section of the **Content Sources > Databases** page in the Admin Console. This stylesheet defines the formatting used between each record.

You can use the default database stylesheet, or upload your own. To view the default database stylesheet, `dbdefault.xml`, download it from this link: <https://support.google.com/gsa/answer/6069358>. You can make changes to it, and then upload it by using the **Upload stylesheet** selection on the **Content Sources > Databases** page.

Each row returned by the database is represented by a unique URL in the appliance index.

The following example shows internally stored output of a database sync, using the default database stylesheet. In this example, the database hostname is mydb.mycompany.com and the database source name is DB_SOURCE_NAME.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE gsafeed SYSTEM "http://ent1:7800/gsafeed.dtd">
<gsafeed>
<header>
<datasource>DB_SOURCE_NAME</datasource>
<feedtype>full</feedtype>
</header>
<group>
<record url="googledb://mydb.mycompany.com/DB_SOURCE_NAME/
azE9MSwwOTk4T0U3NTAwNisrKysrKysrKyZrMj0yLDA"
action="add" mimetype="text/html" lock="true">
<content><![CDATA[<html>
<head>
<META http-equiv="Content-Type" content="text/html;
charset=UTF-8">
<title>Default Page</title>
</head>
<body style="font-family: arial, helvetica, sans-serif;">
<H2 align="center">Database Result</H2>
<table cellpadding="3" cellspacing="0" align="center"
width="100%" border="0">
<tr>
<th style="border-bottom: #ffcc00 1px solid;" align="left">
DOC_ID_COLUMN
</th>
<th style="border-bottom: #ffcc00 1px solid;" align="left">
SECOND_COLUMN_NAME
</th>
<th style="border-bottom: #ffcc00 1px solid;" align="left">
THIRD_COLUMN_NAME
</th>
</tr>
<tr valign="top" bgcolor="white">
<td><font size="-1">2</font></td>
<td><font size="-1">Second column content data.</font></td>
<td><font size="-1">Third column content data.</font></td>
</tr>
</table>
</body>
</html>
]]></content>
</record>
</group>
</gsafeed>
```

Adding Metadata to Database Content

The following example shows how to add metadata to indexed content by customizing the <head> section in the default database stylesheet. In this example, name and content metadata is added:

```
<head>
<title>Default Page</title>
<xsl:for-each select="/database/table/table_rec/*">
<meta>
<xsl:attribute name="name">
<xsl:value-of select="name(.)"/>
</xsl:attribute>
<xsl:attribute name="content">
<xsl:value-of select="."/>
</xsl:attribute>
</meta>
</xsl:for-each>
</head>
```

Configuring Advanced Settings

The Advanced Settings section of the Database Datasource Information contains options for configuring an incremental crawl query and BLOB type and content. The following table describes the advanced settings.

Incremental Crawl Query	<p>Provides a means for the appliance to update the index of database data, without having to retrieve the entire contents of an unconstrained query.</p> <p>The Incremental Crawl Query requires a modified version of the Crawl Query. It must include a last_modified_time condition of the following form:</p> <pre>SELECT ... WHERE last_modified_time > ?</pre> <p>The ? will hold the last successful crawl time from the appliance. If you do not use the ? character, the query will fail.</p> <p>One of the joined tables must have a modification time column. The time format used for modification times is</p> <p>YYYY-MM-DD HH:MM:SS and will be in GMT. Also the column must have a date data type.</p> <p>Incremental feeds and full feeds allow for deletion and addition of data. These take the following form:</p> <pre>SELECT ..., action WHERE last_modification_time > ?</pre>
Action Field	<p>The Action column must specify either "add" or "delete".</p> <p>The database administrator should populate the 'action' column using database triggers. The 'action' column need not be part of the source table, but instead part of a separate logging table which is joined with the source table holding the content by means of primary keys. The database administrator should purge the logging information of all entries dated before the last successful incremental crawl.</p>

BLOB MIME Type Field	The name of the column that contains the standard Internet MIME type values of Binary Large Objects, such as text/plain and text/html.
	Database feeds do support content in BLOB columns. The MIME type information must be supplied as a column. BLOBs use Base64 binary encoding. The XSL transformation from the specified stylesheet is not applied to BLOB data, or its associated row.
	BLOBs are automatically binary encoded as Base64 when it is crawled by the TableCrawler. BLOBs will display HTML snippets but their links will be to the original binary format (e.g. MS Word, PDF). The cache link for the snippet will provide an HTML representation of the binary data.
	Multiple BLOBs in a single query are not supported. A CLOB can be treated as a BLOB column or as text.
	The search appliance usually transforms data from crawled pages, which protects against security vulnerabilities. If you cause the search appliance to crawl BLOB content by filling in these advanced settings, certain conditions could open a vulnerability. The vulnerability exists only if both of these conditions are true: <ul style="list-style-type: none"> <li data-bbox="440 741 1013 768">• A perpetrator has access to the database table. <li data-bbox="440 795 1317 852">• You are using secure search, which causes the search appliance to request usernames and passwords or other credentials.
BLOB Content Field	The name of the column that contains the BLOB content, of type described in the BLOB MIME Type Field.

Setting URL Patterns to Enable Database Crawl

When you set up a database crawl you need to include entries in the **Follow Patterns** fields on the **Content Sources > Web Crawl > Start and Block URLs** page of the Admin Console.

To include all database feeds, use the following crawl pattern:

```
^googledb://
```

To include a specific database feed, use the following crawl pattern:

```
^googledb://<database_host_name>/<database_source_name>/
```

URLs and URL patterns are case sensitive. If you use uppercase for the database source name, you must use the same uppercase in the crawl start URLs and crawl patterns.

If your data source contains a URL column with URLs that point to your own website, add those URL patterns under **Follow Patterns** on the **Content Sources > Web Crawl > Start and Block URLs** page.

For complete information about the **Content Sources > Web Crawl > Start and Block URLs** page, click **Admin Console Help > Content Sources > Web Crawl > Start and Block URLs** in the Admin Console.

For more information about URL patterns, see “Constructing URL Patterns” on page 86.

Starting Database Synchronization

After you configure database crawling and serving (see “Configuring Database Crawling and Serving” on page 76), you can start synchronizing a database by using the **Content Sources > Databases** page in the Admin Console.

To synchronize a database:

1. Click **Content Sources > Databases**.
2. In the **Current Databases** section of the page, click the **Sync** link next to the database that you want to synchronize.

The database synchronization runs until it is complete.

After you click **Sync**, the link label changes to **Sync'ing**, which indicates that the database crawl is in process. When the crawl completes, **Sync'ing** no longer appears. However, to see the updated status for a database synchronization, you must refresh the **Content Sources > Databases** page. To refresh the page, navigate to another page in the Admin Console then navigate back to the **Content Sources > Databases** page.

You can synchronize several databases at the same time. The resulting feeds are also processed concurrently. For any given database, the search appliance must finish synchronization before it can begin to process the database feed.

You can also use the **Current Databases** section of the **Content Sources > Databases** page to edit database data source information for a database, delete a database, or view log information for a database.

For complete information about the **Current Databases** section, click **Admin Console Help > Content Sources > Databases** in the Admin Console.

Monitoring a Feed

You can monitor the progress of a database feed by using the **Content Sources > Feeds** page in the Admin Console. This page shows all feed status, including the automatically-entered database feed. When a feed process successfully finishes, it is marked **completed** on the page.

For complete information about the **Content Sources > Feeds** page, click **Admin Console Help > Content Sources > Feeds** in the Admin Console.

For more information about feeds, refer to the *Feeds Protocol Developer's Guide*.

Troubleshooting

This section contains recommended actions for troubleshooting problems when database synchronization does not appear to result in a feed.

Verify the hostname and port

Verify that a database process is listening on the host and port that has been specified on the **Content Sources > Databases** page. For example, the default port for MySQL is 3306. Run the following command and verify that you get a connection. Be sure to test this from a computer on the same subnet as the appliance.

```
telnet <dbhostname> 3306
```

Verify the remote database connection

Check that you are able to connect to the database from a remote computer using the connection parameters. For example, run the following command line to connect to a remote MySQL database:

```
mysql -u<username> -p<password> -h<dbhostname> <database>
```

In this example, the MySQL client must be available on the computer used.

Check the database logs

Look at the logs on the database server to see whether there was a successful connection.

Check for a database URL in the Follow and Crawl Patterns

Ensure that a suitable follow and crawl URL pattern for the database is entered on the **Content Sources > Web Crawl > Start and Block URLs** page in the Admin Console. If there is no suitable pattern:

1. Enter the following pattern in **Follow Patterns**:

```
^googledb://<appliance_ip_address>/db/
```

2. Click the **Save** button to save your changes.

Check the Serve Query or Serve URL

On the **Content Sources > Databases** page, make sure there is either a valid entry in either **Serve Query** or **Serve URL Field**.

Check the SQL query

Make sure the SQL queries are valid given primary key substitutions for the '?' value. Test the query by using the SQL client for the database being used.

Verify a known-working crawl query

Verify that a known-working crawl query, which produces a small number of results, works correctly.

Check database networking

Run a `tcpdump` on the traffic between the appliance and the database server on the specified port when you do a `sync`. Compare to a `tcpdump` from a successful connection.

Check the feed

If the `sync` has completed successfully, troubleshoot the feed. For information about troubleshooting feeds, refer to the *Feeds Protocol Developer's Guide*.

Frequently Asked Questions

Q: Can multiple databases be synchronized at one time?

A: Yes.

Q: What would happen if a new database synchronization were started and completed while a previous one is still being processed by the feeder? Is the feeder restarted with the new feed or the new feed just queued up behind the one running?

A: Database feeds are processed in queued order. If a feed is being processed, a new feed of the same description is queued up behind the one running.

Q: When will the database be resynchronized?

A: A database is only synchronized fully when you click the **Sync** link on the **Content Sources > Databases** page in the Admin Console. Incremental synchronizations are a more efficient mechanism for updating the index for large query result sets which only change partially. Neither incremental nor full database syncs are scheduled automatically.

Q: Is there a way to schedule a database crawl?

A: There is currently no way to schedule a database crawl. It can be synchronized manually from the **Content Sources > Databases** page.

Q: Can a sync be stopped once it's in progress?

A: There is currently no way to stop a database synchronization once it is started.

Q: How can I tell the status of the sync?

A: On the **Content Sources > Databases** page, the **Sync** link under Current Databases reads **Sync'ing** when the link is clicked. Go to a different page in the Admin Console and returning to the **Content Sources > Databases** page to refresh and update the status. After a successful database synchronization, a feed appears on the **Content Sources > Feeds** page.

Q: Sun defines several types of JDBC adapters. Which ones can be used?

A: The Google Search Appliance supports the Java to DB direct adapter types.

Q: What SQL is supported in the database configuration page? SQL99? ANSI SQL?

A: This is dependent on the JDBC adapters used to connect with your database.

Q: Can the list of JDBC adapters used be updated?

A: Currently the Google Search Appliance does not support modification of the JDBC adapter list.

Q: Is the database connection secure?

A: The Google Search Appliance does not provide an encrypted connection channel to the database unless such is supported by the JDBC adapter. It is recommended that this connection be hosted over a private network link and that JDBC communication is assumed to be insecure.

Q: Can the Google Search Appliance crawl the access information from the database?

A: The Google Search Appliance does not support evaluation of database ACLs.

Q: Can the Fully Qualified Domain Name be used for the crawl patterns?

A: No. Currently, the crawl patterns must specify the appliance using its IP address.

Q: Can the result of a database synchronization be translated by using a customized XSLT to reduce the number of documents counted?

A: The Google Search Appliance does not support translation of the database feed itself. Each row from the crawl query counts as a crawled document. The number of rows produced can only be reduced by optimizing the crawl query.

Q: Is there a DTD for the XML produced from the database for providing a custom stylesheet?

A: There is no DTD for the database XML, as the structure is dependent on the SELECT clause of the crawl query. An identity transformation stylesheet is available, which allows you to see the raw database XML structure, as it appears before it is transformed by a stylesheet.

To use the identity_transform.xml and see the raw XML:

1. Download identity_transform.xml (<https://support.com/gsa/answer/6069358>).
2. Provide information about the database using the **Create New Database Source** section of the **Content Sources > Databases** page.
3. Choose identity_transform.xml after clicking **Upload Stylesheet**.
4. Make sure the **Serve Query** and **Primary Key Fields** are completed.
5. Click **Create**.
6. Click **Sync** for the database.
7. When the database synchronization and feed are completed, perform a test search of the database content. Click a result that you know should be provided by the database. The displayed page should appear as unformatted HTML.
8. To view the XML, view the source of the page in your web browser.

Q: How does the search appliance invalidate the database crawled contents of the index for a particular collection?

A: When you click the **Sync** link on the **Content Sources > Databases** page in the Admin Console for a database source name, the old contents in the index are invalidated.

Chapter 7

Constructing URL Patterns

A URL pattern is a set of ordered characters to which the Google Search Appliance matches actual URLs that the crawler discovers. You can specify URL patterns for which your index should include matching URLs and URL patterns for which your index should exclude matching URLs. This document explains how to construct a URL pattern.

Introduction

A URL pattern is a set of ordered characters that is modeled after an actual URL. The URL pattern is used to match one or more specific URLs. An exception pattern starts with a hyphen (-).

URL patterns specified in the Start and Block URLs page control the URLs that the search appliance includes in the index. To configure the crawl, use the **Content Sources > Web Crawl > Start and Block URLs** page in the Admin Console to enter URLs and URL patterns in the following boxes:

- **Start URLs**
- **Follow Patterns**
- **Do Not Follow Patterns**

The search appliance starts crawling from the URLs listed in the **Start URLs** text box. Each URL that the search appliance encounters is compared with URL patterns listed in the Follow Patterns and Do Not Follow Patterns text boxes.

A URL is included in the index when all of the following are true:

- The URL is reachable through the URLs specified in the **Start URLs** field.
- The URL matches at least one pattern in the **Follow Patterns** field.
- The URL does not match an exception pattern in the **Follow Patterns** field.
- The URL meets one of the following criteria:
 - The URL does not match a pattern in the **Do Not Follow Patterns** field.
 - The URL matches an exception in the **Do Not Follow Patterns** field.

Alternatively, URLs can be excluded from an index through the use of a `robots.txt` file or robots meta tags.

For complete information about the **Start and Block URLs** page, in the Admin Console, click **Admin Console Help > Content Sources > Web Crawl > Start and Block URLs**.

Rules for Valid URL Patterns

When specifying the URLs that should or should not be crawled on your site or when building URL-based collections, your URLs must conform to the valid patterns listed in the table that follows.

Valid URL Patterns	Examples	Explanation
Any substring of a URL that includes the host/path separating slash	<code>http://www.google.com/</code>	Any page on <code>www.google.com</code> using the HTTP protocol.
	<code>www.google.com/</code>	Any page on <code>www.google.com</code> using any supported protocol.
	<code>google.com/</code>	Any page in the <code>google.com</code> domain.
Any suffix of a string. You specify the suffix with the <code>\$</code> at the end of the string.	<code>home.html\$</code>	All pages ending with <code>home.html</code> .
	<code>.pdf\$</code>	All pages with the extension <code>.pdf</code> .
Any prefix of a string. You specify the prefix with the <code>^</code> at the beginning of the string. A prefix can be used in combination with the suffix for exact string matches. For example, <code>^candy cane\$</code> matches the exact string for "candy cane."	<code>^http://</code>	Any page using the HTTP protocol.
	<code>^https://</code>	Any page using the HTTPS protocol.
	<code>^http://www.google.com/page.html\$</code>	Only the specified page.
An arbitrary substring of a URL. These patterns are specified using the prefix "contains".	<code>contains:coffee</code>	Any URL that contains "coffee."
	<code>contains:beans.com</code>	Any URL that contains "beans.com" such as <code>http://blog.beans.com/</code> or <code>http://www.page.com/?goto=beans.com/images</code>
Exceptions denoted by <code>-</code> (minus) sign.	<code>candy.com/</code> <code>-www.candy.com/</code>	Means that "www.chocolate.candy.com" is a match, but "www.candy.com" is not a match.

Valid URL Patterns	Examples	Explanation
<p>Regular expressions from the GNU Regular Expression library. In the search appliance, regular expressions:</p> <ol style="list-style-type: none"> 1. Are case sensitive unless you specify <code>regexIgnoreCase:</code> 2. Must use two escape characters (a double backslash "\") when reserved characters are added to the regular expression. <p>Note: <code>regex:</code> and <code>regexCase:</code> are equivalent.</p>	<p>(Wrapped for readability)</p> <pre> regex:-sid=[0-9A-Z]+/ regex: http://www\\.example \\google\\.com/.*/images/ regexCase: http://www\\.example \\google\\.com/.*/images/ regexIgnoreCase: http://www\\.Example \\Google\\.com/.*/IMAGES/ </pre>	<p>See the GNU Regular Expression library (http://www.cs.utah.edu/dept/old/texinfo/regex/regex_toc.html)</p>
Comments	<code>#this is a comment</code>	Empty lines and comments starting with # are permissible. These comments are removed from the URL pattern and ignored.

Comments in URL Patterns

A line that starts with a # (pound) character is treated as a comment, as shown in the following example.

```
#This is a comment.
```

Case Sensitivity

URL patterns are case sensitive. The following table uses `www.example.com/` to illustrate an example that does not match the URL pattern, and another example that does match the pattern.

URL Pattern	<code>www.example.com/mypage</code>
Invalid URL	<code>http://www.example.com/MYPAGE.html</code>
Matching URL	<code>http://www.example.com/mypage.html</code>

The Google Search Appliance treats URLs as case-sensitive, because URLs that differ only by case can legitimately be different pages. The hostname part of the URL, however, is case-insensitive. To capture URLs with variable case use a regular expression. More information about regular expressions, see "Google Regular Expressions" on page 94.

Simple URL Patterns

The following notation is used throughout this document:

- Brackets <> denote variable strings in the expression format.
- The slash (/) at the end of the site name is required.

Format	<site>/
Example	www.example.com/

Matching domains

To match URLs from all sites in the same domain, specify the domain name. The following example matches all sites in the domain `example.com`.

Format	<domain>/
Example	example.com/
Matching URLs	www.example.com support.example.com sales.example.com

Matching directories

To describe URLs that are in a specific directory or in one of its sub-directories, specify the directory and any sub-directory in the pattern.

The following example matches all URLs in the `products` directory and all sub-directories under `products` on the site `sales.example.com`.

Format	<site>/<directory>/
Example	sales.example.com/products/
Matching URLs	sales.example.com/products/about.html http://www.sales.example.com/products/cost/priceList.html

The following example matches all URLs in the `products` directory and all sub-directories under `products` on all sites in the `example.com` domain.

Format	<domain>/<directory>/
Example	example.com/products/
Matching URLs	accounting.example.com/products/prices.htm example.com/products/expensive.htm

The following example matches all URLs in an `images` directory or sub-directory, in any side.

Note: If one of the pages on a site links to another external site or domain, this example would also match the `/image/` directories of those external sites.

Format	<code>/<directory>/</code>
Example	<code>/images/</code>
Matching URLs	<code>www.example1423.com/images/myVacation/</code> <code>www.EXAMPLE.com/images/tomato.jpg</code> <code>sales.example.com/images/</code>

Matching files

To match a specific file, specify its name in the pattern and add the dollar (\$) character to the end of the pattern. Each of the following examples will only match one page.

Format	<code><site>/<directory>/<file>\$</code>
Example	<code>www.example.com/products/foo.html</code>

Format	<code><domain>/<directory>/<file>\$</code>
Example	<code>example.com/products/foo.html</code>

Format	<code>/<directory>/<file>\$</code>
Example	<code>/products/foo.html</code>

Format	<code>/<file>\$</code>
Example	<code>/mypage.html</code>

Without the dollar (\$) character at the end of the pattern, the URL pattern may match more than one page.

Format	<code>/<directory>/<file></code>
Example	<code>/products/mypage.html</code>
Matching URLs	<code>/products/mypage.html</code> <code>/product/mypage.html</code> <code>/products/mypage.htmlx</code>

Matching protocols

To match URLs that are accessible by a specific protocol, specify the protocol in the pattern. The following example matches HTTPS URLs that contain the `products` directory.

Format	<code><protocol>://<site>/<path>/</code>
Example	<code>https://www.example.com/products/mydir/mydoc.txt/</code>

Matching ports

To match URLs that are accessible by means of a specific port, specify the port number in the pattern. If you don't specify a port, the search appliance matches any URLs with the site regardless of the port.

- These examples match host `www.example.com/foo` on any port: `www.example.com:*/foo` or `www.example.com/foo`
- This example matches host `www.example.com` on port 8888: `www.example.com:8888/`

Note: If you explicitly include a port number, the pattern matches only URLs that explicitly include the port number, even if you use the default port. For example, a URL pattern that includes `www.example.com:80/products/` does not match `www.example.com/products/`.

Using the prefix option

To match the beginning of a URL, add the caret (^) character to the start of the pattern. Do not match a prefix character followed by only a protocol because the result could resolve to most of the Internet.

Format	<code>^<protocol>://<site>/<directory>/</code>
Example	<code>^http://www.example.com/products/</code>

Format	<code>^<protocol>://<site>/</code>
Example	<code>^http://www.example.com/</code>

Format	<code>^<protocol></code>
Example	<code>^https</code>

Format	<code>^<protocol>://<partial_site></code>
Example	<code>^http://www.example</code>
Matching URLs	<code>http://www.example.com/</code> <code>http://www.example.de/</code> <code>http://www.example.co.jp/</code>

Using the suffix option

To match the end of a URL, add the dollar (\$) character to the end of the pattern.

The following example matches `http://www.example.com/mypage.jhtml`, but not `http://www.example.com/mypage.jhtml;jsessionid=HDUENB2947WSSJ23`.

Format	<code><protocol>://<site>/<directory>/<file>\$</code>
Example	<code>http://www.example.com/mypage.jhtml\$</code>

Format	<code><site>/<directory>/<file>\$</code>
Example	<code>www.example.com/products/mypage.html\$</code>

Format	<code><domain>/<directory>/<file>\$</code>
Example	<code>example.com/products/mypage.html\$</code>

Format	<code>/<directory>/<file>\$</code>
Example	<code>/products/mypage.html\$</code>

The following example matches `mypage.htm`, but does not match `mypage.html`.

Format	<code><file>\$</code>
Example	<code>mypage.htm\$</code>

The following example is useful for specifying all files of a certain type, including `.html`, `.doc`, `.ppt`, and `.gif`.

Format	<code><partial_file_name>\$</code>
Example	<code>.doc\$</code>

Matching specific URLs

To exactly match a single URL, use both caret (^) and dollar (\$). The following example matches only the URL: `http://www.example.com/mypage.jhtml`

Format	<code>^<exact url>\$</code>
Example	<code>^http://www.example.com/mypage.jhtml\$</code>

Matching specified strings

To match URLs with a specified string use the `contains:` prefix. The following example matches any URL containing the string "product."

Format	<code>contains:<string></code>
Example	<code>contains:product</code>
Matching URLs	<code>http://www.example.com/products/mypage.html</code> <code>https://sales.example.com/production_details/inventory.xls</code>

SMB URL Patterns

In GSA release 7.4, on-board file system crawling (File System Gateway) is deprecated. For more information, see [Deprecation Notices](#).

To match SMB (Server Message Block) URLs, the pattern must have a fully-qualified domain name and begin with the `smb:` protocol. SMB URLs refer to objects that are available on SMB-based file systems, including files, directories, shares, and hosts. SMB URLs use only forward slashes. Some environments, such as Microsoft Windows, use backslashes ("\") to separate file path components. However, for these URL patterns, you must use forward slashes. SMB paths to folders must end with a trailing forward slash ("/").

The following example shows the correct structure of an SMB URL.

Format	<code>smb://<fully-qualified-domain-name>/<share>/<directory>/<file></code>
Example	<code>smb://fileserver.domain/myshare/mydir/mydoc.txt</code>

The following SMB URL patterns are not supported:

- Top-level SMB URLs, such as the following: `smb://`
- URLs that omit the fully-qualified domain name, such as the following: `smb://myshare/mydir/`
- URLs with workgroup identifiers in place of hostnames, such as the following: `smb://workgroupID/myshare/`

Exception Patterns

The exception patterns below cannot be used with any version of the Google Connector for Microsoft SharePoint.

To specify exception patterns, prefix the expression with a hyphen (-). The following example includes sites in the `example.com` domain, but excludes `secret.example.com`.

Format	-<expression>
Example	<code>example.com/</code> <code>-secret.example.com/</code>

The following example excludes any URL that contains `content_type=calendar`.

Example	<code>-contains:content_type=calendar</code>
----------------	--

You can override the exception interpretation of the hyphen (-) character by preceding the hyphen (-) with a plus (+).

Example	<code>+products.xls\$</code>
Matching URLs	<code>http://www.example.com/products/new-products.xls</code>

Google Regular Expressions

A Google regular expression describes a complex set of URLs. For more information on GNU regular expressions, see the Google Search for “gnu regular expression tutorial” (<http://www.google.com/search?hl=en&q=gnu+regular+expression+tutorial>). Google regular expressions are similar to GNU regular expressions, with the exception of the following differences:

- A case insensitive expression starts with the following prefix: `regexpIgnoreCase:`
- A case sensitive expression does not require a prefix, but the `regexpCase:` and `regexp:` prefixes can be used to specify case sensitivity.
- Special characters are escaped with a double backslash (`\\`).

Metacharacters are either a special character or special character combination, which is used in a regular expression to match a specific portion of a pattern. Metacharacters are not used as literals in regular expressions. The following list describes available metacharacters and metacharacter combinations:

- The `.` character matches any character.
- The `.*` character combination matches any number of characters.
- The `^` character specifies the start of a string.
- The `$` character specifies the end of a string.
- The `[0-9a-zA-Z]+` character combination matches a sequence of alphanumeric characters.
- The following characters must be preceded with the double backslash (`\\`) escape sequence:
`^ . [$ () | * + ? { \`

The following example matches any URL that references an `images` directory on `www.example.com` using the HTTP protocol.

Example	<code>regex:http://www\\.example\\.com.*images/</code>
Matching URLs	<code>http://www.example.com/images/logo.gif</code> <code>http://www.example.com/products/images/widget.jpg</code>

The following example matches any URL in which the server name starts with `auth` and the URL contains `.com`.

Example	<code>regexCase:http://auth.*\\.com/</code>
Matching URLs	<code>http://auth.www.example.com/mypage.html</code> <code>http://auth.sales.example.com/about/corporate.htm</code>

This example does not match `http://AUTH.engineering.example.com/mypage.html` because the expression is case sensitive.

The following pattern matches JHTML pages from site `www.example.com`. These pages have the `jsessionid`, `type=content` parameters, and `id`.

Example	<code>regex:^http://www\\.example\\.com/page\\.jhtml;jsessionid=[0-9a-zA-Z]+&type=content&id=[0-9a-zA-Z]+\$</code>
Matching URLs	<code>http://www.example.com/page.jhtml;jsessionid=A93KF8M18M5XP&type=content&id=gpw9483</code>

Note: Do not begin or end a URL pattern with period+asterisk (.*). If you are using the `regex:` prefix, as this pattern is ineffective and may cause performance problems.

Note: Invalid regular expression patterns entered on the **Content Sources > Web Crawl > Start and Block URLs** page in the Admin Console can cause search appliance crawling to fail.

For proxy servers, regular expressions are also case sensitive, but must use a single escape character (backslash “\”) when reserved characters are added to the regular expression.

Using Backreferences with Do Not Follow Patterns

A backreference stores the part of a URL pattern matched by a part of a regular expression that is grouped within parentheses. The search appliance supports using backreferences with **Do Not Follow Patterns**. The **Content Sources > Web Crawl > Start and Block URLs** page in the Admin Console includes default backreferences in the **Do Not Follow Patterns**. The search appliance does not support using backreferences with **Follow Patterns**.

The following examples illustrate backreferences and are similar to the default backreferences on the **Content Sources > Web Crawl > Start and Block URLs** page. These backreferences prevent the search appliance from crawling repetitive URLs.

Example	<code>regexp:example\\.com/.*/([^\s]*)/\\1/\\1/</code>
Matching URL	<code>http://example.com/corp/corp/corp/...</code>
Example	<code>regexp:example\\.com/.*/([^\s]*)/([^\s]*)/\\1/\\2/</code>
Matching URL	<code>http://example.com/corp/hr/corp/hr/...</code>
Example	<code>regexp:example\\.com/.*&([^\s]*)&\\1&\\1</code>
Matching URL	<code>http://example.com/corp?hr=1&hr=1&hr=1...</code>

Controlling the Depth of a Crawl with URL Patterns

Google recommends crawling to the maximum depth, allowing the Google algorithm to present the user with the best search results. You can use URL patterns to control how many levels of subdirectories are included in the index.

For example, the following URL patterns cause the search appliance to crawl the top three subdirectories on the site `www.mysite.com`:

```
regexp:www\\.mysite\\.com/[^\s]*$
regexp:www\\.mysite\\.com/[^\s]*/[^\s]*$
regexp:www\\.mysite\\.com/[^\s]*/[^\s]*/[^\s]*$
```


Chapter 8

Crawl Quick Reference

Crawling is the process where the Google Search Appliance discovers enterprise content to index. This chapter provides reference information about crawl administration tasks.

Crawling and Indexing Features

The following table lists Google Search Appliance crawl and index features. For each feature, the table lists the page in the Admin Console where you can use the feature and a reference to a section in this document that describes it.

Feature	Admin Console Page	Reference
Always force recrawl URLs	Content Sources > Web Crawl > Freshness Tuning	"Freshness Tuning" on page 58
Content statistics	Index > Diagnostics > Content Statistics	"Using the Admin Console to Monitor a Crawl" on page 45
Continuous crawl	Content Sources > Web Crawl > Crawl Schedule	"Selecting a Crawl Mode" on page 42
Coverage tuning	Content Sources > Web Crawl > Coverage Tuning	"Coverage Tuning" on page 58
Index diagnostics	Index > Diagnostics > Index Diagnostics	"Using the Admin Console to Monitor a Crawl" on page 45
Crawl frequently URLs	Content Sources > Web Crawl > Freshness Tuning	"Freshness Tuning" on page 58
Crawl infrequently URLs	Content Sources > Web Crawl > Freshness Tuning	"Freshness Tuning" on page 58
Crawl modes	Content Sources > Web Crawl > Crawl Schedule	"Selecting a Crawl Mode" on page 42
Crawl queue snapshots	Content Sources > Diagnostics > Crawl Queue	"Using the Admin Console to Monitor a Crawl" on page 45
Crawl schedule	Content Sources > Web Crawl > Crawl Schedule	"Scheduling a Crawl" on page 42

Feature	Admin Console Page	Reference
Crawl status	Content Sources > Diagnostics > Crawl Status	"Using the Admin Console to Monitor a Crawl" on page 45
Crawl URLs	Content Sources > Web Crawl > Start and Block URLs	"Configuring a Crawl" on page 34
Do not follow patterns	Content Sources > Web Crawl > Start and Block URLs	"Configuring a Crawl" on page 34
Document dates	Index > Document Dates	"Defining Document Date Rules" on page 41
Duplicate hosts	Content Sources > Web Crawl > Duplicate Hosts	"Preventing Crawling of Duplicate Hosts" on page 61
Entity recognition	Index > Entity Recognition	"Discovering and Indexing Entities" on page 65
Follow patterns	Content Sources > Web Crawl > Start and Block URLs	"Configuring a Crawl" on page 34
Freshness tuning	Content Sources > Web Crawl > Freshness Tuning	"Freshness Tuning" on page 58
Host load exceptions	Content Sources > Web Crawl > Host Load Schedule	"Configuring Web Server Host Load Schedules" on page 61
Host load schedule	Content Sources > Web Crawl > Host Load Schedule	"Configuring Web Server Host Load Schedules" on page 61
HTTP headers	Content Sources > Web Crawl > HTTP Headers	"Identifying the User Agent" on page 57
Index limits	Index > Index Settings	"Changing the Amount of Each Document that Is Indexed" on page 59
Infinite space detection	Content Sources > Web Crawl > Duplicate Hosts	"Enabling Infinite Space Detection" on page 61
Maximum number of URLs to crawl	Content Sources > Web Crawl > Host Load Schedule	"Configuring Web Server Host Load Schedules" on page 61
Metadata indexing	Index > Index Settings	"Configuring Metadata Indexing" on page 59
Proxy servers	Content Sources > Web Crawl > Proxy Servers	"Crawling over Proxy Servers" on page 60
Recrawl URLs	Content Sources > Web Crawl > Freshness Tuning	"Freshness Tuning" on page 58
Scheduled crawl	Content Sources > Web Crawl > Crawl Schedule	"Selecting a Crawl Mode" on page 42
Start crawling from the following URLs	Content Sources > Web Crawl > Start and Block URLs	"Configuring a Crawl" on page 34
Web server host load	Content Sources > Web Crawl > Host Load Schedule	"Configuring Web Server Host Load Schedules" on page 61

Crawling and Indexing Administration Tasks

The following table lists Google Search Appliance crawl and index administration tasks. For each task, the table gives a reference to a section in this document that describes it, as well as the page in the Admin Console that you use to accomplish the task.

Task	Reference	Admin Console Page
Prepare your data for crawling: robots.txt, Robots META tags, googleoff/googleon tags, no_crawl directories, shared folders, and jump pages	"Preparing Data for a Crawl" on page 28	
Setup the crawl path: start URLs, follow patterns, do not follow patterns	"Configuring a Crawl" on page 34	Content Sources > Web Crawl > Start and Block URLs
Test URL patterns in the crawl path	"Testing Your URL Patterns" on page 37	
Select a crawl mode: continuous crawl or scheduled crawl	"Selecting a Crawl Mode" on page 42	Content Sources > Web Crawl > Crawl Schedule
Schedule a crawl	"Scheduling a Crawl" on page 42	
Configure a continuous crawl: URLs to crawl frequently, URLs to crawl infrequently, URLs to always force recrawl	"Freshness Tuning" on page 58	Content Sources > Web Crawl > Freshness Tuning
Pause or restart a continuous crawl	"Stopping, Pausing, or Resuming a Crawl" on page 43	Content Sources > Diagnostics > Crawl Status
Stop a scheduled crawl		
Submit a URL to be recrawled	"Freshness Tuning" on page 58	Content Sources > Web Crawl > Freshness Tuning
	"Submitting a URL to Be Recrawled" on page 43	Index > Diagnostics > Index Diagnostics
Change the amount of each document that is indexed	"Changing the Amount of Each Document that Is Indexed" on page 59	Index > Index Settings
Configure metadata indexing	"Configuring Metadata Indexing" on page 59	
Set up entity recognition	"Discovering and Indexing Entities" on page 65	Index > Entity Recognition
Control the number of URLs the search appliance crawls for a site	"Coverage Tuning" on page 58	Content Sources > Web Crawl > Coverage Tuning
Set up proxies for Web servers	"Crawling over Proxy Servers" on page 60	Content Sources > Web Crawl > Proxy Servers
Locate or change the user-agent name	"Identifying the User Agent" on page 57	Content Sources > Web Crawl > HTTP Headers
Enter additional HTTP headers for the search appliance crawler to use		

Task	Reference	Admin Console Page
Prevent recrawling of content that resides on duplicate hosts	"Preventing Crawling of Duplicate Hosts" on page 61	Content Sources > Web Crawl > Duplicate Hosts
Prevent crawling of duplicate content to avoid infinite space indexing	"Enabling Infinite Space Detection" on page 61	
Define rules for the search appliance crawler to use as it indexes documents	"Defining Document Date Rules" on page 41	Index > Document Dates
Specify the maximum number of URLs to crawl for a host and the average number of concurrent connections to open to each Web server for crawling	"Configuring Web Server Host Load Schedules" on page 61	Content Sources > Web Crawl > Host Load Schedule
View the current crawl mode and summary information about events of the past 24 hours in a crawl	"Using the Admin Console to Monitor a Crawl" on page 45	Content Sources > Diagnostics > Crawl Status
View crawl history for all hosts, a specific host, or a specific file		Index > Diagnostics > Index Diagnostics
Define and view a snapshot of uncrawled URLs in the crawl queue		Content Sources > Diagnostics > Crawl Queue
View summary information about files that have been crawled		Index > Diagnostics > Content Statistics
View current license information	"What Is the Search Appliance License Limit?" on page 22	Administration > License

Admin Console Basic Crawl Pages

The following table lists Google Search Appliance Admin Console pages that are used to administer a basic crawl. For each Admin Console page, the table provides a reference to a section in this document that describes using the page.

Admin Console Page	Reference
Content Sources > Web Crawl > Start and Block URLs	"Configuring a Crawl" on page 34
Content Sources > Web Crawl > Crawl Schedule	"Selecting a Crawl Mode" on page 42 "Scheduling a Crawl" on page 42
Content Sources > Web Crawl > Proxy Servers	"Crawling over Proxy Servers" on page 60
Content Sources > Web Crawl > HTTP Headers	"Identifying the User Agent" on page 57
Content Sources > Web Crawl > Duplicate Hosts	"Preventing Crawling of Duplicate Hosts" on page 61
Index > Document Dates	"Defining Document Date Rules" on page 41
Content Sources > Web Crawl > Host Load Schedule	"Configuring Web Server Host Load Schedules" on page 61
Content Sources > Web Crawl > Coverage Tuning	"Coverage Tuning" on page 58
Content Sources > Web Crawl > Freshness Tuning	"Freshness Tuning" on page 58
Index > Index Settings	"Changing the Amount of Each Document that Is Indexed" on page 59 "Configuring Metadata Indexing" on page 59
Index > Entity Recognition	"Discovering and Indexing Entities" on page 65
Content Sources > Diagnostics > Crawl Status	"Using the Admin Console to Monitor a Crawl" on page 45
Index > Diagnostics > Index Diagnostics	
Content Sources > Diagnostics > Crawl Queue	
Index > Diagnostics > Content Statistics	

Index

Symbols

- ^ character 91, 92, 94
- . character 94
- * character 94
- .tar files 37
- .tar.gz files 37
- .tgz files 37
- .zip files 37
- #suffixes 55
- \$ character 90, 92, 94

Numerics

- 200 status code 22
- 301 status code 51
- 302 status code 51
- 304 status code 17
- 401 status code 51, 52
- 404 status code 51, 52
- 500 status code 51
- 501 status code 51

A

- Administration > License page 23, 26
- Administration > Network Settings page 29, 35
- Administration > System Settings page 24
- area tag 12
- authentication 9, 26, 28, 47, 51

B

- backreferences 96
- BroadVision Web server 53

C

- caret character 91, 92
- case sensitivity in URL patterns 88
- checksum 16, 17
- ColdFusion application server 56
- collections
 - default collection 62
 - description 62
 - URL patterns 62
- comments in URL patterns 88

- compressed files, indexing 10, 37
- contains prefix 93
- content

- amount indexed 59
- can be crawled 9–10
- checksum 17
- complex 48
- compressed files 10
- database data source 76
- databases 10
- network file shares 10
- non-HTML 48
- not crawled 10–11
- public web content 9
- secure web content 9

- Content Sources > Databases page 38, 44, 74–85
- Content Sources > Diagnostics > Crawl Queue page 46, 50

- Content Sources > Diagnostics > Crawl Status page 24, 43, 46, 48

- Content Sources > Feeds page 82

- Content Sources > Web Crawl > Coverage Tuning page 58

- Content Sources > Web Crawl > Crawl Schedule page 42, 51

- Content Sources > Web Crawl > Duplicate Hosts page 61

- Content Sources > Web Crawl > Freshness Tuning page 21, 22, 43, 48, 58

- Content Sources > Web Crawl > Host Load Schedule page 18, 48, 61

- Content Sources > Web Crawl > HTTP Headers page 57

- Content Sources > Web Crawl > Proxy Servers page 60

- Content Sources > Web Crawl > Secure Crawl > Crawler Access page 28, 52

- Content Sources > Web Crawl > Start and Block URLs 21

- Content Sources > Web Crawl > Start and Block URLs page 18, 23, 34, 52, 56, 62, 83, 86–87, 96

- Content Sources > Web Crawl > Start and Block
 - URLs page 26, 35, 36, 37
- continuous crawl 8
- coverage tuning 58
- crawl
 - compressed files 37
 - configuring 34–38
 - continuous 8
 - coverage tuning 58
 - crawl query, SQL 77
 - databases 38, 44, 72–85
 - depth 96
 - description 7
 - do not crawl URLs 36
 - duplicate content 61
 - duplicate hosts 61
 - end of 21
 - excluded content 10
 - excluding directories 33
 - excluding URLs 36
 - file shares 17
 - follow URLs 35
 - freshness tuning 58
 - Google regular expressions 37
 - JavaScript 63–65
 - modes 8, 42
 - monitoring 45
 - new 16
 - overview 13–20
 - path 12
 - patterns 11, 19, 21, 23, 24, 25, 26, 34, 35, 36, 86–96
 - pausing 43
 - preparing 28–41
 - prohibiting 11
 - proxy servers 60
 - queue 14, 15–16, 17, 19, 21
 - recrawl 16, 21, 43
 - resuming 43
 - scheduled 8, 42
 - secure content 40
 - slow rate 48–49
 - SMB URLs 38–40
 - start URLs 34
 - status messages 47
 - stopping 43
 - testing patterns 37
 - URLs to crawl frequently 21

D

- database crawl
 - advanced settings 78, 80
 - settings 76
 - troubleshooting 82–83
 - URL patterns 81
- database feed 75
- database stylesheet 78
- database synchronization 82

- databases 10
 - crawling 38, 44, 72–85
 - data source information 76
 - supported 73
 - synchronizing 74
- date formats, metadata 60
- dates
 - configuring 40
 - document date rules 41
- DB2 73
- depth of crawl 96
- directories
 - matching in URLs 89
- directories, excluding from crawl 33
- do not crawl URLs 36
- do not follow patterns 86
- document dates 24
- documents
 - amount indexed 59
 - removing from index 25
 - removing from servers 26
- dollar character 90, 92
- Domain Name Services (DNS) 40
- domains, matching 89
- duplicate hosts 61

E

- entity recognition 65–66
- exception patterns 94

F

- feeds
 - database 73, 75
 - monitoring 82
 - stylesheet 78
 - web 22
 - XML 78
- file shares 17, 33
- files
 - matching in URLs 90
 - size 18
 - type 18
- follow URLs 35, 86
- freshness tuning 58

G

- Google regular expressions 37, 94–95
- googleoff and googleon tags 31
- gsa-crawler 57

H

- host load 48, 61
- hostname resolution 40
- HTML documents 18
- HTTP status codes 51

I

- If-Modified-Since headers 17
- index 7, 8, 16, 18, 19, 21, 22, 25, 26, 31, 35, 47, 59, 62

- Index > Collections page 62
- Index > Diagnostics > Content Statistics page 46
- Index > Diagnostics > Index Diagnostics page 19, 21, 22, 43, 46, 47, 50, 51, 62
- Index > Document Dates page 41
- Index > Entity Recognition page 65, 66
- Index > Index Settings page 59, 60
- index pages 56
- indexing, wildcard 71
- infinite space detection 61

J

- Java servlet containers 54
- JavaScript crawling 63

L

- license
 - expiration 24
 - limit 21, 22–24, 30, 36, 75, 77
- links 19
- Lotus Domino Enterprise 54

M

- metadata
 - date formats 60
 - entity recognition 65–66
 - excluding names 59
 - including names 59
 - indexing 59–60
 - multivalued separators 60
- Microsoft Commerce Server 54
- multivalued separators 60
- MySQL 73

N

- network
 - connectivity 22, 47
 - file shares 10
 - problems 49
- new crawl 16
- noarchive 30, 31
- no_crawl directories 33
- nofollow 30, 31
- noindex 30, 31
- non-HTML content 48

O

- OpenDocument 55
- Oracle 73

P

- Pattern Tester Utility page 37
- patterns, URL 12, 19, 21, 23, 24, 25, 26, 62, 81, 86–96
- ports, matching in URLs 91
- prefix option in URLs 91
- protocols, matching in URLs 91
- proxy servers 60
- public web content 9

Q

- query load 49
- queue, crawl 14–16, 17, 19, 21

R

- recrawl 16, 21, 43, 50
- redirects
 - cyclic 53
 - logical 63
- regular expressions 88
- removing
 - documents from index 25–27, 62
 - documents from servers 26
- robots meta tag 11, 30
- robots.txt file 11, 28–29

S

- scheduled crawl 8
- Search > Search Features > Dynamic Navigation page 59
- Search > Search Features > Front Ends page 26
- search results
 - database 76
 - delay after crawl 21
- secure content 40
- secure web content 9
- SMB URLs 38–40, 93
- SQL crawl query 77
- SQL serve query 77
- SQL Server 73
- start URLs 34, 86
- status messages 47
- strings, matching 93
- suffix option in URLs 92
- Sun Java System Web Server 54
- Sybase 73
- synchronizing a database 74

T

- TableCrawler 73
- TableServer 73
- test network connectivity 47
- text, excluding from the index 31

U

- unlinked URLs 12, 33

URLs

- #suffixes 55
- BroadVision web server 53
- ColdFusion application server 56
- crawl frequently 21
- directory names 89
- domain names 89
- exception patterns 94
- fetching 16
- file names 90
- following 19
- generated URLs, database crawl 75
- index pages 56
- Java servlet containers 54
- Lotus Domino Enterprise 54
- maximum number crawled 24
- Microsoft Commerce Server 54
- multiple versions 55
- OpenDocument 55
- patterns 19, 21, 23, 24, 25, 26, 34, 35, 36, 62, 81, 86–96
- ports 91
- prefix option 91
- priority in the crawl queue 15
- protocols 91
- rewrite rules 53–56
- SMB 38–40, 93
- specific matches 92
- start URLs 34
- suffix option 92
- Sun Java Web Server 54
- testing patterns 37
- unlinked 12, 33
- user agent 57

V

- valid URL patterns 87

W

- wait times 50
- web servers 49
 - errors 50
- wildcard indexing 71

X

- X-Robots-Tag 31