

Google Search Appliance

Creating the Search Experience

Google Search Appliance software version 7.2 and later



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

GSA-SE_200.04
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
	About this Document	7
	What Is the Search Experience?	8
	What Is Personalization?	9
	Focusing on End Users	9
	Starting with a Basic Search Experience	9
	Customizing the Basic Search Experience	11
	Creating Multiple Search Experiences	11
	Managing the Search Experience	14
	Using Collections with Front Ends	14
	Maximum Number of Front Ends and Collections	15
	Improving Searches	16
	Suggesting Alternative Search Terms	16
	Guiding End Users to Specific URLs	17
	Narrowing Searches	18
	Widening Searches	19
	Enhancing Search Results	19
	Integrating Real-Time Data	20
	Providing Expert Profiles with Search Results	21
	Giving Users the Ability to Add Results	21
	Showing Document Previews	21
	Refining Search Results	22
	Removing Specific URLs from Results	22
	Influencing Results Ranking	23
	Enabling Alerts	24
	Changing the User Interface	24
	Where Is the Search Experience Created?	26
	Elements Defined in the Front End	27
	Elements Defined on Other Admin Console Pages	28
	Built-In Elements	28
	Search Experience Background	29
	Entering a Search Query	29
	Converting the Search Query HTML to a URL	29
	Executing the Search	30
	Returning Search Results as XML	31
	Applying an XSLT Stylesheet to the XML Results and Create HTML Output	31
	Presenting Search Results to the End User	31

Chapter 2	Best Practices	32
	Search Experience Best Practices	32
	Checklist of Best Practices	32
	Personalizing the Search Experience	34
	The Relationship Between the Search Experience and Front Ends	35
	Creating Multiple Front Ends	36
	Gathering Information about the Search Experience	36
	Viewing Detailed Data about User Clicks	37
	Using Advanced Search Reporting to Personalize the Search Experience	40
	Generating an Advanced Search Report	41
	Using the Automatic Self-Learning Scorer	42
	Using KeyMatches to Guide Users to URLs	43
	Working with KeyMatches	44
	Identifying KeyMatches	44
	Changing the Appearance of KeyMatches	44
	Using Related Queries to Suggest Alternative Searches	45
	Working with Related Queries	46
	Identifying Related Queries	46
	Changing the Appearance of Related Queries	46
	Using Dynamic Result Clusters to Narrow Searches	47
	Enabling or Disabling Dynamic Result Clusters	47
	Using Dynamic Navigation to Help Users Explore Results	49
	Dynamic Navigation for Secure Search	50
	Enabling Dynamic Navigation	51
	Creating a Configuration and Adding Attributes	51
	Showing Dynamic Navigation in a Front End	52
	Configuring Query Expansion for Dynamic Navigation	52
	Providing Expert Search for Users	53
	Configuring Data Sources for Expert Search	53
	Enabling and Configuring Expert Search	53
	Using People Search	54
	Experimenting with Host Crowding Options	56
	Using Filters to Restrict Search Results	56
	Working with Filters	57
	Restricting Search Results by Domain Name	57
	Restricting Search Results by Language	58
	Restricting Search Results by File Type	58
	Restricting Search Results by Meta Tag	58
	Removing URLs from Search Results	59
	Using Query Expansion to Widen Searches	60
	Working with Query Expansion Files	60
	Setting the Query Expansion Policy for a Front End	65
	Enabling Translation of Search Results	66
	Changing Languages for Query Expansion and Spelling Suggestions	66
	Enabling/Disabling Click-Jacking Defense	70
	Using OneBox Modules to Integrate Structured Results	70
	Using Result Biasing to Influence Result Ranking	71
	Working with Result Biasing	72
	Using Source Biasing	72
	Using Date Biasing	73
	Using Metadata and Entity Biasing	74
	Providing Alerts for End Users	75
	Enabling Alerts	76
	Showing the My Alerts Link	76

Providing User Results	77
Adding a User Results Configuration	77
Moderating User Results	78
Enabling Authentication for User Results	78
How Users Can Add Results	78
Providing Query Suggestions	79
Exporting Query Suggestions	80
Exporting and Importing the Suggestions Blacklist	80
Providing Document Previews	81
Enabling Wildcard Search	82
Tuning Search Results	83
Optimizing the Speed of Results	83
Analyzing Searches that Do Not Return Results	84
Managing the Search Index	85
Keeping the Search Index Clean	86
Segmenting Data in the Search Index	86
Evaluating Search Performance	88
Adding a Feedback Mechanism for Users	88
Conducting a User Satisfaction Survey	89
Exchanging Information on Google Groups	90
Updating Your Search Appliance	91
Chapter 3	
Customizing the User Interface	92
What Is the Google Search Appliance User Interface?	92
What Is the Relationship Between a User Interface and a Front End?	93
What Is the Relationship Between a User Interface and an XSLT Stylesheet?	94
What Tools Can I Use to Customize a User Interface?	95
What Knowledge Do I Need to Customize a User Interface?	95
Getting Started with Customizing the User Interface	96
Working with the Page Layout Helper	96
Customizing Global Attributes	97
Customizing the Search Box	99
Modifying Search Results	101
Customizing the User Interface in the XSLT Stylesheet	105
Working with the XSLT Stylesheet Editor	106
Changing Variables in the XSLT Stylesheet	108
Changing the Language of the User Interface	113
Customization Process Overview	113
Viewing User Interface Changes in a Browser Window	114
Managing Customized XSLT Stylesheets	115
Migrating a Customized XSLT Stylesheet to a New Software Release	115
User Interface Design Principles	116
Keep Search Pages Clean, Simple, and Fast	116
Keep Advanced Search Separate	116
Make Search Ubiquitous	116
Make Sure the Search Box Is Big Enough	116
Make Sure the User Knows what Documents Have Been Searched	117
Make Help Easily Available	117
Make Search Results Pages Useful	117
Keep the Number of Results Reasonable	117

Chapter 4	Advanced Customization Topics	118
	About This Document	118
	Audience	118
	The Customizations and Example Pages	118
	For More Information	120
	Support Limitations	120
	Integrating Search into Web Pages	121
	Presenting the Search Interface in an Inline Frame	121
	Creating an HTML Form Search Box	122
	Combining an HTML Form Search Box with Inline Results	124
	Modifying the Search Query	124
	Replacing the Secure Search Radio Button with a Check Box	124
	Specifying Query Parameters with XSLT	127
	Enhancing Search Results	129
	Including an Image Link for Results	130
	Replacing the Result Snippet with Custom Text	131
	Display Links in Dynamic Navigation to Sort Results by Metadata	133
	Configuring a Front End to Serve Secure and Public Results	136
	Disabling Filtering for a Front End	137
	Disabling Filtering for a Search Results Page	137
	Disabling Filtering for an Advanced Search Results Page	137
	Reloading the XSLT Stylesheet	138
	Reference	138
	Optional Example Materials	138
	Comparisons of Default and Customized Stylesheets	140
Appendix A	Quick Reference	141
	Search Experience Features	141
	Search Experience Administration Best Practices	143
	Admin Console Pages	144
	Index	146

Chapter 1

Introduction

The Google Search Appliance has features that enable system administrators to enhance the search experience for end users. This chapter introduces fundamental concepts of the search experience.

About this Document

Creating the Search Experience describes how system administrators can use Google Search Appliance features to create the search experience for end users. Understanding the Search Experience is the first chapter of Creating the Search Experience. This chapter is an overview of features that contribute to an end-user's search experience using the Google Search Appliance. These features include:

- Filters
- Front ends
- KeyMatch
- OneBox modules
- Related queries
- Remove URLs
- Dynamic result clusters
- Dynamic navigation
- Query expansion
- Result biasing
- Expert search
- User results
- Document previews

Other chapters in Creating the Search Experience provide information about how search appliance administrators can use these features to enhance and personalize the search experience.

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

What Is the Search Experience?

Whenever an end user tries to find information using a search box on a Web page, the end user has a search experience. The end user may be researching a topic, trying to locate a specific document, or just trying to find an answer to a question. An end user's search experience has three basic steps:

1. Formulating and entering a search query on a Web page
2. Getting search results back from the search engine
3. Interacting with the search results

Following these three steps to search for information on Google.com has become an everyday experience for many people. With the Google Search Appliance, an end user can have a search experience that is similar to that of using Google.com. The search appliance can be used by various, distinct groups of end users, including consumers and internal staff. End users can search enterprise content, which ranges from consumer-oriented public documents to secure proprietary documents.

With minimal customization, you, as a search appliance administrator, can create one or more search experiences that address the special considerations of enterprise search. With each search experience, you can focus on the needs and levels of different end users. You can:

- Present customized search pages for specific types of end users
- Improve searches in ways that pertain to specific types of end users
- Serve results that contain the right information for specific types of end users


This document describes how you can use search appliance features to create appropriate search experiences for your end users. The following table gives an overview of the major sections in this document.

Section	Describes
"Focusing on End Users" on page 9	How you can create different search experiences for several types of end users
"Managing the Search Experience" on page 14	How a search appliance feature called a "front end" manages various elements of the search experience
"Improving Searches" on page 16	Which Search appliance features enable you to improve the end-user's search
"Enhancing Search Results" on page 19	Which search appliance features enable you to enhance results listings
"Changing the User Interface" on page 24	Which search appliance features enable you to customize the search and results pages
"Where Is the Search Experience Created?" on page 26	Where to find features in the search appliance that you can use to customize the search experience
"Search Experience Background" on page 29	What happens to a single search query behind the search experience

What Is Personalization?

Several Google Search Appliance features enable you, as a Google Search Appliance administrator, to personalize the search experience. With personalization, users get results that are appropriate to their interests, roles, departments, locations, languages, or other characteristics. For more information, refer to *Personalizing the Search Experience*.

Before you personalize the search experience, you should gather knowledge about your end users, such as their roles, functional groups, locations, what they are searching for, and whether they are finding it or not. Advanced Search Reporting enables you to gather information about user clicks. For more information, refer to *Gathering Information About the Search Experience*.

 In this document, descriptions of features that you can use to personalize the search experience are marked with the following personalization icon.

Focusing on End Users

The most effective way for you to create an appropriate search experience is to focus on the end user. End users might be:

- Customers, about whom little is known other than they want to search within the enterprise for general information about products
- Members of the organization, with different jobs, different levels of expertise, different levels of security, and different expectations about search results

This section describes how you can create search experiences for different types of end users. Suppose you have two major goals for your search appliance:

- To begin serving search results immediately
- To present multiple search experiences to various types of end users

To accomplish both goals, you have decided to deploy search experiences in three phases:

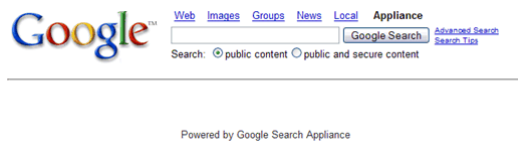
- Phase one—Start with a basic search experience that uses the search appliance defaults
- Phase two—Present a single, customized search experience that replaces the Google visual identity with that of your company
- Phase three—Present multiple search experiences aimed at different types of end users

The following sections describe how you might implement each of these three phases.

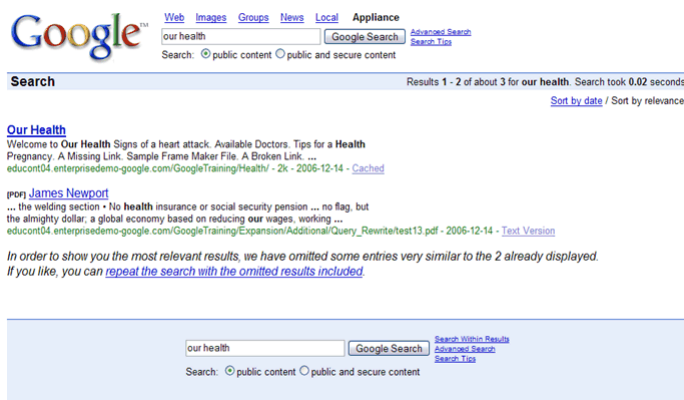
Starting with a Basic Search Experience

Suppose you want to begin serving information to your end users as soon as possible, so you have decided to begin by using the search appliance without any customization. The search appliance comes with several built-in features that make it ready for end-user searches after it has been installed and has a search index. Once end users are directed to the search page, they can immediately start entering search queries and getting relevant results.

In this phase, you can use the default search and results pages, which are both hosted by the search appliance. By default, the search page presents the Google identity and enables end users to search public content, secure content, or both. It also includes links for **Advanced Search** and **Search Tips**. The following figure illustrates the default search page.



Using this search page, an end user can begin a search by entering search terms in the search box and clicking **Google Search**. The search appliance serves search results on the default results page, shown in the following figure.



Result Listings

For each result in the list, the default result page includes:

- A title
- A snippet
- A link URL
- File size
- Date
- A link to a cached page

For security reasons, the cached result page does not contain some HTML data that is in the crawled page. For example, Javascript code is removed from cached pages.

The default results page also includes a search box at the bottom of the page, as well as a link to **Search Within Results**.

For more information about search appliance defaults, refer to “Built-In Elements” on page 28.

Customizing the Basic Search Experience

In phase two, suppose you want to customize one search experience for all your end users. You want to replace the Google visual identity with that of your company, and make a few other minimal changes. As in phase one, the search page is hosted by the search appliance.

Simple changes that you can make to the search page include:


- Changing the font face to a serif typeface
- Adding your company's logo
- Changing the search button label
- Removing the radio buttons for Search public content or public and secure content

These changes are also apparent on the new results page. Other changes that you can make to the results page include:

- Removing the link URL from the results listings
- Displaying a cached link in the results listings
- Removing the search box from the bottom of the page

For information about making the types of changes described in this section, refer to “Changing the User Interface” on page 24.

Creating Multiple Search Experiences

 In phase three, suppose you want to address various types of end users, including consumers, as well as your company's employees, including engineers, sales people, and support staff. You plan to deploy multiple search experiences that:

- Address diverse end users
- Support two languages

With each search experience, end users search the same corpus (a set of data or documents stored in a repository that is searchable by end users). However, each search experience:

- Presents a different user interface to end users
- Searches only part of the entire search index (called a “collection”)
- Behaves differently when searching and serving results

An alternative to including a search box and button on a search page that is hosted by the search appliance is to include them on a home page or other proprietary Web page that is hosted by a Web server. During this phase, you might move the search box and search button to your company's home page.

Addressing Diverse End Users

This example illustrates how you might deploy a search experience for consumers with varying levels of knowledge about your company's products. End users who might search the site range from people who know nothing about your products to knowledgeable professionals. When the search appliance serves results with this search experience, it only presents two elements in each results listing, a title and a snippet. The following example shows the results listing for the search term “headphones”:

Headphones and headsets

Suitable for all users, these headphones are for any type of listening and feature natural sound...

Headphones - stereo

For DJs, these headphones feature powerful bass and frequency...

Headsets-Microphone

Hands-free headband microphone with a portable amplifier...

To navigate to a document, an end user clicks a result title.

Suppose that in addition to general end users, the search appliance also serves engineers who are employees of your company. For these engineers, you create a specific search experience. When an internal engineer searches using the term “headphones,” the search appliance serves the same results as in the general search experience. However, in this instance, the results listings include a link URL, page size, and date information, as shown in the following example. Because more knowledgeable end users often search by URL rather than result title, they need the additional information to navigate to the appropriate page.

Headphones and headsets

Suitable for all users, these headphones are for any type of listening and feature natural sound...

<http://www.cosmoaud.com/support/allusers.html> -4k-2007-2-12

Headphones - stereo

For DJs, these headphones feature powerful bass and frequency...

<http://www.cosmoaud.com/support/djs.html> -2k-2006-11-21

Headsets-Microphone

Hands-free headband microphone with a portable amplifier...

<http://www.cosmoaud.com/support/handsfree.html>-2k-2007-3-14

To navigate to a document, an end user clicks a result title or a link URL.

For information about making the types of changes described in this section, refer to “Changing the User Interface” on page 24.

The following example shows another way of addressing diverse end users with different search experiences. Suppose both consumers and human resources staff search on the term “SA,” but expect completely different results. Because search results can be customized to return search suggestions at the top of the results list, you might create different search suggestions for the search term “SA.” For customers, the search term “SA” causes the search appliance to return the following search suggestion:

You could also try: [Service Agreement](#)

For human resources staff, the same search term causes the search appliance to return a different search suggestion:


You could also try: [Salary Adjustment](#)

For information about making the types of changes described in this section, refer to “Suggesting Alternative Search Terms” on page 16.

Other elements that you can use to provide feedback to customers include specific URLs that are promoted to the top of the results and sub-categories of search terms that are based on the initial search term. For more information about these and other elements, refer to “Improving Searches” on page 16.

For another example of a front end that serves a diverse audience, visit <http://www.apple.com>.

Supporting Multiple Languages

 This example illustrates how you can deploy search experiences in two different languages. Suppose your company serves consumers in both the United States and Canada. You might create two search experiences: one for English-speaking users and one for French-speaking users. On the search page, you can give end users the choice of viewing pages in English or French. This approach enables the search appliance to serve results in the language of users.

The following figure illustrates a results page in French.



The following table gives highlights of some differences between the English and French search experience.

Element	English	French
Search button	"Search Google "	"Recherche Google "
Search Information bar	"Search"	"Rechercher"
Results summary	"Results 1- n of about n for..."	Resultats 1- n sur environ n pour...
Next link	"Next"	"Suivant"
Results	In English	In French

For more examples of how search experiences can support multiple languages, visit Google Canada (<http://www.google.ca/>) and Google Mexico (<http://www.google.com.mx/>).


For information about making the types of changes described in this section, refer to "Refining Search Results" on page 22 and "Changing the User Interface" on page 24.

Managing the Search Experience

The Google Search Appliance feature that enables you to create different search experiences for end users is the “front end.” A front end is a framework that manages most of the elements of a single search experience, including:

- The appearance of search and results pages
- The data that is returned in search results
- The arrangement of the search results

A default front end is built into the search appliance. You can use the default front end to deploy a single search experience for end users. The examples in “Starting with a Basic Search Experience” on page 9 illustrate this approach. Another approach that uses a single front end is to deploy a single, customized search experience for end users. The examples in “Customizing the Basic Search Experience” on page 11 illustrate this approach.

 You can create multiple front ends to deploy multiple search experiences for end users. The examples in “Creating Multiple Search Experiences” on page 11 illustrate this approach.

There are several search appliance features associated with a front end, including features that give end users feedback on their searches and features that refine search results. You create and manage search experiences using anywhere from a few to all front end features. For descriptions of these features, refer to:

- “Improving Searches” on page 16
- “Enhancing Search Results” on page 19
- “Changing the User Interface” on page 24

For a summary of all front end features, refer to “Elements Defined in the Front End” on page 27.

To create a front end, use the **Search > Search Features > Front Ends** page. For complete information about the **Front Ends** page, click **Admin Console Help > Search > Search Features > Front Ends** in the Admin Console.

Using Collections with Front Ends

A collection is a subset of the complete search index. A collection lets end users:

- Search over a specific part of the index
- Narrow a search
- Get relevant results more quickly


A collection is analogous to a playlist in media player software. To create a playlist, you define it, add songs in it, and store it. If you have more than one playlist, one song can appear in multiple playlists. You can create a playlist for a specific group of listeners, such as your family.

To create a collection, you define it, add entries to it from the search index, and store it. If you define more than one collection, the same entry can appear in multiple collections. You can define collections for specific end users.

Suppose you define a collection to support end users in Human Resources (HR). This collection contains information that is related only to HR.

You define a collection by URL patterns. All content that matches the URL pattern belongs to the collection. The same content can appear in multiple collections. Search results from a collection have the same relevance ranking as full index searches. Only the content searched differs because it is restricted to the individual collection's content.

You can search multiple collections by using the `site` parameter, as described in the *Search Protocol Reference*. When the search appliance displays results for a query against multiple collections, it does not group the results by collection.

 Collections are independent of front ends. However, you can use a custom front end with a specific collection to help improve searches and enhance results. Continuing the previous example, in addition to an HR collection, suppose you have also configured an HR front end and associated it with the HR collection. When end users search using the HR front end, the search is restricted to the HR collection.

Another example is to configure a front end for customers and associate it with a customer collection. The customer collection contains only public, non-secure information, such as descriptions of products and services. When customers search using this front end, the search is restricted to information in the customer collection.

There are two ways to associate a collection with a front end:

- Add an element to the search page, such as a select list or radio button that enables end users to select collections for their searches
- Use query parameters to bind a collection to a front end, then mask the query parameters using a proxy server

For details about adding an element to a search page to enable searching by collection, refer to *Changing the User Interface*.

To create a collection, use the **Index > Collections** page in the Admin Console. For information about using this page to create a collection, see **Admin Console Help > Index > Collections**.

Note: When defining collections that include content from a content connector, the URL patterns refer to the internal URL representation (the “googleconnector:” patterns), rather than the displayed URLs. For more information about the content connectors, refer to the Documentation for the Google Search Appliance Connectors.

Maximum Number of Front Ends and Collections

Do not create more than the recommended number of front ends or collections for a search appliance:

- For front ends, the maximum number is 200
- For collections, the maximum number is 200

Google recommends that you keep the number of front ends and collections as low as possible. If you create more than 200 front ends or 200 collections, importing or exporting a configuration might be slow. If Google for Work Support determines that this performance issue caused by too many front ends or collections, they might require you to reduce their numbers.

Improving Searches

One way to improve an end user's search experience is to provide feedback that helps her find information that she might otherwise miss. One form of feedback that Google provides by default is spelling suggestions. This is a built-in feature of the Google Search Appliance that works the same as it does on Google.com.

When an end user types a search term that seems to be a misspelling, the search appliance responds with a spelling suggestion. For example, if an end user types "aduio," the search appliance responds with the following spelling suggestion:

Did you mean: [audio](#)

This feedback gives the user an opportunity to:

- Run the search again
- Get relevant results

The choice of clicking a spelling suggestion is completely up to the end user.


The spell checker supports Dutch, US English, Brazilian Portuguese, French, Italian, German, and Spanish. (You can change the supported languages by installing and activating a different language bundle, see "Changing Languages for Query Expansion and Spelling Suggestions" on page 66.) For information about how the search appliance makes spelling suggestions for supported languages, see "How Does the Search Appliance Make Spelling Suggestions?" in *Search Appliance Internationalization*.

You cannot edit the search appliance's spelling dictionary. However, the search appliance offers other features that improve searches. The following table gives an overview of these features.

Feature	Described in
Related queries	"Suggesting Alternative Search Terms" on page 16
KeyMatch	"Guiding End Users to Specific URLs" on page 17
Dynamic results clusters	"Narrowing Searches" on page 18
Dynamic navigation	
Query expansion	"Widening Searches" on page 19

Spelling suggestions are not returned when special parameters such as `as_sitesearch`, `inurl:`, `intitle:`, and the like are used in a query.

Suggesting Alternative Search Terms

 As shown in Addressing Diverse End Users, the Google Search Appliance can suggest alternative search terms ("related queries") based on an end user's original search terms. For example, an end user searches using the term "Iwo To," which is the current name for "Iwo Jima." Searching for "Iwo To" returns results indexed under "Iwo To," but misses the results indexed under "Iwo Jima." However, the search appliance returns the following related query at the top of the search results:

You could also try: [Iwo Jima](#)

When the user clicks "Iwo Jima," the search appliance runs the search again and returns additional results. The choice of whether to click a related query is completely up to the end user.

In addition to suggesting alternative search terms, related queries can also suggest more specific keyword searches, such as your own product names. For example, an end user searches for “turntables” and, using related queries, the search appliance returns specific product names, for example:

You could also try: [Acme Portable Turntable](#)


Unlike spelling suggestions, related queries are not available by default. You can create them for a specific front end by associating a search term to a related query. In the previous example:

- The search term is “turntables”
- The related query is “Acme Portable Turntable”

To create a related query, use the **Search > Search Features > Front Ends > Related Queries** page. For complete information about the **Related Queries** page, click **Admin Console Help > Search > Search Features > Front Ends > Related Queries** in the Admin Console.

For more information about related queries, refer to “Using Related Queries to Suggest Alternative Searches” on page 45.

Guiding End Users to Specific URLs

 You can also customize search feedback by guiding end users to specific URLs using KeyMatches. KeyMatches are preferential search results, or recommended links, that appear at the top of the search results. Like related queries, KeyMatches are results that are based on the end user’s original search terms.

For example, an end user searches with the term “401K.” The search appliance returns the following KeyMatch at the top of the search results:



A KeyMatch gives an end user an opportunity to navigate immediately to the recommended document. This means that the end user spends less time searching for documents and more time looking at them. As with related queries, the choice of clicking a KeyMatch is completely up to the end user.

KeyMatches let you promote specific documents higher in the search results, even when documents are not indexed or have low relevance. Because a KeyMatch is specific to a front end, it can be aimed at specific types of end users.


KeyMatches are not available by default. You can create them for a specific front end by associating a search term to a specific URL and specifying a title for the match. In the previous example:

- The search term is “401K”
- The URL is <http://www.cosmoaud.com/hr/retirements.html>
- The title is “Just Published: New Retirement Options”

To create a KeyMatch, use the **Search > Search Features > Front Ends > KeyMatch** page. For complete information about the **KeyMatch** page, click **Admin Console Help > Search > Search Features > Front Ends > KeyMatch** in the Admin Console.

For more information about KeyMatch, refer to “Using KeyMatches to Guide Users to URLs” on page 43.

Narrowing Searches

 For some search terms, the Google Search Appliance can narrow searches by providing dynamically formed subcategories ("dynamic result clusters") based on the results of each search query. Each subcategory groups similar documents together. Instead of reading through results to understand the results, end users can browse a subcategory.

By clicking a subcategory link, an end user can

- Refine the original search query
- Get more accurate results than from the original search term alone

For example, suppose an end user who looking for information about the history of the vikings. He searches for this information using the term "vikings." A dynamic result cluster appears with the results, as shown in the following example.

narrow your search

[vikings history](#)
[vikings schedule](#)
[vikings football team](#)
[vikings weapons](#)
[vikings in america](#)
[viking update](#)
[viking names](#)
[viking clothing](#)

The subcategories group the results into meaningful clusters, enabling the user to focus on the history of vikings while ignoring irrelevant information.

By default, dynamic result clusters is not enabled for each front end. To modify settings for dynamic result clusters, use the Search results section of the Page Layout Helper. The Page Layout Helper is on the **Search > Search Features > Front Ends > Output Format** page. For complete information about the Page Layout Helper, click **Admin Console Help > Search > Search Features > Front Ends > Output Format** in the Admin Console. You can also modify settings for dynamic result clusters using the eXtensible Stylesheet Language Transformations (XSLT) stylesheet. For access to the XSLT stylesheet, click **Admin Console Help > Search > Search Features > Front Ends > Output Format** in the Admin Console

For more information about dynamic result clusters, refer to "Using Dynamic Navigation to Help Users Explore Results" on page 49.

Dynamic navigation is a feature that provides another method for helping users to refine their searches. With dynamic navigation, options that are based on metadata in your corpus appear along with the search results. Options list document counts for each value.

Suppose your corpus includes metadata for each department, for example, `<META NAME="dept" CONTENT="Sales">`. When you configure dynamic navigation for this metadata, the following Navigate options might appear with search results:

Department

Engineering (33)
Marketing (35)
Sales (27)

For more information about dynamic navigation, refer to "Using Dynamic Navigation to Help Users Explore Results" on page 49.

Widening Searches



Without any input from the end user other than a search term, the Google Search Appliance can expand a query by adding synonymous terms. This helps end users get results that they would otherwise miss. The feature is called "query expansion."

For example, an end user searches on the term "documentation," and the search appliance returns the most relevant results that contain the keyword "documentation." However, the end user misses results that contain alternative terms, such as "guide" and "manual." If the search term "documentation" is expanded to include "guide," "guides," "manual," and "manuals," the search is wider and returns an increased number of relevant results.

Google dictionaries of synonyms for English, French, Italian, German, Spanish, and Portuguese are built into the search appliance. Whenever an end user enters a search query that matches a synonym in one of these languages, the term is expanded. For information about how the search appliance expands queries in supported languages, see "Which Languages can use Query Expansion?" in *Search Appliance Internationalization*.

However, you can create and upload custom synonym lists to improve search quality. You can also create and upload a blacklist or stopword file. A blacklist is a file that contains terms that should not be expanded. A stopword file contains search terms that are ignored by the search appliance. Take note that if a stopword is the only keyword in a query, it is not ignored.

To widen searches:

1. Upload custom synonym lists, blacklists, and stopword files.
2. Set the Query Expansion policy for a specific front end.

An uploaded synonym list, blacklist, or stopword file is applied to a front end when you set the query expansion policy for the front end.

To upload synonyms, blacklist terms, or stopwords, use the **Search > Search Features > Query Settings** page. For complete information about the **Query Settings** page, click **Admin Console Help > Search > Search Features > Query Settings** in the Admin Console.

Query expansion is OFF by default and should be turned on to FULL in each front end to maximize the potential relevancy of the results. To set the Query Expansion policy for a front end, use the **Search > Search Features > Front Ends > Filters** page. For complete information about the **Filters** page, click **Admin Console Help > Search > Search Features > Front Ends > Filters** in the Admin Console.

For more information about query expansion, refer to "Using Query Expansion to Widen Searches" on page 60.

You can also widen searches by enabling wildcard search. This feature enables your users to search by entering a word pattern rather than the exact spelling of a term. To enable wildcard search, use the **Search > Search Features > Front Ends > Filters** page in the Admin Console.

For more information, see "Enabling Wildcard Search" on page 82.

Enhancing Search Results

Without any administrator intervention, the Google Search Appliance enhances search results by performing the following actions:

- Sorting the results by relevance—The search appliance uses over 100 different algorithms to sort results by relevance intelligently and dynamically.

- Filtering duplicate snippets—If multiple documents contain identical titles, as well as the same information in their snippets, only the most relevant document of that set appears in the results.
- Filtering duplicate directories—If there are many results in a single web directory, only the two most relevant results for the directory appear. This feature is also known as “directory crowding.”

In addition to these built-in features, the search appliance offers other features that enhance search results. The following table gives an overview of these features.

Feature	Described in
OneBox modules	“Integrating Real-Time Data” on page 20
Expert search	“Providing Expert Profiles with Search Results” on page 21
User results	“Giving Users the Ability to Add Results” on page 21
Document previews	“Showing Document Previews” on page 21
Filters	“Refining Search Results” on page 22
Remove URLs	“Removing Specific URLs from Results” on page 22
Results biasing	“Influencing Results Ranking” on page 23
Alerts	“Enabling Alerts” on page 24

Integrating Real-Time Data

In some instances, the most relevant result for a search query is real-time, structured data. This type of data does not usually reside in the search index because it would be obsolete before it could be indexed. For example, an end-user searches on “expense reports mlock.” Specially formatted real-time data showing current expense reports for the name “mlock” appears at the top of the search results, as illustrated in the following figure.



This type of result is served by a “OneBox module.” Instead of going to multiple sources for information, the search appliance executes instructions to get the result from a OneBox provider. Like KeyMatches, OneBox modules represent preferential results that enable end users to receive relevant content without paging through other search results.


A OneBox module is returned when an end user's search term matches a “trigger” term. In the previous example:

- The trigger is “expense reports”
- The search term that matches the trigger is “mlock”

Each trigger can have up to four OneBox module results. Other examples of this type of data include current flight information and tracking information for shipping orders.

The search appliance supports two types of OneBox modules:

- Internal—Provides real-time access to data from a collection on the search appliance. The OneBox will only trigger if the search is performed on a collection that is different from the collection specified in the onebox configuration.
- External—Provides real-time access to data from an external source, such as an application or database.

 A OneBox module that has been integrated with the search appliance can be used with any of the front ends on the search appliance. A front end can use an unlimited number of OneBox modules. To add a OneBox Module:

1. Integrate the OneBox Module into the search appliance.
2. Select a OneBox Module for use in a specific front end.


To integrate structured data in search results, use the **Content Sources > OneBox Modules** page. For complete information about the **OneBox Modules** page, click **Admin Console Help > Content Sources > OneBox Modules** in the Admin Console.

To select a OneBox module to use with a front end use **Search > Search Features > Front Ends > OneBox Modules** page.

For detailed information about developing OneBox modules, refer to the following documents:

- *Google OneBox for Enterprise Developer's Guide*
- *Google OneBox for Enterprise Design Principles*

Providing Expert Profiles with Search Results

 Users might want to search for experts on specific topics in your organization. You can provide this capability by configuring expert search. With expert search, when a user searches on a keyword, such as "hiring" a list of hiring experts appears in a sidebar next to search results. You might also provide an option for the user to click a link to view a more detailed list of experts on a separate page.

To configure expert search, use the **Search > Search Features > Expert Search** page.

For more information about expert search, see "Providing Expert Search for Users" on page 53.

Giving Users the Ability to Add Results

You can give users the ability to add search results for certain keyword searches. User results appear for the specified keyword searches on the search results page of a specific front end.

To add a user results configuration, use the **Search > Search Features > User Results** page.

For more information, see "Providing User Results" on page 77.


Showing Document Previews

For documents in Microsoft Word (doc, docx), Microsoft PowerPoint (ppt) and Adobe pdf formats, the search appliance can show preview images in search results. To view a document preview, the user hovers the pointer over a magnifying glass icon next to the search result.

To show document previews, use the **Search > Search Features > Document Preview Module** page.

For more information about document previews, see “Providing Document Previews” on page 81

Refining Search Results

 Enterprise content often contains information that is not appropriate for serving to all end users. For example, enterprise content may contain sensitive documents that are appropriate for members of an organization to view, but not for consumers to view. To ensure that the search appliance serves appropriate results to end users, you can create filters that prevent the sensitive data from appearing in search results for a particular front end. In this situation, you would probably create a meta tag filter.

The search appliance includes built-in filters for:

- Duplicate snippets
- Duplicate directories

These filters apply to the entire search index. For an overview of these filters, refer to “Built-In Elements” on page 28. You can also create filters for specific front ends results based on:

- Language
- Domain
- File type


Unlike Query Expansion and OneBox Modules, filtering is not based on keywords in the search query. The search appliance filters all results for all end users of a particular front end.

To create filters for a front end, use the **Search > Search Features > Front Ends > Filters** page. For complete information about the **Filters** page, click **Admin Console Help > Search > Search Features > Front Ends > Filters** in the Admin Console.

For more information about filters, refer to “Using Filters to Restrict Search Results” on page 56.

Removing Specific URLs from Results

Occasionally, a search index contains URLs that the search appliance should not serve to some or all end users. For example, an administrator has added jump pages, which are just lists of URLs, to the enterprise content for the purpose of getting unlinked URLs into the search index. The administrator wants to keep these jump pages in the search index, but does not want to serve the jump page URLs to end users. Other examples of URLs that administrators might want to prevent serving include URLs that are out-of-date and URLs that contain sensitive data.

 You can prevent the search appliance from serving URLs that match specific patterns. Take note that specifying a long list of patterns can cause increased latency at serve time. Because you remove URLs from results for a front end, you can remove them for specific types of end users.

To specify URLs to remove from results for specific front ends in the Admin Console, use the **Search > Search Features > Front Ends > Remove URLs** page. For complete information about the **Remove URLs** page, click **Admin Console Help > Search > Search Features > Front Ends > Remove URLs** in the Admin Console.

For more information about removing URLs, refer to “Removing URLs from Search Results” on page 59.

Removing URLs from the Search Index

The remove URLs feature affects results only. It does not remove URLs from the search index. To remove URLs from the search index, enter them in the **Do Not Follow Patterns** section on the **Content Sources > Web Crawl > Start and Block URLs** page in the Admin Console. For more information about removing URLs from the index, refer to *Administering Crawl*.

Influencing Results Ranking

On the search appliance, 100 algorithms are used to determine the sort order of the results that are returned. However, you may want to have some influence over how the search appliance ranks results.

The search appliance supports three ways to influence results ranking:

- Source biasing—Lets you influence the way that the search appliance ranks search results based on the URLs in the result.
- Date biasing—Lets you specify the age considerations that should influence a document's score.
- Metadata and entity biasing—Lets you influence the way that the search appliance ranks search results based on metadata or entities in or associated with the result.



Because result biasing is specific to a front end, it can be aimed at specific types of end users.

To influence search appliance rankings, use a result biasing policy. A default result biasing policy (default_policy) is built into the search appliance. You can use default_policy, or create one or more custom result biasing policies. For a result biasing policy to affect search results, you must select it for use with a front end.


To set up result biasing:

1. Create a result biasing policy by using the **Search > Search Features > Result Biasing** page in the Admin Console.
2. Configure the result biasing policy by selecting features for influencing the score of a document by using the **Search > Search Features > Result Biasing > edit** page in the Admin Console.
3. Enable the result biasing policy by selecting it for use with a front end by using the **Search > Search Features > Front Ends > Filters** page in the Admin Console.

For complete information about using these pages, refer to the **Admin Console Help** in the Admin Console.

For more information about result biasing, refer to “Using Result Biasing to Influence Result Ranking” on page 71.

Enabling Alerts

 Another way of enhancing search results is by enabling users to monitor topics of interest by receiving search results for these topics in email messages. You can enable users to monitor topics this way by providing alerts. Alerts only work with public, non-secure results.


To provide alerts for users, you must:

1. Configure an authentication mechanism for the search appliance to use to authenticate the user.
2. Enable alerts for the search appliance by using the **Index > Alerts** page in the Admin Console.
3. Show the **My Alerts** link for a specific front end by using the **Search > Search Features > Front Ends > Output Format** page.

For complete information about using these pages, refer to the **Admin Console Help** in the Admin Console.

For more information about enabling alerts, refer to “Providing Alerts for End Users” on page 75.

Changing the User Interface

 As shown in “Focusing on End Users” on page 9, the Google Search Appliance user interface consists of a search page and results page. A user interface is associated with a specific front end. The search and results page examples in “Starting with a Basic Search Experience” on page 9 illustrate the user interface for the default front end. This user interface includes Google-specific elements, such as:

- Google logo
- Search Google button label

The user interface for the default front end can also be used with any other front ends that you create for a search appliance.


The search appliance allows maximum flexibility for customizing the appearance of the search and results pages. Often, an organization creates its own visual identity using elements such as:

- Logo
- Colors
- Font faces

You can apply these elements to the user interface. The search and results page examples in Customizing the Basic Search Experience illustrate this type of change. In this example, the administrator adds the company logo and changed the font for the user interface.

When a search appliance has multiple front ends:

- Each front end can have its own user interface
- Each user interface can be customized for a particular type of end user, as illustrated in “Creating Multiple Search Experiences” on page 11

 In this example, the administrator customized the results listings, the KeyMatches, and the language of the user interface for various type of end users.

The appearance of the results page is created when the search appliance's XSL transformation engine applies an XSLT stylesheet to search results in HTML. For an overview of this process, refer to “Search Experience Background” on page 29.

The Page Layout Helper is a tool that you can use to make simple changes to the user interface. Without any knowledge of XSLT, you can use the Page Layout Helper to customize the appearance of:

- Global attributes—Custom logo, font face, custom header, custom footer, analytics account
- Search box—Length, button text or image, menu for searching by collection, radio button for searching public or public and secure content
- Search results—Logo, Advanced search link, search tips link, search box (top), page divider (search information), Previous/Next link, Sort by Date/Relevance link, Dynamic result clusters, Snippet, URL link, page size, modified date, cache link, page footer, search box (bottom)

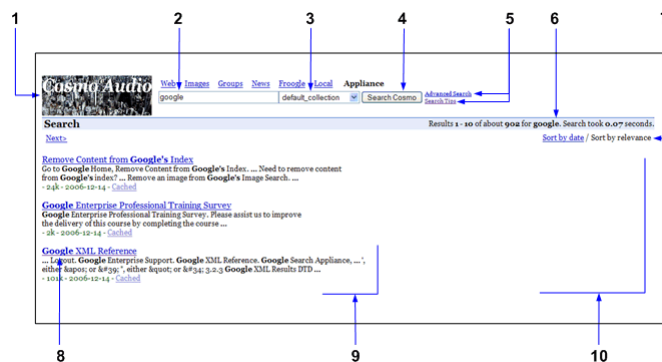
The Page Layout Helper is on the **Search > Search Features > Front Ends > Output Format** page. For complete information about the Page Layout Helper, click **Admin Console Help > Search > Search Features > Front Ends > Output Format** in the Admin Console.

With knowledge of XSLT, you can make even more extensive changes to the user interface, such as changing labels and colors for

- Spelling suggestions
- Related queries
- KeyMatches
- Results listings

You can edit the XSLT stylesheet using the search appliance XSLT Stylesheet Editor on the **Search > Search Features > Front Ends > Output Format** page, or another editor outside the search appliance. If you develop an XSLT stylesheet outside the search appliance, you can upload it to the search appliance using the **Search > Search Features > Front Ends > Output Format** page.

The following figure illustrates some of the user interface elements that you can customize.



For descriptions of changes that you can make to each of the user interface elements, refer to the key number in the following table.

Key	Description
1	Change the logo.
2	Customize the search box.
3	Show or hide a menu to search by collection.
4	Customize the search button.
5	Show or hide search options.
6	Customize the separation bar.
7	Show or hide sort options.
8	Customize the appearance of keyword matches.
9	Customize the contents of results listings.
10	Change the font faces, colors, and sizes of text on the page.

For details about modifying the user interface, refer to “Customizing the User Interface” on page 92.

Where Is the Search Experience Created?

As indicated in the previous sections, you control most aspects of the search experience using the search appliance Admin Console. Many of the Admin Console pages that you use to create the search experience are front end pages. However, some elements of the search experience are created using different pages in the Admin Console.

This section provides overview tables of the different pages that you can use to create the search experience. This section also includes an overview of the search appliance's built-in elements.

Elements Defined in the Front End

The following table provides an overview of search experience elements defined using Admin Console front end pages.

Element	Defined Using Admin Console Page
Front end	Search > Search Features > Front Ends page
Page format	Search > Search Features > Front Ends > Output Format page
Logo	Search > Search Features > Front Ends > Output Format page, Page Layout Helper or XSLT Stylesheet Editor
Font face and color	
Results page header	
Search box	
Search button	
Separation bar	
Navigation bars	
Dynamic result clusters	
Translation	
Document previews	
Advanced search reporting	
Sort options	Search > Search Features > Front Ends > Output Format page, XSLT Stylesheet Editor
Spelling suggestions	
Show/Hide secure results radio button	
KeyMatch	Search > Search Features > Front Ends > KeyMatch page
Related queries	Search > Search Features > Front Ends > Related Queries page
Query expansion policy	Search > Search Features > Front Ends > Filters page
Filters	Search > Search Features > Front Ends > Filters page
Select a result biasing policy	Search > Search Features > Front Ends > Filters page
Remove URLs	Search > Search Features > Front Ends > Remove URLs page
Select a OneBox module to use with a front end	Search > Search Features > Front Ends > OneBox Modules page

Elements Defined on Other Admin Console Pages

The following table provides an overview of search experience elements defined on Admin Console pages other than front end pages. While these elements are defined externally to the front end, each one must be enabled using a front end page (as listed in the previous section).

Element	Defined Using Admin Console Page
Query expansion	Search > Search Features > Query Settings page
OneBox modules	Content Sources > OneBox Modules page
Result biasing policy	Search > Search Features > Result Biasing page
Result biasing policy configuration	Search > Search Features > Result Biasing > edit page
Alerts	Index > Alerts page
Expert search	Search > Search Features > Expert Search page
User results	Search > Search Features > User Results page

Built-In Elements

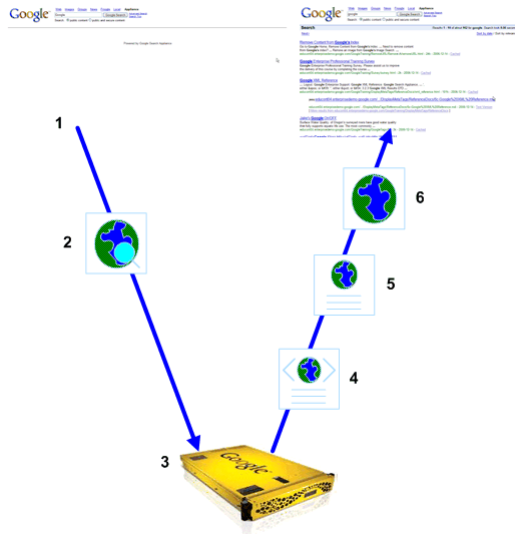
The following table provides an overview of elements that are available by default.

Element	Comments
Automatic filtering: duplicate snippet filter	If multiple documents contain identical titles, as well as the same information in their snippets, only the most relevant document of that set is displayed in the results. Default: enabled. When a search filter is enabled and removes some results, the search results output indicates that results were filtered.
Automatic filtering: duplicate directory filter	If there are many results in a single web directory, then only the two most relevant results for the directory are displayed. An output flag indicates that more results are available from that directory. Default: enabled. When a search filter is enabled and removes some results, the search results output indicates that results were filtered.
Automatic language filter: Limit search to a specified language, as determined by the majority language used in the web document body	Automatic language filter. Possible to override using the <code>lr</code> query parameter and Boolean operators.
Number of search results	By default, ten results appear. Possible to override with the <code>num</code> query parameter.
Sorting results based on relevance.	By default, the search appliance uses hypertext-matching analysis and PageRank technologies to sort results by relevance.

For information about the `lr` and `num` query parameters, refer to the *Search Protocol Reference*.

Search Experience Background

The Google Search Appliance response to a search query may appear to be instantaneous to an end user. However, in the background, each search query follows a six-step search process. The following diagram provides an overview of the search process.



The numbers in the diagram refer to the following steps in the process:

1. The end user enters a search query using a search page.
2. The Web browser converts the search query into a URL.
3. The search appliance receives the search query and executes it.
4. The search appliance returns search results in XML.
5. The search appliance applies an XSLT stylesheet to the XML results and creates the search results page in HTML.
6. The Web browser presents the search results page to the end user.

Entering a Search Query

An end user enters a search query into the search box. An end user can select other options on the search page by using check boxes, pull-down menus, or radio buttons. In the HTML code for the search page, these end-user input fields appear within `<form>` tags.

Converting the Search Query HTML to a URL

After an end-user clicks Search, the Web browser converts the fields within the `<form>` tags of the HTML page into a URL. This URL is sent to the search appliance as part of an inbound HTTP request message. The URL includes:

- The search appliance name
- An HTTP GET method

- Attributes that specify input elements, such as a search box and search button
- The search query, which is made up of several pairs of search appliance query parameters and values.

For information about search appliance query parameters and values, refer to the *Search Protocol Reference*.

Executing the Search

The Google Search Appliance receives the URL and uses the values in it to process the request and retrieve the results. The search appliance may perform additional actions when executing the search. Search terms and front end settings determine these additional actions. The following table lists front end settings and the actions they cause the search appliance to take.

Front End Setting	Condition	Action
The query expansion policy is set to standard, local, or full	The search term matches a synonym	Expand the query to include all synonyms
The query expansion policy is set to standard, local, or full	The search term matches a blacklist synonym	Do not search on the keyword that matches the synonym.
A OneBox module has been integrated in the front end	The search term matches a trigger	Return the OneBox module at the top of the search results.
The front end includes a KeyMatch	The search term matches a KeyMatch	Return the URL for the KeyMatch at the top of the search results.
The front end includes a related query	The search term matches a related query	Return the related query at the top of the search results.
The front end includes a domain filter	Applies to all searches	Search only for results in the specified domain.
The front end includes a language filter		Search only for results in the specified language.
The front end includes a file type filter		Search only for results with the specified file type.
The front end includes a meta tag filter		Search only for results that match the meta tag value.
The front end includes dynamic result clusters	The search term matches a dynamic result cluster	Return subcategories in the search results.
Result biasing policy is selected for the front end	The search returns results that match the result biasing policy configuration	Recalculate the document's ranking and display it in the results.
Remove URL patterns are identified in the front end	The search returns results that match a remove URL pattern	Remove matching URLs from the search results.

Returning Search Results as XML

The Google Search Appliance returns search results in standard XML in the body of an outbound HTTP response message. The body of the outbound HTTP response message includes the search results in XML. While it is not usual to return results to the end user in XML format, it is possible.

For information about returning XML results to end users, refer to the *Search Protocol Reference*.

Applying an XSLT Stylesheet to the XML Results and Create HTML Output

Before the Google Search Appliance displays the results set to the user, the search appliance applies the XSLT stylesheet to the XML results. The XSLT stylesheet contains instructions on how to format the results and results page in HTML. You can edit this XSLT stylesheet using the Page Layout Helper or XSLT Stylesheet Editor, which are briefly described in “Changing the User Interface.”

Presenting Search Results to the End User

The process ends when the Web browser presents the search results to the end user on an HTML results page.

Chapter 2

Best Practices

The Google Search Appliance has features that enable system administrators to enhance the search experience for end users. This chapter describes how system administrators can follow best practices to improve end-users searches.

Search Experience Best Practices

A Google Search Appliance can begin serving relevant results as soon as it has an index. By serving relevant results, the search appliance ensures a positive search experience for your users. However, you, as a search appliance administrator, can use the best practices and search appliance features described in this document to personalize or enhance the search experience for users.

Checklist of Best Practices

Use the following table as a best practices checklist. For each best practice, the table identifies the search appliance feature or resource to use, with a reference to the relevant section in this document.

Best Practice	Feature or Resource to Use	Reference
Export and analyze data about user clicks	Advanced search reporting	"Gathering Information about the Search Experience" on page 36
Enable end users to create email alerts for topics of interest	Alerts	"Providing Alerts for End Users" on page 75
Prevent click-jacking web attacks	Click-jacking defense	"Enabling/Disabling Click-Jacking Defense" on page 70
Divide the search index into meaningful partitions	Collections	"Segmenting Data in the Search Index" on page 86
Show thumbnail images of documents in search results	Document preview module	"Providing Document Previews" on page 81
Help user explore search results by using specific metadata attributes	Dynamic navigation	"Using Dynamic Navigation to Help Users Explore Results" on page 49
Allow users to choose results by topic	Dynamic result clusters	"Using Dynamic Result Clusters to Narrow Searches" on page 47

Best Practice	Feature or Resource to Use	Reference
Return search results from Google Apps	Content Sources > Google Apps	<i>Integrating with Google Apps</i>
Enable searching for experts in your organization	Expert search	“Providing Expert Search for Users” on page 53
Restrict search results by domain name, language, file type, or meta tag	Filters	“Using Filters to Restrict Search Results” on page 56
Take advantage of search appliance expertise	Google for Work and Google Groups	“Exchanging Information on Google Groups” on page 90
Update to the most recent software version	Google for Work Support site	“Updating Your Search Appliance” on page 91
Eliminate duplicate search results and provide the most relevant set of diverse results	Host crowding	“Experimenting with Host Crowding Options” on page 56
Present recommended links above search results	KeyMatches	“Using KeyMatches to Guide Users to URLs” on page 43
Enable query expansion and spelling suggestions for a set of languages	Language bundles	“Changing Languages for Query Expansion and Spelling Suggestions” on page 66
Add structured, real-time application data to search results	OneBox modules	“Using OneBox Modules to Integrate Structured Results” on page 70
Create a user interface that focuses on your users	Page Layout Helper or XSLT Stylesheet Editor	“Customizing the User Interface” on page 92
Return user profiles along with ranked search results	People search (deprecated)	“Using People Search” on page 54
Provide your own synonyms for use in query expansion	Query expansion	“Using Query Expansion to Widen Searches” on page 60
Enable search queries to auto-complete and query suggestions to appear as a user types in the search box.	Query suggestions	“Providing Query Suggestions” on page 79
Present alternative search terms above search results	Related queries	“Using Related Queries to Suggest Alternative Searches” on page 45
Prevent specific URLs from appearing in search results	Remove URLs	“Removing URLs from Search Results” on page 59
Influence the order of documents in search results	Result biasing	“Using Result Biasing to Influence Result Ranking” on page 71
Identify problematic query topics for related queries, KeyMatches, and query expansion synonyms	Search report	“Analyzing Searches that Do Not Return Results” on page 84
Obtain user responses about the search experience	Surveys	“Conducting a User Satisfaction Survey” on page 89
Translate search results into the user’s language in real time	Translation	“Enabling Translation of Search Results” on page 66

Best Practice	Feature or Resource to Use	Reference
Enable users to add search results for certain keywords in a specific front end	User results	“Providing User Results” on page 77
Allow users to give you feedback	User interface	“Adding a Feedback Mechanism for Users” on page 88
Optimize the response time of results	Various features	“Optimizing the Speed of Results” on page 83
Remove obsolete or non-essential content from the search index	Whitelist URLs or Blacklist URLs	“Keeping the Search Index Clean” on page 86
Enable your users to search by entering a word pattern rather than the exact spelling of a term.	Wildcard search	“Enabling Wildcard Search” on page 82

Personalizing the Search Experience

The Google Search Appliance enables you to personalize the search experience. Rather than providing one centralized search experience for everyone, you can provide different search experiences for different groups of users. Each personalized search experience is based on the interests, roles, departments, locations, or languages of the user group.

For example, suppose your Google Search Appliance provides search for internal users from different organizations. Users often search for “acme widgets,” but they are not all searching for the same results. More typically, when searching for acme widgets:

- Engineering staff is searching for design documents and status information
- Sales staff is searching for sales forecasts and reports
- Customer support is searching for support metrics and update information


With a centralized search experience, some users may find what they are looking for at the top of the results listings while other users might have to view several results before finding what they are looking for. With a personalized search experience:

- Each group of users has a unique search experience where results are ranked according to their interests
- Users find what they are looking for at the top of the search results


Users do not need to take any action to have a personalized search experience. The search appliance automatically presents a personalized search experience based on changes that you, as the system administrator, have implemented using search appliance features, including front ends (see “The Relationship Between the Search Experience and Front Ends” on page 35), KeyMatches (see “Using KeyMatches to Guide Users to URLs” on page 43), and source biasing (see “Using Source Biasing” on page 72).

The first step to personalizing the search experience is gathering information about the search experience (see “Gathering Information about the Search Experience” on page 36) as it is currently deployed on the Google Search Appliance. After gathering this information, you can analyze it and decide:

- Who needs a personalized search experience
- How to personalize the search experience for each target group
- Which search appliance features you should use to create personalized search experiences

 In this document, descriptions of features that you can use to personalize the search experience are marked with the following personalization icon.

The Relationship Between the Search Experience and Front Ends

 Several of the Google Search Appliance features described in this document are implemented in specific front ends. A front end is a framework that manages the search experience for specific types of end users. Features implemented in front ends include:

- KeyMatches (see “Using KeyMatches to Guide Users to URLs” on page 43)
- Related queries (see “Using Related Queries to Suggest Alternative Searches” on page 45)
- Dynamic result clusters (see “Using Dynamic Result Clusters to Narrow Searches” on page 47)
- Dynamic navigation (see “Using Dynamic Navigation to Help Users Explore Results” on page 49)
- Filters (see “Using Filters to Restrict Search Results” on page 56)
- Remove URLs (see “Removing URLs from Search Results” on page 59)
- Query expansion policy (see “Setting the Query Expansion Policy for a Front End” on page 65)
- Result biasing (see “Working with Result Biasing” on page 72)
- My Alerts link (see “Providing Alerts for End Users” on page 75)

The search appliance has a default front end that uses default settings for the elements that influence search quality. You can use the default front end without changing any settings.

If you plan on changing anything in a front end, it is recommended that you create at least one front end to customize. For information about this task, refer to “Creating a Front End” on page 94.


Creating Multiple Front Ends

For one Google Search Appliance, you can create multiple front ends, each with its own customized settings. For example:

- You might create multiple front ends to serve different results to users in various regions, worldwide. For users in a specific country, you might apply a language filter. This example is described in “Restricting Search Results by Language” on page 58.
- You might use a customized front end with a specific collection to improve searches and enhance results. For information about collections, refer to “Using Collections with Front Ends” on page 14 and “Segmenting Data in the Search Index” on page 86.
- You might modify the appearance of search and results pages. For information about changing the appearance of a front end, refer to Customizing the User Interface.

For more information about front ends, refer to “Managing the Search Experience” on page 14.

Gathering Information about the Search Experience

 Before you can effectively personalize the search experience, you need to have knowledge about your end users, such as their roles, functional groups, locations, what they are searching for, and whether they are finding it or not. The Google Search Appliance enables you to gather information about user clicks. From this information, you can gain knowledge about end users and their interactions with search results. Information about user clicks can tell you:

- If users are finding what they're searching for
- If groups of users are searching for the same information
- If certain URLs are harder for users to find than others

By analyzing user clicks, you can identify ways to personalize and improve the search experience. For example, suppose information about user clicks shows that a users in range of IP addresses are all searching for information about project Malta. None of the users are finding a satisfactory URL at the top of the search results. Some of the users are finding a satisfactory URL on page 3 of the results. However, many users are abandoning the search after viewing page 2 of the results.

The range of IP addresses tells you that the users who are searching for the results are all Engineers in the U.S. who are working on project Malta. You might create a front end for this group of users. For information about this task, refer to “Creating a Front End” on page 94.

For this front end, you might promote project Malta URLs to the top of the search results using KeyMatches (see “Using KeyMatches to Guide Users to URLs” on page 43). Another alternative for promoting these URLs is to use source biasing (see “Using Source Biasing” on page 72) to move the result up in the listings.

Some time after you deploy the front end for Engineers in the U.S., you can gather more information about user clicks to find out if the personalization that you deployed in the front end is successfully improving the search experience. You can also use this information to make other changes to improve and personalize the search experience.

Viewing Detailed Data about User Clicks

The Google Search Appliance feature that enables you to gather information about user clicks is advanced search reporting (ASR). When advanced search reporting is enabled in the Admin Console (see “Enabling Advanced Search Reporting for a Front End” on page 42), you can export an advanced search report (see “Exporting an Advanced Search Report” on page 42). Each entry in an advanced search report represents a single user click or other action, such as page load, in the search appliance user interface.

When you enable advanced search reporting, the search appliance uses its automatic self-learning scorer (see “Using the Automatic Self-Learning Scorer” on page 42), which fine tunes relevance and scoring for results.

Values in Advanced Search Reports

An entry is composed of comma-separated values. The following table describes each value in an advanced search report.

Value	Description	Example
Click time	Time of the click in 100ths of a second since midnight on January 1, 1970.	125001323979
IP address	IP address of the user who clicked.	172.18.75.121
Session ID	Holding place for session ID, always blank.	
Click type	Type of action, which can be a user click or other action. For more information, refer to “Click Types in Advanced Search Reports” on page 38	c
Click start	The results page of the user click, where 0=1st page of results.	0
Click rank	The rank of the result on the page of the user click, where 1=the 1st result, 2=2nd result, and so on. A smaller number click rank indicates higher user click satisfaction.	1
Click data	Usually blank.	
Query	Search query that returned results.	Google+Search+Appliance
URL	URL of the user click.	http://www.google.com/

The following example shows a complete entry in an advanced search report:

```
1217555170,172.18.75.121,,c,0,1,,Google+Search +Appliance,http://www.google.com/
```

This example shows that on July 31, 2008 at 6PM (1217555170), a user (172.18.75.121) clicked on a result (c) on the first page of results (0). The result was the first result (1) that appeared on the page. The search term that the user entered was Google Search Appliance and the URL of the result was http://www.google.com/.

Click Types in Advanced Search Reports

The following table describes built-in click types that can appear in an advanced search report.

Click Type	Description
advanced	Advanced search link on the search page
advanced_swf	Advanced search for anchor text
c	Search result
cache	Cached document on results page
cluster	Cluster label on results page
db	Database content on results page
desk.groups	Groups link at the top of the search page
desk.images	Images link at the top of the search page
desk.local	Local link at the top of the search page
desk.news	News link at the top of the search page
desk.web	Web link at the top of the search page
help	Search Tips link on the search page
keymatch	Keymatch on results page
load	Load results page
logo	Hyperlinked logo
nav.next	Navigation, next page
nav.page	Navigation, specific page
nav.prev	Navigation, previous page
onebox	OneBox on results page (see “Adding Click Types to OneBox Modules” on page 39)
sitesearch	More results from... link on results page
sort	Sort link on results page
spell	Spelling suggestion
synonym	Related query on results page
OTHER	Any link on the results page that does not have a defined ctype

Optionally, you can also:

- Add custom click types (see “Adding Custom Click Types” on page 39)
- Add click types to OneBox modules (see “Adding Click Types to OneBox Modules” on page 39)

Adding Custom Click Types

You can also add your own custom types to an XSLT stylesheet. Any ctype values that you add remain within your system only and are not tracked by Google. Ensure that any ctype values that you add do not duplicate the built in ctype values.

If you want to track user clicks on elements other than links (such as buttons), refer to the clicklog.js file on the Google Search Appliance. (clicklog_compiled.js is an optimized version of this file that reduces the download size and browser latency.)

Adding Click Types to OneBox Modules

If you have OneBox modules where you want to track user clicks, add the onebox ctype to onebox-default.xsl. To edit onebox-default.xsl, use the **Content Sources > OneBox** Modules page in the Admin Console or an editor of your choice.

Interpreting Detailed Data About User Clicks

The examples in this section show you how to interpret detailed data about user clicks in an advanced search report. The examples omit some values from the each entry in the advanced search report, as indicated by the ellipses (...).

The following example shows a user searching on the term enterprise and finding a result on page 2 of the listings:

1. On a search page, the user enters the search term enterprise and clicks the search button. The results page loads:
`...,172.18.75.121,,load,,0,0,,enterprise`
2. The user does not find satisfactory results on the first results page and clicks Next:
`...,172.18.75.121,,nav.next,0,0,,enterprise...`
3. On the second result page, the user clicks the third result and navigates to `http://www.google.com/`:
`...,172.18.75.121,,c,1,3,,enterprise,http://www.google.com/`

The following example shows a user searching on the term corfu clicking on the tenth result on the first page of listings:

1. On a search page, the user enters the search term corfu and clicks the search button. The results page loads:
`...,174.22.35.109,,load,,0,0,,corfu`
2. On the second result page, the user clicks the third result and navigates to `http://www.corfu.com/`:
`...,174.22.35.109,,c,0,10,,corfu,http://www.corfu.com/`

Using Advanced Search Reporting to Personalize the Search Experience

The following steps outline an effective method for using advanced search reporting to personalize the search experience:

1. Getting baseline measurements of percentage of satisfied queries and average click rank (see “Getting Baseline Measurements” on page 40).
2. Analyzing user clicks to identify areas for improvement in a specific front end (see “Analyzing User Clicks” on page 41).
3. Personalizing the search experience using front ends (see “Personalizing the Search Experience Using Front Ends” on page 41).
4. Measuring the effectiveness of personalization (see “Measuring the Effectiveness of Personalization” on page 41).

Because an advanced search report is a comma-separated values file, you can import it into a spreadsheet where you can sort the values. For example, you might sort an advanced search report by URL to find all user clicks that pertain to a specific URL.

Getting Baseline Measurements

Before you begin personalizing the search experience, identify the initial state of the search experience by taking baseline measurements. You can get baseline measurements by:

1. Generating an advanced search report (see “Generating an Advanced Search Report” on page 41)
2. Calculating the percentage of satisfied queries (see “Calculating the Percentage of Satisfied Queries” on page 40)
3. Calculating the average click rank (see “Calculating the Average Click Rank” on page 40)

As you personalize the search experience, continue to generate advanced search reports that you can compare with baseline measurements.

Calculating the Percentage of Satisfied Queries

Using the information in the report, you can calculate the percentage of satisfied queries. A satisfied query is a search that ends with a user clicking on a result, as indicated in the report by a click type of c. When counting the total number of satisfied clicks only count the first click type of c after the first load.

The formula for calculating percentage of queries that are satisfied is:

$$\text{total number of satisfied clicks} / \text{total number of queries} \times 100 = \text{percentage of satisfied clicks}$$

For example, if the total number of satisfied clicks is 54 out of 90 total queries, the percentage of satisfied queries would be 60%.

Calculating the Average Click Rank

To calculate the average click rank, you must first identify the absolute click rank for an individual entry. Using the click rank alone will not give accurate results because a click rank number does not indicate the results page of the click. For example, when a user clicks on the first result on the first page, the click rank is 1. However, when a user clicks on the first result on the tenth page of results, the click rank is also 1. By using the following formula, you can account for the page of the user click.

The formula for identifying a click rank for an individual entry is:

$$(\text{click start} \times 10) + \text{click rank} = \text{absolute click rank}$$

where click start is the page of the click

The formula for calculating average click rank is:

$$\text{total of absolute click ranks} / \text{number of clicks} = \text{average click rank}$$

For example, if the total of all the click ranks was 255 for 75 clicks, the average click rank would be 3.4.

Analyzing User Clicks

To analyze user clicks, determine how users are searching based on click type and how the search ends.

For example, you can identify top queries by sorting an advanced search report by query. After sorting queries, you might check the click types on the top queries. For satisfied queries with low click rank and unsatisfied queries, you might improve search by creating a KeyMatch, applying source biasing, or developing a OneBox module.

You also might want to identify IP address ranges for most queries. This might indicate that members of a specific group of users, such as the sales department, are all searching for the same information. You might consider personalizing their search experience by creating a collection (see “Segmenting Data in the Search Index” on page 86) and front end (see “Creating a Front End” on page 94) especially for this group.

Personalizing the Search Experience Using Front Ends

Using your analyses of user clicks, you can identify ways to personalize the search experience using front ends. For an overview of ways to personalize the search experience, refer to “Checklist of Best Practices” on page 32.

Throughout this and other chapters of *Creating the Search Experience*, descriptions of features that you can use to personalize the search experience are marked with a personalization icon.

Measuring the Effectiveness of Personalization

After you personalize the search experience, generate another advanced search report and compare with it the baseline measurements (see “Getting Baseline Measurements” on page 40). This comparison enables you to measure the impact that personalization has had on the search experience.

For example, suppose that after you personalize the search experience, the percentage of satisfied user clicks is 80%. This shows a measurable improvement from the baseline of 60%. Because of personalizations, users are more frequently finding relevant information.

Generating an Advanced Search Report

Use the Google Search Appliance Admin Console to generate an advanced search report by:

- Enabling advanced search reporting for a front end (see “Enabling Advanced Search Reporting for a Front End” on page 42)
- Exporting an advanced search report (see “Exporting an Advanced Search Report” on page 42)

Enabling Advanced Search Reporting for a Front End

You can enable or disable advanced search reporting for any front end. To enable advanced search reporting, use the **Search > Search Features > Front Ends > Output Format** page in the Admin Console. For complete information about using the **Output Format** page, refer to **Admin Console Help > Search > Search Features > Front Ends > Output Format - Page Layout Helper** in the Admin Console.

Exporting an Advanced Search Report

After you enable advanced search reporting for a front end, you can export an advanced search report. To export an advanced search report, use the **Reports > Search Logs** page in the Admin Console. For complete information about using the **Search Log** page, click **Admin Console Help > Reports > Search Logs** in the Admin Console.

You can also use the **Reports > Search Reports** page to view the following advanced search report data:

- Rank of selected result
- Page of selected result
- URLs that are popular
- IP addresses of the most frequent users of the system

Using the Automatic Self-Learning Scorer

If you enable advanced search reporting, the search appliance uses its automatic self-learning scorer. This feature automatically analyzes user behavior and the specific links that users click on for specific queries to fine tune relevance and scoring. The search appliance uses advanced statistical regression to determine the statistical significance of user behavior, and adjusts for trust bias (that is, users clicking on the first result solely because it's first). Thus, over time, results become more and more precise without the need of administrator intervention.

The learning-module runs daily, and learns over a 90 day window. It has the capability to forget old user behavior, and learn from new ASR data. It is not possible to remove ASR data without waiting for the 90 day window to pass.


You may allow/disallow the scorer to use the information learned by the learning module. However, even if the scorer is not allowed to use the data gathered by the learning module, no information is lost. As long as ASR is activated, the learning module continues to learn. If the scorer is allowed to use that information at a later point, it immediately begins using everything learned over past days.

To deactivate the automatic self-learning scorer, navigate to the following URL: `http://<yourMachine>:8000/EnterpriseController#a=deActivateAdvancedScoring`

To activate the automatic self-learning scorer, navigate to the following URL: `http://<yourMachine>:8000/EnterpriseController#a=activateAdvancedScoring`

The automatic self-learning scorer is activated by default when you enable advanced search reporting.

Using KeyMatches to Guide Users to URLs

 A document might not be in the search index or might not appear among the highest search results. However, it might be valuable to users as a search result. You can promote such documents in the results by using KeyMatches. A KeyMatch guides users to recommended links, and appear when a user enters a specific search term.

Because a KeyMatch is specific to a front end, you can aim it at specific types of end users, as shown in the following example. Suppose you administer a search appliance for an e-commerce business, *mediacompany.com*. This business sells DVDs online in various countries. The business maintains an e-commerce website with a home page, *www.mediacompany.com*. This home page contains links to pages for end users in different regions, including North America, South America, Europe and the Middle East, Asia, Australia, and Africa.

You know that when end users search for “new movies,” they are most interested in navigating to *http://mediacompany.com/DVD/recentreleases*. You might provide this link on top of the search results as a KeyMatch, as shown in the following figure.



KeyMatches are not set up by default. You create a KeyMatch for a specific front end by associating a search term to a specific URL and specifying a title for the match. In the previous example:

- The search term is “new movies”
- The URL is *http://mediacompany.com/DVD/recentreleases*
- The title is “New DVD releases”

You can submit any number of URLs per *word, phrase, or exact match*. By default, however, only a maximum of three KeyMatches are returned for a search. You can increase the maximum number of KeyMatches returned up to 50 by using the `numgm` query parameter, which is described in the *Search Protocol Reference*.

The search appliance returns KeyMatch results based on the order in which the rules are created. For example, the following figure shows a front end with four KeyMatch rules defined:

Search Terms	Terms Occur As	URL for Match	Title for Match
stock	KeywordMatch ▼	http://stock.com	Stock
bond	KeywordMatch ▼	http://bond.com	Bond
stock bond	PhraseMatch ▼	http://stock-bond.com	Stock and Bond
bond stock	PhraseMatch ▼	http://bond-stock.com	Bond and Stock

If an end user searches for “stock bond,” the search appliance returns the following KeyMatch results, in the order shown:

1. <http://stock.com>
2. <http://bond.com>
3. <http://stock-bond.com>

If an end user searches for “bond stock,” the search appliance returns the following KeyMatch results, in the order shown:

1. http://stock.com
2. http://bond.com
3. http://bond-stock.com

Working with KeyMatches

To work with KeyMatches, use the **KeyMatch** tab on the **Search > Search Features > Front Ends > Output Format** page in the Admin Console. For any front end, you can:

- Select KeyMatch types
- Add KeyMatches individually
- View KeyMatches
- Edit KeyMatches
- Search for KeyMatches
- Delete KeyMatches
- Import KeyMatches from a URL or file
- Export KeyMatches

Any KeyMatch file that you import must be in UTF-8 encoding. For complete information about using the **KeyMatch** tab, click **Admin Console Help > Search > Search Features > Front Ends > KeyMatch** in the Admin Console.

Identifying KeyMatches

To identify potential KeyMatches:

- Check results for top queries by running a report on searches that returned results using the **Reports > Search Reports** page in the Admin Console. Examine each query. If the best result for a query is not the first result, you might create a KeyMatch.
- Check missed query terms by creating a report of unsuccessful search queries, as described in “Analyzing Searches that Do Not Return Results” on page 84. If several users are searching on missed query terms, you might create KeyMatches for the missed query terms.

You might also get input for KeyMatches from people in your own organization who have domain expertise. For example, suppose you want to improve searches that pertain to sales issues. You might ask people in your sales offices to identify keywords that need KeyMatches.

Changing the Appearance of KeyMatches

You can change the appearance of KeyMatches in the user interface for a front end. Components that you can change include:

- Label text (**KeyMatch**)

- Label color
- Background color

For example, you might make the following changes:

- Reword the label from **KeyMatch** to **Recommended Links**
- Change the KeyMatch text color from blue (#2255aa) to violet (#6600cc)
- Change the KeyMatch background color from pale violet (#e8e8ff) to aqua (#ccffff)

The following figure illustrates these changes.



To make changes to the appearance of KeyMatches, edit the Result Page Component section of the XSLT stylesheet. For information about editing the XSLT stylesheet, refer to “Customizing the User Interface in the XSLT Stylesheet” on page 105.

Using Related Queries to Suggest Alternative Searches

You can help users refine their searches by offering related queries. A related query is a suggestion for an alternative word or phrase for the user's original search term. The related query appears at the top of search results.

Because a related query is specific to a front end, it can be aimed at a specific type of end user, as shown in the following example. In an organization, projects and products are often developed under an internal name, but launched under a different name. For example, suppose *mediacompany.com* develops a new MP3 player under the code name “Malta,” but releases it as “Valise.”

Some internal users may not be familiar with the name “Malta.” When they search for “Valise,” they miss all the documents indexed under “Malta.” You can create a related query for “Malta” that is returned when users search for “Valise,” as shown in the following example:

You could also try: [Malta](#)

When the user clicks “Malta,” the search appliance runs the search again and returns additional results.

An alternative search term might be useful for users of several front ends. In this instance, you might consider using query expansion (see “Using Query Expansion to Widen Searches” on page 60) to add it as a local synonym.

Related queries are not set up by default. You can create a related query by associating a search term with a related query. In the previous example:

- The search term is “Valise”
- The related query is “Malta”

Search terms and related queries can be words or phrases.

Related queries are unidirectional only. When the end user's query matches the search term, the related query appears. However, when the end user's query matches the related query, the related query does not appear. Continuing the previous example, if a user searches for “Malta,” the search term “Valise” does not appear.

When you add related queries to a new front end, they can appear immediately. If you add them to an existing front end, it can take 30 minutes or more for them to appear.

Working with Related Queries

To work with related queries, use the **Related Queries** tab on the **Search > Search Features > Front Ends** page in the Admin Console. For any front end, you can:

Add related queries individually

- View related queries
- Edit related queries
- Search for related queries
- Delete related queries
- Import related queries from a URL or a file
- Export related queries

For complete information about using the **Related Queries** tab, click **Admin Console Help > Search > Search Features > Front Ends** in the Admin Console.

Identifying Related Queries

To identify potential related queries, get a list of the query terms that do not return results by creating a report of unsuccessful search queries. For information about this task, refer to “Analyzing Searches that Do Not Return Results” on page 84.

You can ask domain experts in your organization for input. For example, suppose you want to improve searches that pertain to personnel issues. You might ask people in your Human Resources office to identify keywords to use.

Changing the Appearance of Related Queries

You can change the appearance of related queries in the user interface for a front end. Components that you can change include:

- Label text (**You could also try**)
- Label color

For example, you might make the following changes:

- Reword the label from **You could also try** to **Search again using this term?**
- Change the label color from red (#CC0000) to orange (#FF6600)

The following figure illustrates these changes.

Search again using this term?: Malta

To make changes to the appearance of related queries, edit the Result Page Component section of the XSLT stylesheet. For information about editing the XSLT stylesheet, refer to “Customizing the User Interface in the XSLT Stylesheet” on page 105.

Using Dynamic Result Clusters to Narrow Searches



Sometimes, users search with terms that return an overly broad set of results. You can help end users narrow searches by enabling dynamic result clusters. Dynamic result clusters show different topics for a specific search term. These topics enable users to focus on areas of interest while ignoring irrelevant information.

For example, suppose your website offers memberships to users. When an end user searches for “membership,” a dynamic result cluster appears with the a list of topics found in the results. The following example illustrates this dynamic results cluster.

narrow your search

[membership options](#)

[membership help](#)

[membership pricing](#)

[membership privileges](#)

[membership specials](#)

When a user clicks on any of the topics, the search appliance returns a new, narrower set of results.

Dynamic result clusters are formed from each set of results. Dynamic result clusters work best when the results contain sufficient documents to allow the search appliance to categorize them into topics.

By default, dynamic result clusters are disabled. For any front end, you can:

- Enable or disable dynamic result clusters.
- Specify the placement of dynamic result clusters at the top or to the right of search results. If you have implemented KeyMatches (see “Using KeyMatches to Guide Users to URLs” on page 43) or OneBox modules (see “Using OneBox Modules to Integrate Structured Results” on page 70), you might want to place the dynamic result clusters to the right of search results. This minimizes user scrolling down the page for natural search results.

Dynamic result clusters work with secure queries. Before displaying dynamic result clusters for secure queries, the search appliance ensures that the user has permission to view secure results by triggering authorization checks. These authorization checks might have a negative impact on system performance.

Enabling or Disabling Dynamic Result Clusters

The Google Search Appliance supports two methods of enabling and disabling dynamic result clusters:

- “Using the Page Layout Helper” on page 48
- “Using the XSLT Stylesheet Editor” on page 48

You cannot use the Page Layout Helper to enable or disable dynamic result clusters if you have customized the XSLT stylesheet. If you have done this, you must enable or disable dynamic result clusters directly in the XSLT stylesheet.

If you want to use a customized XSLT stylesheet from an earlier release, refer to the update instructions for the current search appliance software release.

Using the Page Layout Helper

To enable or disable dynamic result clusters using the Page Layout Helper:

1. Choose **Search > Search Features > Front Ends**.
The **Front Ends** page appears.
2. Select a front end from the **Current Front Ends** list and click **Edit**.
The **Output Format** page appears.
3. In the **Page Layout Helper** box, select the **Search Results** section.
The section expands.
4. To enable dynamic result clusters, click **Dynamic result clusters**. A checkmark appears.
Notice that dynamic result clusters appear in the preview.
To disable dynamic result clusters, click **Dynamic result clusters**. The checkmark disappears.
5. To change where cluster topics appear on the page, click **Top** or **Side**.
6. Click **Save**.

Using the XSLT Stylesheet Editor


This section describes enabling dynamic result clusters using the XSLT Stylesheet Editor. If you prefer, you can edit the XSLT stylesheet in another editor.

To enable dynamic result clusters in an XSLT stylesheet:

1. Choose **Search > Search Features > Front Ends**.
The **Front Ends** page appears.
2. Select a front end from the **Current Front Ends** list and click **Edit**.
The **Output Format** page appears.
3. In the XSLT Stylesheet Editor, scroll to the dynamic result cluster options in the Other Variables section. The following code shows the dynamic result cluster options.

```
<!-- *** dynamic result cluster options *** -->
<xsl:variable name="show_res_clusters">0</xsl:variable>
<xsl:variable name="res_cluster_position">right</xsl:variable>
```
4. Enable or disable dynamic result clusters:
 - To enable dynamic result clusters, change the `show_res_clusters` value from 0 (zero) to 1 (one).
 - To disable dynamic result clusters, change the `show_res_clusters` value from 1 (one) to 0 (zero).
5. To change where the cluster topics appear on the page, from the right side (default) to the top of the page, change the `res_cluster_position` value from `right` to `top`.
6. Click **Save**.

Using Dynamic Navigation to Help Users Explore Results

 In many cases, content already has considerable metadata associated with it. As a search appliance administrator, you can use metadata to help users refine their search results by using Dynamic Navigation. When a user clicks on an attribute value, the search results are filtered to contain results from the original search query that also have that specific attribute value. The options are refreshed with the attribute values that are applicable to the new result set. Users can select multiple attributes and can easily back out of their selections to navigate the result set and quickly locate the results they are looking for.

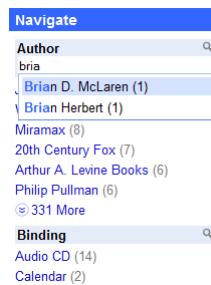
With dynamic navigation, you can use the following types of attributes:

- Attributes based on `META` tags found in the HTML of the documents in the search index
- Attributes based on metadata from databases, connectors, or feeds
- Attributes based on entities discovered by entity recognition

For example, suppose multiple HTML documents in *mediacompany.com*'s corpus include the following metadata value pairs:

```
<META NAME="author" CONTENT="Nikhil Jain">
<META NAME="author" CONTENT="Radha Chandika">
<META NAME="author" CONTENT="Rajat Mukherjee">
```

By using the **Search > Search Features > Dynamic Navigation** page, you can add the `author` attribute so that it appears as a “Author” option on the search results page. All the values associated with the `author` attribute appear under the option, as shown in the following figure.



Dynamic navigation displays the counts for all matching results, as well as the number of values not displayed for an attribute, with a **More** link. Dynamic Navigation also provides search with auto-completion for attributes that have more values to display. This is indicated by a search icon (small magnifying glass) in the attribute name bar. To activate search with auto-completion, the user clicks the attribute name bar. As the user types in the search box, she can select any value in the auto-completion drop-down menu to filter the results.

You can use dynamic navigation by performing the following tasks:

1. “Enabling Dynamic Navigation” on page 51.
2. “Creating a Configuration and Adding Attributes” on page 51.
3. “Showing Dynamic Navigation in a Front End” on page 52.

A configuration defines the metadata attributes that are used to generate dynamic navigation options. By using the **Search > Search Features > Dynamic Navigation** page, you can create different configurations and apply the configurations to different front ends.

There is no limit to the number of attributes that you can configure in dynamic navigation. However, adding more attributes might affect search latency. The maximum number of results returned per attribute is 5000.

Also, for any configuration, you can enable dynamic navigation for secure search. With secure search, dynamic navigation only includes documents that the user is authorized to see. For more information, see “Dynamic Navigation for Secure Search” on page 50. You can also configure the search appliance to automatically expand configured dynamic navigation attributes to include other terms. For more information, see “Configuring Query Expansion for Dynamic Navigation” on page 52.

Dynamic Navigation for Secure Search

With dynamic navigation for secure search, the search appliance only uses documents that the user is authorized to see to generate navigation options. Only attributes and values that are in the accessible documents appear in the options and the counts include only the accessible documents.

To use dynamic navigation for secure search, enable it for a configuration and select one of the following authorization modes:

- “Fast Authorization Mode” on page 50
- “All Authorization Mode” on page 50

For information about how to enable dynamic navigation for secure search, see “Creating a Configuration and Adding Attributes” on page 51.

Fast Authorization Mode

In fast authorization mode, the search appliance only performs authorization based on Access Control Lists (ACLs) and checks results that are

already available in cache to generate dynamic navigation options. It ignores documents that require real-time authorization. If the search appliance ignores some documents, it indicates this with a greater than sign (>) in the count that appears with the dynamic navigation options.

Google recommends using fast authorization mode, which is the default mode.

All Authorization Mode

In all authorization mode, the search appliance uses all types of authorization methods to generate dynamic navigation options. If ACLs are not the primary method of authorization, users might receive query timeout errors (500) when performing searches. If this happens, you might consider using fast authorization mode instead.

Google recommends using all authorization mode only under the following conditions:

- Real-time authorization on the content server is fast.
- Only 3-5% of the documents in the corpus need real-time checks.
- The rest of the documents in the corpus use ACLs for authorization.

Enabling Dynamic Navigation

Dynamic navigation is disabled by default. To enable dynamic navigation:

1. Choose **Search > Search Features > Dynamic Navigation**.
2. Click **Enable**.

To disable dynamic navigation, click **Disable**.

Creating a Configuration and Adding Attributes

By using the **Search > Search Features > Dynamic Navigation** page, you can configure different attributes in different configurations and apply the configurations to different front ends. Because an attribute that you add by using this page must exactly match the metadata NAME, ensure that you have the correct name from the metadata attribute before adding the dynamic navigation attribute.

To add a new configuration and attribute:

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.
4. Under **Attributes**, click **Add**.
5. In the **Display Label** box, type the label for the attribute.
6. In the **Attribute Name** box, type the name of the attribute.
7. Select the attribute type from the **Type** pull-down menu.
8. If the type is INTEGER, FLOAT, CURRENCY, or DATE, click **Range Attribute**, if appropriate.
9. If the type is STRING, click **Case-Insensitive Attribute**, if appropriate.
If the attribute is a **Range Attribute**, then provide the range values. No sorting options are available for range attributes. The values appear in the order entered.

If the attribute is not a **Range Attribute**, then select the sorting options for the attribute. You can select to sort the attribute values by either counts or by values themselves and in ascending or descending order.
10. Click **OK**.
11. To add more attributes in this configuration, repeat steps 4 to 10.
12. Under **Front Ends**, apply the configuration to one or more front ends by selecting a front end names from the **Available Front Ends** box. Selecting them moves them to the **Added Front Ends** box. To remove the added front ends, select them in the **Added Front Ends** box and they will move back to the **Available Front Ends** box. The front ends added to one configuration are not available in other configurations.
13. Optionally, to use secure dynamic navigation, click **Enable dynamic navigation for secure search** under **More Options** and select one of the following authorization modes:
 - **Use only fast authorization** (see “Fast Authorization Mode” on page 50)
 - **Use all types of authorization** (see “All Authorization Mode” on page 50)
14. To create the configuration, click **Create**.

After this step, the search appliance starts computing the dynamic navigation results for the selected front ends but does not yet show the results on the search pages. However, you can see the dynamic navigation results in the search XML output. To display the dynamic navigation results on the search page, follow the procedure in “Showing Dynamic Navigation in a Front End” on page 52.

For complete information about dynamic navigation configurations, click **Admin Console Help > Search > Search Features > Dynamic Navigation** in the Admin Console.

Showing Dynamic Navigation in a Front End

After enabling dynamic navigation, creating a configuration with some attributes, and applying the configuration to a front end, you can show dynamic navigation with search results in a specific front end.

To show dynamic navigation in a front end:

1. Click **Search > Search Features > Front Ends** and click **Edit** for a particular front end.
2. In the **Page Layout Helper** box on the **Output Format** tab, select the **Search Results** section.
3. Click **Show Dynamic Navigation**.
4. Click **Save**.

Configuring Query Expansion for Dynamic Navigation

The search appliance enables you to configure query expansion (see “Using Query Expansion to Widen Searches” on page 60) for configured dynamic navigation attributes.

To configure query expansion for dynamic navigation:

1. Click **Search > Search Features > Query Settings**.
2. Under **Add Query Expansion File**, click **Synonyms**.
3. For **Language**, select either **all** or the particular desired language.
4. Browse to a synonyms file containing synonyms for dynamic navigation attributes
5. Click **Apply Settings**.
6. Click **Search > Search Features > Front Ends**.
7. Click the **Edit** link next to the front end for which dynamic navigation has been applied and which needs attributes names synonyms.
8. Select the **Filters** tab.
9. For **Query Expansion Policy for Meta Tags**, select **Names only** from the pull-down menu.
10. Click **Save**.

Dynamic Navigation starts using these synonyms in its calculations for a query against that front end.

Providing Expert Search for Users

You can enhance the search experience for your users with expert search, which enables them to find experts in your organization. When a user searches on a keyword, such as “security,” a list of security experts appears in a sidebar next to search results.

This list includes photos, names, job titles, email addresses, locations, and phone numbers. There might also be a more detailed list of experts on a separate page that is linked to the search results page.

To use expert search, perform the following tasks:

1. Configuring data sources for expert search
2. Enabling and configuring expert search

You can determine the profile elements and their layout in both the sidebar and on a separate page when you configure expert search. Because expert search is configured by front end, you can create different configurations for that address the needs and levels of different end users.

Configuring Data Sources for Expert Search

The search appliance gets profile properties for expert search from collections that only contain profile metadata. Data sources for this collection can include Microsoft SharePoint, LDAP, or any other profile content with metadata that can be crawled or fed into the search index.

Before configuring expert search for a front end, add profile metadata to the index. You can add it through crawling documents or databases with metadata, using feeds, or by using connectors, such as the SharePoint or LDAP connectors. After populating the index with metadata, you can add a collection that only contains the profile metadata.

For complete information about configuring data sources for expert search, click **Admin Console Help > Search > Search Features > Expert Search** in the Admin Console.

Enabling and Configuring Expert Search

To enable and configure expert search for a front end, use the **Configure User Interface** tab on the **Search > Search Features > Expert Search** page. This tab lists all the front ends that have been created for the search appliance.

To enable and configure expert search:

1. Click **Search > Search Features > Expert Search**.
2. On the **Configure User Interface** tab, click **Configure** on the line corresponding to the front end where you want to set up expert search.
3. Under **Selected Front End**, click **Save**.
4. Perform the following tasks:
 - a. Configuring a collection containing expert data
 - b. Selecting meta tags for the configuration
 - c. Configuring expert layout

For complete information about enabling and configuring expert search, click **Admin Console Help > Search > Search Features > Expert Search** in the Admin Console.

Using People Search Deprecated

In the current search appliance release, people search is deprecated.

With people search, when a user searches on a keyword, the search appliance searches any source collection that you specify and displays people search profiles that match the search term in a sidebar next to ranked search results.

For example, suppose a user searches for “Lee.” People search results appear in the sidebar on the search results page, as shown in the following figure.



People search profiles can include:

- Images—Such as photos of members of your organization
- Text—Such as a member's name, job title, manager, office location, and phone number
- Links—Both images and text can be linked to web pages

The profile information that appears in people search results is derived from metadata attributes found in the HTML of the documents in the search index, or metadata from databases, feeds, or connectors, including the LDAP connector. If the profile information for a single person is drawn from multiple sources, then this information must be merged into a single document before it is sent to the search appliance.

Before configuring people search, populate the search index with content and metadata for people search. By using metadata attributes and the **Search > Search Features > Expert Search** page, you can configure the attribute profile information that you want to appear in search results.

For example, suppose your people database includes the following metadata attribute value pairs:

```
<meta name="cn" content="Benjamin Grimm"/>
<meta name="uid" content="bgrimm"/>
<meta name="title" content="Test Pilot"/>
<meta name="location" content="401D"/>
<meta name="telephoneNumber" content="+1 (555) 465-2782"/>
<meta name="manager" content="Lee"/>
```

By configuring people search using this metadata, you can enable the search appliance to return people search profiles, such as the one for Benjamin Grimm, shown in the preceding figure. For information about how to specify attribute names on the **Search > Search Features > Expert Search** page, click **Admin Console Help > Search > Search Features > Expert Search** in the Admin Console.

You can enable the search appliance to return people search results by performing the following tasks:

- “Configuring People Search” on page 55
- “Showing People Search Results in a Front End” on page 56

Before configuring people search, create a people search source collection. A people search source collection is a segment of the search index that contains profile information. The source collection must contain one document for each person. You can create a people search source collection by using the **Index > Collections** page in the Admin Console. For information about creating a collection, click **Admin Console Help > Index > Collections**.

Configuring People Search

To configure people search:

1. Click **Search > Search Features > Expert Search** and select the **People Search** tab.
2. Select a collection for people search from the **Source Collection** pull-down menu.
3. To change the number of profiles that appear in the people search sidebar, enter a number in the **Maximum Results** box.
4. To change the title of the people search results sidebar element, enter text in the Title box.
5. To change the size of the people search sidebar element, enter the number of pixels in the **Sidebar Height** box.
6. To specify a URL for profile images, enter it in the **Image Source URL** box.
7. To make the profile image a link, enter a URL in the **Image Link URL** box.
8. To change the scale of the profile image, enter the number of pixels in the **Image Height** and **Image Width** boxes.
9. To change the sort order of the people search results, enter one or more attribute substitution patterns in the **Sort Fields** box.
10. Specify a row of information in each people search profile:
 - a. To specify a label for the display value, enter a text string in the **Label** box. To specify formatting of the label, enter HTML and/or XSL tags.
 - b. To specify the content of a people search results row, enter one or more attribute substitution patterns in the **Display Value** box. To specify formatting of the attribute value, enter HTML and/or XSL tags.
 - c. To make the display value a link, enter a URL in the **Link URL** box.
11. To specify another row, go to step 7.

You can specify a maximum of 16 rows.
12. Click **Save**.

For complete information about configuring people search, click **Admin Console Help > Social Connect > People Search** in the Admin Console.

Showing People Search Results in a Front End

After you configure people search, show the people search results sidebar element in a front end.

To show the people search sidebar element in a front end:

1. Click **Search > Search Features > Front Ends** and click **Edit** for a particular front end.
2. In the **Page Layout Helper** box on the **Output Format** tab, select the **Search Results** section.
3. Under **Sidebar Elements**, click **People Search results**.
4. Click **Save**.

Experimenting with Host Crowding Options

You might notice that search results are sometimes indented, as shown in the following example.

[Google Enterprise](#)

Google enterprise products make your users more productive by combining the innovation and ease of use of Google's consumer products with the features, ...

[www.google.com/enterprise/](#) - 10k - [Cached](#) - [Similar pages](#) - [Note this](#)

[Google Enterprise: Google Mini Search Appliance](#)

Employees' familiarity with Google's user interface from searching on Google.com will facilitate adoption in your enterprise, and instant access to ...

[www.google.com/enterprise/mini/](#) - 13k - [Cached](#) - [Similar pages](#) - [Note this](#)

[[More results from www.google.com](#)]

The indenting indicates “host crowding,” or presenting only the most relevant search results by eliminating duplicates. To eliminate duplicate search results, the search appliance uses the following automatic filters:

- **Duplicate Snippet Filter**—If multiple documents contain identical titles as well as the same information in their snippets in response to a query, only the most relevant document of that set is displayed in the results.
- **Duplicate Directory Filter**—If there are many results in a single web directory, then only the two most relevant results for that directory are displayed. The previous example of host crowding illustrates this type of filtering.

By default, automatic filters are enabled and host crowding significantly reduces the number of results returned.

In some situations, you can turn off host crowding by disabling one or more automatic filters to produce better relevancy. You might want to experiment with the settings for the Duplicate Directory filter and/or the Duplicate Snippet filters.

To disable the Duplicate Directory filter and/or the Duplicate Snippet filter, use the filter query parameter. For information about this topic, refer to *Filtering* in the *Search Protocol Reference*.

Using Filters to Restrict Search Results



As an administrator, you can create custom filters for a front end to ensure that the search appliance serves appropriate results to end users. Filters are especially useful when a search appliance has multiple front ends for multiple types of end users.

The following table lists the types of filters you can create, with references to sections that provide more details on each type of filter.

Filter	Description	Refer to
Domain	This type of filter restricts searches to one or more domain names (not IP addresses).	"Restricting Search Results by Domain Name" on page 57
Language	This type of filter restricts searches to either all languages or a selected set of languages.	"Restricting Search Results by Language" on page 58
File type	This type of filter restrict searches to one or more content types or file types, such as HTML, PDF, and so on.	"Restricting Search Results by File Type" on page 58
Meta tag	This type of filter restricts searches by values and value types in meta tags.	"Restricting Search Results by Meta Tag" on page 58

You can use multiple filters in a front end. For example, a front end that is used in a specific region could filter results by both domain and language.

Working with Filters

To work with filters, use the **Filters** tab on the **Search > Search Features > Front Ends > Output Format** page in the Admin Console. For complete information about using the **Filters** tab, click **Admin Console Help > Search Features > Front Ends > Filters** in the Admin Console.

Restricting Search Results by Domain Name

You can use a domain filter to restrict results by domain name.

Suppose you want to ensure that searches in various regions return only results with local information. For example, you want to restrict results on the Australian pages to show only products and special offers available there, so you create a front end for users in Australia.

The domain name for Australia is *www.mediacompany.com.au*. You might use this domain name to create a domain filter so that when end users in Australia perform a search, only results that match the domain name appear. All domains ending with the name are filtered. For example, the domain *mediacompany.com.au* returns search results in all of the following domains:

- *www.mediacompany.com.au*
- *news.mediacompany.com.au*
- *ops.mediacompany.com.au*

However, if the domain name is followed by a directory name, the domain name must be fully qualified when you enter it on the **Filters** tab. For example, *mediacompany.com.au/marketing* does not return any results under the domain name filter alone. The correct filter is *www.mediacompany.com.au/marketing*. If the directory *marketing* is in multiple domains, each filter must be specified individually on the **Filters** tab—one per line, one for each domain, as shown in the following example:

```
www.mediacompany.com.au/marketing
news.mediacompany.com.au/marketing
```

Restricting Search Results by Language

You can use a language filter to restrict results by language.

Suppose you have created a front end for users in Switzerland. You want to restrict search results for this front end to the country's three official languages, German, French, and Italian. You might create a language filter for the Switzerland front end with these languages selected. When end users in Switzerland perform a search, only results in the selected languages appear.

When the search appliance receives a language-restricted search request for which there are no results in the languages specified by a filter, it displays search results in *all* languages. If search users in Switzerland perform searches for which there are no results in German, French, or Italian, the users see results in other languages that are represented in the indexed content.

You can also create a filter with the **Filters** tab to display results for pages in any language.

Restricting Search Results by File Type

The search appliance can crawl many file types. You can use a file type filter to restrict results by file type. For a list of file types that the search appliance can crawl, click **Admin Console Help > More Information > Crawling and Indexing** in the Admin Console.

Suppose you have created a front end for *mediacompany.com*'s sales staff. The only documents that interest the sales staff are in Portable Document Format (.pdf) or Microsoft PowerPoint (.ppt) format. You can create a file type filter that serves only these file types in the sales staff front end. When members of the sales staff perform a search, only results of the selected file types appear.

Restricting Search Results by Meta Tag

You can use meta tag filters to restrict results by meta tags and meta tag values.

The site *www.mediacompany.com* sells DVDs worldwide. Most DVDs are encoded by region. For example, DVDs encoded for region 1 (Canada, the United States, Bermuda, the Cayman Islands, and U.S. territories) play only on region 1 players.

On the website, each DVD has a product page that includes a meta tag indicating its region, as shown in the following example.

```
<META NAME="region" CONTENT="region_1">
```

Suppose you have created a front end for users in region 1. You can restrict search results to products for region 1 only. You might create a meta tag filter for the region 1 front end so that only documents with CONTENT=region_1 appear. When end users in region 1 perform a search, only results for DVDs that are playable in region 1 appear.

Value Type Input Parameters

You can also filter results by the values of the results' meta tags using the **requiredfields** and **partialfields** input parameters in a search query. For information about these input parameters, refer to the *Search Protocol Reference*.

If you set meta tag filters in the front end (using the **Meta Tag Filter** area), the search appliance appends these settings to the search query as **requiredfields** and **partialfields** input parameters. If you set meta tag filters in both the front end and in **requiredfields** and **partialfields** input parameters, the front end settings overwrite any input parameters that you have added to the search query.

The following table lists front end meta tag value types and the input parameters that the search appliance appends to the search request for each value type.

Front End Value Type	Input Parameter
Exact	requiredfields=name:value
Partial	partialfields=name:value
Existence	requiredfields=name

Selecting a Query Expansion Policy for Meta Tags

Use a query expansion policy for meta tags to select the parts of the name/value pair in a meta tag that the search appliance expands with synonyms. The data used in this expansion comes from the query expansion files listed under **Synonyms Data** on the **Search > Search Features > Query Settings** page. Each file can be enabled and disabled independently for normal query expansion, meta name expansion and meta value expansion. Google recommends using a separate file for meta tag expansion. Using the normal language file may result in unintended expansions, such as stemming words.

If you have dynamic navigation enabled on this front end, enabling query expansion for meta tag also affects dynamic navigation. For more details, click **Admin Console Help > Search > Search Features > Dynamic Navigation**.

To set the query expansion policy for meta tags, use the **Filters** tab of the **Search > Search Features > Front Ends** page in the Admin console. For complete information about using the **Filters tab**, click **Admin Console Help > Search > Search Features > Front Ends > Filters** in the Admin Console.


To set the query expansion policy for meta tags for a front end:

1. Click **Search > Search Features > Front Ends > Filters**.
2. In the **Query Expansion Policy for Meta Tags** text box, select one of the following values:
 - None: Disables query expansion for meta tags in the front end.
 - Names only: Enables query expansion for names only.
 - Values only: Enables query expansion for values only.
 - Names and values: Enables query expansion for names and values.
3. Click **Save**.

To disable the query expansion policy for meta tags for a front end:

1. In the **Query Expansion Policy for Meta Tags** text box, select **None**.
2. Click **Save**.

Removing URLs from Search Results

 Occasionally, a search index contains URLs that the search appliance should not serve. For any front end, you can prevent the search appliance from serving URLs that match specific patterns.


For example, suppose you do not want the search appliance to serve results about out-of-print DVDs in the customer front end. All documents that pertain to out-of-print DVDs are located in *www.mediacompany.com/obsolete/*. To prevent results from this URL being served, enter the URL pattern in the box on the **Remove URLs** tab. The URL patterns you provide must conform to the rules for valid URL patterns (see “Constructing URL Patterns” in *Administering Crawl*).

You can add URL patterns at any time, and they will be immediately removed from the served results. You can also delete the URL patterns at any time to return those patterns to the served results immediately.

For complete information about using the **Remove URLs** tab, click **Admin Console Help > Search > Search Features > Front Ends > Remove URLs** in the Admin Console.

The remove URLs feature affects results only. It does not remove URLs from the index. To remove URLs from the index, enter them in the **Do Not Follow Patterns** section on the **Content Sources > Web Crawl > Start and Block URLs** page in the Admin Console. For more information about removing URLs from the index, refer to *Administering Crawl*.

Using Query Expansion to Widen Searches

 The Google Search Appliance can automatically expand search terms to include other terms. This expansion is based on synonyms, or alternative terms for the original search term, as well as inflection variants, and related terms. The search appliance includes the following types of query expansion files:

- Built-in query expansion files, which you cannot download or edit
- Preconfigured local query expansion files, which you can download and edit

Additionally, you can add your own customized local query expansion files to a search appliance.

Whenever a term matches a built-in synonym, the query is expanded to include the synonym. For example, suppose a user searches with the term “video.” Because this term matches a built-in synonym (“DVD”), the search is expanded to include the term DVD.

You can add your own set of meaningful terms to be added as synonyms. When query expansion is enabled, all synonyms on a search appliance apply to all front ends. To add terms for a specific front end, consider adding related queries instead (see “Using Related Queries to Suggest Alternative Searches” on page 45).

There are two steps to adding synonyms to query expansion:

1. “Working with Query Expansion Files” on page 60
2. “Setting the Query Expansion Policy for a Front End” on page 65

Working with Query Expansion Files

This section describes tasks for managing query expansion files for a Google Search Appliance, including:

- “Using Preconfigured Local Query Expansion Files” on page 61
- “Using Customized Local Query Expansion Files” on page 62
- “Using Blacklists” on page 63
- “Using Stopwords” on page 64

To manage query expansion files, use the **Search > Search Features > Query Settings** page in the Admin Console. For complete information about using the **Query Settings** page, click **Admin Console Help > Search > Search Features > Query Settings** in the Admin Console.

Using Preconfigured Local Query Expansion Files

By default, query expansion terms are available in several languages. The Google Search Appliance's default language bundle contains these languages. You can change the supported languages by installing and activating a different language bundle (see “Changing Languages for Query Expansion and Spelling Suggestions” on page 66).

For customers who are using the supported languages, the following preconfigured synonyms files are provided with the search appliance:

- Google_Arabic_stems
- Google_Czech_stems
- Google_Dutch_stems
- Google_English_stems
- Google_French_stems
- Google_German_stems
- Google_Italian_stems
- Google_Polish_stems
- Google_Portuguese_stems
- Google_Russian_stems
- Google_Slovak_stems
- Google_Spanish_stems
- Google_Swedish_stems

These files appear by default in the list of query expansion files on the **Search > Search Features > Query Settings** page in the Admin Console. Each file contains a set of common words that can supplement standard terms. By default, Google_English_stems is enabled.

You can use a preconfigured local file as it is. Alternatively, you can:

- Download a preconfigured local file to edit it
- Upload a modified file
- Enable or disable a preconfigured local file

To perform any of these tasks, go to the **Search > Search Features > Query Settings** page.

You cannot delete preconfigured local files.

Using Customized Local Query Expansion Files

You can use customized local query expansion files to configure site-specific terminology. For example, you might:

- Configure synonyms that match obsolete catalog numbers with their replacement catalog numbers. A user who searches for an obsolete catalog number would get results for both the obsolete catalog number and the new catalog number.
- Configure synonyms that expand catalog abbreviations to full names. For example, a user who searches for “L. Bunuel” would get results for both the catalog abbreviation and Luis Bunuel.
- Configure synonyms that expand generic categories to product names. For example, a user who searches for “documentary” would get results for both documentary and Best Documentary Series.

A local synonyms file is a text file of three megabytes or less, containing case-insensitive entries.

Follow these steps to create a local synonyms file:

1. Create a text (.txt) file.
2. If the file will contain accented characters and you have not already checked your editor's ability to save a file with UTF-8 encoding, do so now. As an example, if you are using Notepad, do this:
 - From the File menu, choose **Save As**.
 - Check that the Save options include Encoding, as well as Name and File Type.
 - Pull down the Encoding menu and choose UTF-8.
3. Edit the file as follows:
 - Put one directive on each line, using the formats described under Entry format 1 and Entry format 2, below.
 - Use only alphanumeric characters and spaces, substituting spaces for hyphens. For example, instead of entering “pro- democracy,” enter “pro democracy.” The results will include the hyphenated version of the term. Each term can consist of multiple words separated by spaces; the maximum number of words for each term is 4.
 - Specify any number of synonyms for a particular term. There is no limit to the number of expansions in a query.
 - Use the pound sign (#) to start a comment line.
4. Save the file. If the file has accented characters, save it with UTF-8 encoding.

Entry format 1: *term1 operator term2*

In this format:

- *term1* consists of one word or multiple words that are separated by single spaces.
- *term2* consists of one word or multiple words that are separated by single spaces. *operator* is one of the following:
 - = Specifies that the expansion is bi-directional. The appliance expands a search query for *term1* or *term2* by adding the other term.
 - > Causes the appliance to add *term2* when a search query contains *term1*. To expand *term1* into multiple terms, you specify multiple entries separated by commas: *term1 > term2, term1 > term3*.

Examples:

```
ebu = education business unit
tbu = telecomm business unit
telecom business unit = telecomm business unit
partner > indirect sales
widgets > parts, widgets > items
```

A query where the search term is `ebu` is expanded to include additional documents that contain the phrase `educational business unit`. A query where the search term is `educational business unit` is expanded to include additional documents that contain the term `ebu`. Notice that the two queries, `ebu` and `educational business unit`, are not equivalent. Also, a query where the search term is a phrase (enclosed by quotation marks), such as `"educational business unit"`, is not expanded.

Entry format 2: `{term, term, ...}`

In this format:

- The curly brackets are required.
- Each term in the list will be used to expand queries for each other term.
- Up to 32 terms are permitted.
- Terms can contain space characters but they cannot contain commas.
- A minimum of two terms is required.

Examples of valid entries:

```
{run, runs, running, ran}
{widgets, parts, items}
```

Example of an invalid entry:

```
{safe}
```

For information about uploading a synonyms file, click **Admin Console Help > Search > Search Features > Query Settings** in the Admin Console.

Using Blacklists

You can control query expansion by creating a blacklist. A blacklist is a set of words that are excluded from query expansion. A blacklist can be useful for eliminating unwanted search results that result from synonym matching and clarifying special words used in your environment.

For example, suppose *mediacompany.com* starts a new project with the code name "glue." You could add "glue" to the blacklist to ensure that user queries do not expand to include "adhesive."

A blacklist file is a text file of three megabytes or less, containing a simple case-insensitive list of single words, with one word on each line.

Follow these steps to create a blacklist file:

1. Create a text (.txt) file.
2. If the file will contain accented characters and you have not already checked your editor's ability to save a file with UTF-8 encoding, do so now. As an example, if you are using Notepad, do this:
 - From the File menu, choose Save As.
 - Check that the Save options include Encoding, as well as Name and File Type.
 - Pull down the Encoding menu and choose UTF-8.

3. Edit the file as follows:
 - Put one word on each line.
 - Blacklists should use only alphanumeric characters. Spaces are not allowed.
 - Use the pound sign (#) to start a comment line.
4. Save the file. If the file has accented characters, save it with UTF-8 encoding.

This is an example of a blacklist file:

```
#Blacklist file created July 2006
#Author: lana
glue
component
```

This file prevents queries for “glue” and “component” from being enhanced with synonyms.

For information about uploading a blacklist file, click **Admin Console Help > Search > Search Features > Query Settings** in the Admin Console.

Using Stopwords

A stopwords is a search term that is ignored by the search appliance. Examples of stopwords include “to,” “a,” and “the.” However, if a stopword is the only keyword in a query, it is not ignored. For example, if “salary” is a stopword and a user submits a query where “salary” is the only keyword, the query will execute and display results. But if a user searches for “my salary,” “salary” is ignored.

A stopwords file is a text file of three megabytes or less, containing a case-insensitive list of one or more alphanumeric stopwords. By default, the search appliance has 26 files of stopwords for supported languages. For a list of the default stopwords files, see the files listed under **Stopwords Data** on the **Search > Search Features > Query Settings** page. Of these files, two are enabled by default: `Google_Default_Stopwords` and `Google_English_Stopwords`. Additionally, you can create and upload your own stopwords files.

To create a Stopwords file:

1. Create a text (.txt) file.
2. If the file will contain accented characters and you have not already checked your editor's ability to save a file with UTF-8 encoding, do so now. As an example, if you are using Notepad, do this:
 - From the File menu, choose Save As.
 - Check that the Save options include Encoding, as well as Name and File Type.
 - Pull down the Encoding menu and choose UTF-8.
3. Edit the file as follows:
 - Put one word on each line. Blank lines are ignored.
 - Use only alphanumeric characters. Spaces are not allowed.
 - Use the pound sign (#) to start a comment line.
4. Save the file. If the file has accented characters, save it with UTF-8 encoding.

The following example shows an excerpt from the contents of the `Google_Default_stopwords.txt` file. Each of the following words is ignored in a search query.

```
the
of
to
in
for
is
on
that
by
with
this
be
it
www
are
...
```

For information about uploading a stopwords file, click **Admin Console Help > Search > Search Features > Query Settings** in the Admin Console.

For a stopwords file for a particular language to take effect, searches must be restricted to the same language. For example, if you upload and enable a stopwords file for French, searches must be restricted to French for those stopwords to take effect.

Searches can be restricted to a particular language in either of the following ways:

- Creating a language filter in a front end by using the **Filters** tab of the **Search > Search Features > Front Ends** page
- Adding the `lr=lang_<xyz>` (language restrict) query parameter to the search request

For information about creating a language filter in a front end, see “Restricting Search Results by Language” on page 58. For information about language restrict query parameter, see the *Search Protocol Reference*.

Files with Macintosh style end-of-line characters

Some text files created in Macintosh OS use Mac-style end-of-line characters (`/r`). Before uploading a synonyms or blacklist file that contains this style of end-of-line characters, ensure that the first line in the file is not a comment (that is, starts with `#`).

Setting the Query Expansion Policy for a Front End

After you have configured and enabled the appropriate query expansion files, you can set a query expansion policy for a front end. A query expansion policy determines the synonym or blacklist files that are used with a front end. You can use just one type of file or use both types of files together. You can create a combined total of up to 300 files with a front end.

The following table describes the query expansion policies that you can set for a front end.

Query Expansion Policy Setting	Description
None	Disables query expansion for the front end
Standard	Enables query expansion for the front end, using only the search appliance's internal contextual files
Local	Enables query expansion for the front end, using all displayed and activated synonym files, including any uploaded files.
Full	Enables query expansion for the front end, using both Google's built-in synonyms and the files that you upload to the appliance

Query expansion files are used only if the query expansion policy for a front end is set to **Local** or **Full** on the **Search > Search Features > Front Ends > Filters** tab.

To set the query expansion policy for a front end, use the **Filters** tab of the **Search > Search Features > Front Ends** page in the Admin console. For complete information about using the **Filters** tab, click **Admin Console Help > Search > Search Features > Front Ends > Filters** in the Admin Console.


Enabling Translation of Search Results

The Google Search Appliance can translate titles and snippets in search results, as well as cached documents into the user's language in real time. The user's language is determined by the default language set in the user's browser. When translation is enabled, translation links appear in search results. The user can translate everything on the page or just individual titles, snippets, or cached documents. Take note that translate does not work for the Document Preview feature.

When the user clicks a translation link, the search appliance sends the content, client-id, and user language to Google Translate using a secure connection. For more information about the Website Translator plugin, visit the [Google Translate Help Center](#). Google recommends that Internet Explorer customers use translate from the HTTPS search results page.

You can enable or disable translation for any front end. To enable translation, use the **Search > Search Features > Front Ends > Output Format** page in the Admin Console. For complete information about using this page, click **Admin Console Help > Search > Search Features > Front Ends > Output Format - Page Layout Helper** in the Admin Console.

Changing Languages for Query Expansion and Spelling Suggestions

 A language bundle is a collection of resource files that the Google Search Appliance uses for query expansion and spelling in several languages. The following table lists the language bundles that Google provides.

Language Bundle	Languages	Download Link
Built-in	<ul style="list-style-type: none"> • Dutch • English • French • German • Italian • Portuguese • Spanish 	Installed on the Google Search Appliance by default
All languages	<ul style="list-style-type: none"> • Arabic • Danish • German • Greek • English • Spanish • Finnish • French • Hungarian • Italian • Hebrew • Japanese • Korean • Dutch • Norwegian (includes both Bokmål and Nynorsk) • Polish • Portuguese • Romanian • Russian • Swedish • Thai • Turkish 	http://dl.google.com/dl/enterprise/all_langs-lang-pack-2.2-1.bin

Language Bundle	Languages	Download Link
Scandinavia	<ul style="list-style-type: none"> English German Dutch Swedish Norwegian (includes both Bokmål and Nynorsk) Danish Finnish 	http://dl.google.com/enterprise/scandinavia-lang-pack-2.1-1
Middle East	<ul style="list-style-type: none"> Arabic Greek English French Hebrew Turkish 	http://dl.google.com/enterprise/mid_east-lang-pack-2.1-1
Eastern Europe	<ul style="list-style-type: none"> German English French Hungarian Hebrew Polish Romanian Russian 	http://dl.google.com/enterprise/east_europe-lang-pack-2.1-1
egfisdfp	<ul style="list-style-type: none"> English German French Italian Spanish Dutch Finnish Polish 	http://dl.google.com/enterprise/egfisdfp-lang-pack-2.1-1.bin

Language Bundle	Languages	Download Link
Northern Europe + Eastern Europe + Extra languages	<ul style="list-style-type: none"> • English • German • Dutch • Swedish • Norwegian (includes both Bokmål and Nynorsk) • Danish • Finnish • Russian • Polish • Spanish 	http://dl.google.com/dl/enterprise/mea_north_eastern-lang-pack-2.1-1-1.bin

You can enable search appliance support for a language bundle by performing the following tasks:

1. Downloading and installing the language bundle
2. Activating the language bundle

You can install multiple language bundles on a search appliance, but only one language bundle can be active at any time. By default, the built-in language bundle is active.

The currently active language bundle provides spelling support for the languages in the bundle, as well as query expansion. The Google Search Appliance supports several types of query expansion, including:

- Contextual—alternative terms for the search term, for example, where the search term “latest apple” is expanded to include “apples,” “fruit,” and “ipod.”
- Non-contextual—replacement of the search term, for example, where the search term “apple” is expanded to include “apples.”

The search appliance supports contextual query expansion for all languages in the language bundles. Support for non-contextual query expansion is currently only available for Dutch, English, French, German, Italian, Portuguese, and Spanish.

You cannot delete the currently active language bundle. However, you can delete any other inactive language bundle that you no longer need.

To install and activate a language bundle, use the **Search > Search Features > Language Bundles** page in the Admin Console. For information about using this page, click **Admin Console Help > Search > Search Features > Language Bundles** in the Admin Console.

Enabling/Disabling Click-Jacking Defense

Click-Jacking (sometimes called UI Redress) is a type of web attack where an attacker modifies the user interface of a target web site so that a victim does not realize that he is taking an important action.

For example, a malicious web site could iframe an approval page for granting access to a third-party application. When a user visits the malicious web site, the site would overlay the approval button on the targeted site with a dancing hamster. When the user clicked on the hamster, the click would be processed by the targeted site. The user would unknowingly have granted access to the third-party application.

When the click-jacking defense is enabled the search appliance sends an `X-Frame-Options: SAMEORIGIN` header to prevent the iframe of search results pages.

Also, when click-jacking defense is enabled:


- The Admin Console Test Center is unable to display the “cached version” of the URLs returned there.
- Any Iframe-based application used by the customer to show results will, most likely, stop working if it's accessed by any of the following browsers:
 - Chrome 4.1.249.1042 +
 - Firefox 3.6.9 + (or earlier with NoScript)
 - Internet Explorer (IE) E8 and IE9
 - Opera 10.50 +
 - Safari 4 +
 - Others based on those engines (WebKit, Trident, Gecko)

By default, click-jacking defense is enabled.

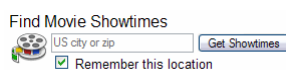
To enable/disable click-jacking defense:

1. Click **Search > Search Features > Query Settings**.
2. Under **Click-Jacking Defense Settings**, click the **Enable Click-Jacking Defense** checkbox.
3. Click **Save**.

Using OneBox Modules to Integrate Structured Results

 In some instances, the most relevant result for a search query is real-time, structured data. This type of data does not usually reside in the search index because it would be obsolete before it could be indexed.

For example, as a service to its users, *mediacompany.com* provides information about local movie times. When a user searches for “movies,” a specially formatted search box appears at the top of the search results, as illustrated in the following figure.



When a user enters a location in the search box, formatted, real-time data about theater show times appear, as illustrated in the following figure.

The image shows two search results for theater showtimes. The first result is for the movie "Outsourced", which has a 3-star rating and 3 reviews. It is a 43-minute comedy rated PG-13, shown at the Bellevue Galleria Stadium 11. The showtimes are 12:35, 2:55, 5:15, 7:35, and 9:55. The second result is for the movie "Music Within", which has a 4-star rating and 4 reviews. It is a 33-minute drama rated R, shown at the Regal Meridian 16. The showtimes are 12:25, 2:55, 5:15, 7:55, and 10:15. Both results include a "Map" link.

This type of result is served by a OneBox module. A OneBox module sends the user's query to a back-end or third party system and retrieves relevant data immediately. A OneBox module is returned when an end user's search term matches a "trigger" term. In the previous example, the trigger is "movies."

The search appliance supports two types of OneBox modules:


- Internal—Provides real-time access to data from a collection on the search appliance
- External—Provides real-time access to data from an external source, such as an application or database

You can develop OneBox modules with the use of simple APIs and a standard XML format. A OneBox module that has been integrated with the search appliance can be used with any front end on the search appliance. A front end can use an unlimited number of OneBox modules.

For detailed information about developing OneBox modules, refer to the following documents:

- *Google OneBox for Enterprise Developer's Guide*
- *Google OneBox for Enterprise Design Principles*

Using Result Biasing to Influence Result Ranking

 The Google Search Appliance ranks the documents that it finds in response to a user search query. For each document, the search appliance calculates a score that:

- Reflects the probable relevance of the document content
- Determines the order in which results appear on the search results page

You can use result biasing to increase or decrease the scores of specified sources, or types of sources, in the search index. These local settings may affect the order of the search results. You can influence results ranking by using:

- Source biasing—enables you to affect scores of documents that belong to a specified collection, match certain URL patterns, or are fed from a data source. See "Using Source Biasing" on page 72.
- Date biasing—enables you to affect scores of documents by date. See "Using Date Biasing" on page 73.
- Metadata and entity biasing—enables you to affect scores of documents based on metadata or entities associated with documents. See "Using Metadata and Entity Biasing" on page 74.

Result biasing is one way to affect the order of results. You might consider using other features to meet the following objectives:

- To ensure that a particular document appears at the top of the results, consider using a key match. For more information about this topic, refer to “Using KeyMatches to Guide Users to URLs” on page 43.
- To allow users to search only a certain set of documents, consider using a collection. For more information about this topic, see “Using Collections with Front Ends” on page 14.

Working with Result Biasing

To influence search appliance rankings, use a result biasing policy. A result biasing policy determines the source biasing, date biasing, and metadata and entity biasing settings that are used with a front end. A default result biasing policy (default_policy) is built into the search appliance. You can use default_policy, or create one or more custom result biasing policies. A result biasing policy is specific to a front end, so it can be aimed at specific types of end users.

To work with result biasing:

1. Create a result biasing policy using the **Search > Search Features > Result Biasing** page in the Admin Console.
2. Configure source biasing, date biasing, or metadata and entity biasing for the policy using the **Search > Search Features > Result Biasing > edit** page in the Admin Console. A menu-driven interface allows weak or strong increases or decreases, and requires no complex coding or scripting. You can use 11 settings to adjust result biasing from least influence to most influence.
3. Enable the result biasing policy by selecting it for use with a front end using the **Search > Search Features > Front Ends > Filters** page in the Admin Console.

If you are updating a search appliance with existing result biasing settings, your existing settings and any existing front ends that have result biasing turned on remain intact after the update.

For complete information about using the **Search > Search Features > Result Biasing** page, the **Search > Search Features > Result Biasing > edit** page and the **Search > Search Features > Front Ends > Filters** page, refer to the **Admin Console Help** in the Admin Console.

Using Source Biasing

Using source biasing, you can boost or weaken the relevancy score of a document that belongs to a specified collection, matches certain URL patterns, or is fed from a data source. Boosting a score usually moves a result up in the rankings. Weakening a score usually moves it down. The search appliance offers two controls over source biasing, as described in the following table.

Control	Description
Influence	Specifies how much influence source biasing has in calculating scores for results rankings overall. The default influence is No influence . You can change this to one of ten settings in a range of More influence for a search appliance.
Strength	Specifies how much influence source biasing has in calculating scores for documents that match specific URL patterns. The default strength is Leave unchanged . You can change this to Strong, Medium, or Weak increase or Strong, Medium, or Weak decrease for each collection or specified URL pattern.

For example, suppose most official content in your organization is drafted in Microsoft Word (.doc), but published in Portable Document Format (pdf). In some instances, both the .doc and .pdf versions of documents have the same content. However, each type of document is stored in a different location.

- Microsoft Word documents are stored on *www.mediacompany.com/drafts*
- PDF documents are stored on *www.mediacompany.com/publications*

Both types of documents are crawled and appear in the search index. However, you want .pdf versions to be higher in the search results than .doc versions. You create a result biasing policy where you can influence the score of .pdf documents. For this result biasing policy, you can use source biasing to:

- Boost *www.mediacompany.com/publications* by setting a strong increase
- Weaken *www.mediacompany.com/drafts* by setting a weak decrease

By adjusting the strength of individual URLs you influence the score that reflects the relevance of documents that match the URL patterns. After you select the result biasing policy for use with a front end, .pdf documents are more likely to appear closer to the top of results listings.

The effect of changing a document's score is not always predictable. The order of a search result among the other results is determined by many factors, including the scores of the other documents with which it is returned.

Using Date Biasing

Using date biasing, you can specify the influence date biasing has in increasing the scores of more recent documents relative to older documents. The default influence is **No influence**. You can change this to one of ten settings in a range of **More influence** for a search appliance.

For example, suppose the *mediacompany.com* search index contains all DVD reviews written during the past ten years. Typically, end users are interested in only searching the most recent reviews. However, staff is interested in searching for all reviews.

You have created a front end for end users and another front end for staff. In the front end for end users, you create a result biasing policy where you can configure date biasing to increase the rankings of new documents and lower the rankings of legacy documents. After you select the result biasing policy for use with the front end, documents with a more recent date are more likely to appear closer to the top of results listings.

In the front end for staff, you do not create a result biasing policy. This front end preserves normal ranking.

You can also specify how quickly documents' scores are decreased as they age, by selecting an age at which you would consider a document to be moderately old. This setting indicates that a document of the specified age should get a moderate decrease in score from date biasing. More recent documents get smaller date biasing effects, and documents older than that age get larger decreases in score.

For example, if you select "a day" as the age at which documents become moderately old, then documents that are a day old receive a moderate decrease in score from date biasing, and documents much older than that quickly get nearly the maximum effect of date biasing. On the other hand, if you choose "a year" as a moderately old age, then documents that are a year old get a moderate score decrease, those that are a few months old get their score decrease slightly, while documents that are a day old get very little date biasing effect.

In general, a good starting point for using this parameter is to think about some queries your users might issue and the types of pages they would get back. Consider how old a result would have to be before it is likely to be obsolete compared to pages authored or modified more recently. That age may be a reasonable setting for a document to be considered "moderately old," although experimenting with different settings is the best way to find out how it affects your search results.

The date biasing feature does not provide results that are ordered strictly by date, because other factors are considered in a document's score. To order results strictly by date, any user can click the **Sort by date** link.

Date biasing is applied to search results in a flexible manner. To experiment with the settings to see how it affects the documents in your index. You'll typically notice that date biasing affects the order of recent documents with small differences in date more than it affects older documents with small differences in date.

Which Document Dates Are Used for Date Biasing?

The document date that the Google Search Appliance evaluates for date biasing is specified on the **Index > Document Dates** page. For example, the search appliance can use one of the following dates:

- The last modified date of the document
- A date in the URL
- A date in the document title

However, if a document is added to the index by a feed rather than by a crawl, the document might not have a useful associated date for use with date biasing.

Using Metadata and Entity Biasing

Using metadata and entity biasing, you can boost or weaken the relevancy score of a document based on metadata or entities associated with a document. Boosting a score usually moves a result up in the rankings. Weakening a score usually moves it down. The search appliance offers two controls over metadata and entity biasing, as described in the following table.

Control	Description
Influence	Specifies how much influence metadata and entity biasing has in calculating scores for results rankings overall. The default influence is No influence . You can change this to one of ten settings in a range of More influence for a search appliance.
Strength	Specifies how much influence metadata and entity biasing has in calculating scores for documents that match specific META tag NAME-CONTENT value pairs. The default strength is Leave unchanged . You can change this to Strong, Medium, or Weak increase or Strong, Medium, or Weak decrease for each specified URL pattern.

For example, suppose that you have created a front end for your organization's human resources department. For users of this front end, you want to boost the score of HTML documents that include meta tags that indicate human resources as the author, as shown in the following example.

```
<META NAME="author" CONTENT="human_resources">
```

To bias metadata, create a result biasing policy called `human_resources` and configure metadata and entity biasing for this policy by entering the metadata name and content (value) pair and strength to apply to the score of any document that contains the metadata. After you select the `human_resources` policy for use with the front end, documents by `human_resources` are more likely to appear closer to the top of results listings. If you use metadata and entity biasing to weaken the score of documents by `human_resources`, those documents are likely to appear lower in the results listing.


If more than one meta data pair is configured, the document is tested against each metadata pair in turn until one matches. If more than one metadata pair matches, only the first one is used. In addition to working with metadata that is contained within HTML documents, metadata biasing also works with external metadata that is associated with documents.

If you have configured entity recognition, the search appliance discovers interesting entities in documents and stores them in the index. For example, suppose that the search appliance has discovered the term “human resources” in documents with poor or missing metadata. To bias this entity, you can configure metadata and entity biasing by selecting “human resources” from the pull-down menu and entering entity content.

The search appliance only applies metadata and entity biasing to the more relevant URLs, rather than all the URLs, that match the search query.

The effect of changing a document's score is not always predictable. The order of a search result among the other results is determined by many factors, including the scores of the other documents with which it is returned.

Providing Alerts for End Users

 Some users might want to monitor topics of interest by receiving search results for these topics in email messages. You can allow users to monitor topics this way by enabling alerts. Alerts are email updates of the latest relevant search results based on a user's topic of interest.

A user sets up an alert by clicking **My Alerts** on the search page (shown in the following figure) and then logging in to the search appliance.

On the **Manage your Alerts** page, the user can create an alert by identifying search terms that will return relevant results and a frequency of searches. After the user creates an alert, the search appliance sends the user an email whenever it finds new or changed documents about the topic of interest. The following figure shows the **Manage your Alerts** page.

For example, suppose a user is interested in changes to personnel policies. He might create an alert for this topic. The search appliance will run searches on all the publicly shared documents and send an email to the user whenever it finds new or changed publicly-shared documents about changes to personnel policies (secure results are not returned). The email contains a batch of result listings. Clicking any result listing in an alert email displays the relevant document.

To provide alerts for users, you must:

1. Configure an authentication mechanism for the search appliance to use to authenticate the user. If the configured authentication mechanism does not provide the user's email address, then the user can set the email address after logging into the **My Alerts** page.

For information about authentication mechanisms, see “Authentication,” in *“Managing Search for Controlled-Access Content.”*

2. Specify rules for document dates by using the **Index > Document Dates** page or ensure that the rules are working.

To verify that document dates rules are working, perform a search and ensure that dates are associated with each result.

3. Enable alerts for the search appliance (see “Enabling Alerts” on page 76).
4. Show the **My Alerts** link for a specific front end (see “Showing the My Alerts Link” on page 76).

Alert emails use the **sender of outgoing emails** that is specified in the **Email Notification** area on the **Administration > System Settings** page during search appliance installation. If the **sender of outgoing emails** is not specified during installation, the default value of nobody@localhost is used. For information about specifying the value of **sender of outgoing emails**, refer to Configuring the Network Settings in *Installing the Google Search Appliance*.

Enabling Alerts

If alerts are not currently enabled for a search appliance, you can enable them using the **Index > Alerts** page in the Admin Console (see “Providing Alerts for End Users” on page 75). If alerts are enabled, you can disable them.

To enable alerts for a search appliance:

1. Choose **Index > Alerts**.
2. Click **Enable**.

To disable alerts for a search appliance:

1. Choose **Index > Alerts**.
2. Click **Disable**.

Showing the My Alerts Link

The Google Search Appliance supports two methods of showing the **My Alerts** link on a search page:

- “Using the Page Layout Helper” on page 76
- “Using the XSLT Stylesheet Editor” on page 77

You cannot use the Page Layout Helper to show the **My Alerts** link if you have customized the XSLT stylesheet. If you have done this, you must show the **My Alerts** link directly in the XSLT stylesheet.

If you want to use a customized XSLT stylesheet from an earlier release, refer to the update instructions for the current search appliance software release.

Using the Page Layout Helper

To show the **My Alerts** link using the Page Layout Helper:

1. Choose **Search > Search Features > Front Ends**.
2. Select a front end from the **Current Front Ends** list and click **Edit**.
3. In the **Page Layout Helper** box, select the **Global Attributes** section.
4. To show the **My Alerts** link, select **Show Alerts Link**.
To hide the **My Alerts** link, deselect **Show Alerts Link**.
5. Click **Save**.

Using the XSLT Stylesheet Editor

This section describes showing the **My Alerts** link using the XSLT Stylesheet Editor. If you prefer, you can edit the XSLT stylesheet in another editor.

To show the **My Alerts** link in an XSLT stylesheet:

1. Choose **Search > Search Features > Front Ends**.
2. Select a front end from the **Current Front Ends** list and click **Edit**.
3. In the XSLT Stylesheet Editor, scroll to the alerts2 options in the Other Variables section. The following code shows the alerts2 options.

```
<!-- *** alerts2 options *** -->
<xsl:variable name="show_alerts2">0</xsl:variable>
```

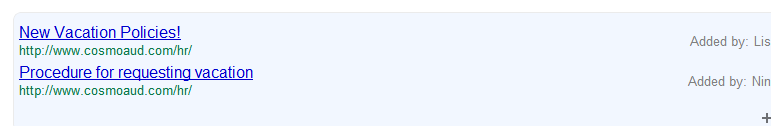
4. Show or hide the **My Alerts** link:
 - To show the link, change the show_alerts2 value from 0 (zero) to 1 (one).
 - To hide the link, change the show_alerts2 value from 1 (one) to 0 (zero).
5. Click **Save**.

Providing User Results

You can give users the capability of enhancing the search experience collaboratively by adding search results for certain keyword searches. User results appear for the specified keyword searches on the search results page of a specific front end.

For example, suppose a user wants a document about your organization's new vacation policies to appear on the results page when anyone searches using the keyword "vacation." To accomplish this, the user can create a result for the document.

When anyone searches on "vacation," the user result always appears on the results page. The result displays the title, the specified URL, and the user's name, as shown in the following example.



Duplicate results are not displayed. So for a given query if any newly submitted result has the same URL as a pre-existing result, then it is not returned in the search page.

To provide user results capability, add one or more user results configurations. Google recommends enabling query suggestions (see "Providing Query Suggestions" on page 79) with user results so that they can appear as query suggestions.

Adding a User Results Configuration

To add a user results configuration, use the **Search > Search Features > User Results** page.

For each configuration, you can specify:

- A name for the configuration.
- A description of the configuration.
- Whether user results are moderated, that is, if they require administrator approval before appearing in search results, and which front ends use the configuration.
- The front ends to associated with the configuration

Because a user result configuration can be associated with one or more front ends, you can create multiple configurations with different settings and associate them with separate front ends.

For complete information about adding a user results configuration, refer to the **Admin Console Help > Search > Search Features > User Results** page in the Admin Console.

Moderating User Results

By using the **Search > Search Features > User Results** page, you can moderate new or existing user results. When there is one or more new user results to validate, a **Validate** link on the page is highlighted in red and shows the number of results awaiting validation.

You can moderate either all user results at once or one or more individual results.

Enabling Authentication for User Results

When you enable authentication for user results, a user must be properly authenticated with a verified identity before the user can add, edit, or remove user results.

If authentication for user results is enabled, and the user is not logged in with a proper verified identity, the user cannot add, edit, or delete user results. If authentication for user results is not enabled, the user is not required to be properly authenticated before adding, editing, or removing them.

To enable authentication for user results:

1. Choose **Search > Secure Search > Access Control**.
2. Click the check box for **Require authentication for User Results**.
3. Click **Save**.

To disable authentication for user results:

1. Choose **Search > Secure Search > Access Control**.
2. Clear the check box for **Require authentication for User Results**.
3. Click **Save**.

How Users Can Add Results

To add a result, a user performs the following steps:


1. Type the keyword for the user result in the search box and click Search. For example, the keyword is "vacation."
2. On the search results page, click the + icon.

3. Type a title for the result. For example, the title might be "New Vacation Policies!"
4. Type the URL of the document.
5. Type a user name.
6. Click Save.

If you, as the search administrator, are not moderating search results, the user result begins to appear immediately for the appropriate keyword searches.

If you are moderating user results, a confirmation box appears telling the user that the new result requires your approval. To submit the result, click **OK**. After you approve the result, it begins to appear for the appropriate keyword searches.

Providing Query Suggestions

 You can help users improve searches by enabling query suggestions for a new or existing front end. When query suggestions are enabled, search queries auto-complete and query suggestions appear as a user types in the search box.

For example, suppose a user starts typing "Google Search Appliance" in the search box. Query suggestions causes the term to auto-complete before the user finishes typing it. Alternative terms that narrow the search also appear in a menu below the search box. For example, for this search, terms that appear might include "Google Search Appliance training," "Google Search Appliance documentation," "Google Search Appliance Forum," and other similar terms. Query suggestions can include user results (see "Providing User Results" on page 77), which are displayed in italic font style. The user can execute a search by selecting one of the terms.

To provide query suggestions:

1. Choose **Search > Search Features > Front Ends**.
2. Create a new front end or select an existing front end from the **Current Front Ends** list and click **Edit**.
3. Under **Page Layout Helper**, select the **Search Box** section.
4. Check the **Enable auto- completion in search box** check box.
5. Click **Save**.

Query suggestions are created by using the terms entered in previous searches. Only queries that returned results are used to build the database of query suggestions.

Each query suggestion entry is tagged under a unique front end and collection pair based on the previous searches. Query suggestion requests must be made against the same front end and collection pair to get suggestion results. Queries against multiple collections (using AND or OR boolean operators) are not made available as suggestions.

The search appliance automatically refreshes query suggestions every 24 hours, so it might be a day before a new query suggestion is available. However, you can manually trigger a refresh of query suggestions by toggling suggestion off and on again for the front end.

The search appliance keeps query suggestions for 90 days. If you delete the matching document from the index or reset the index, the query suggestion continues to appear, even though the suggestion does not return results.

To refresh query suggestions:

1. Uncheck the **Enable auto-completion in search box** check box.
2. Click **Save**.
3. Recheck the **Enable auto-completion in search box** check box.
4. Click **Save**.

Exporting Query Suggestions

You can see which query suggestions the search appliance is using by exporting them. When you export query suggestions, the search appliance creates a file that contains an inventory of all query suggestions. Using the options on this page, you can export query suggestions for:

- Collections—All collections or just a selected collection
- Front ends—All front ends or just a selected front end

You can also select one of the following options for the format of the exported query suggestions file:

- xml—Provides a structured inventory of query suggestions
- txt—Provides an unstructured inventory of query suggestions, with one suggestion per line

To export query suggestions:

1. Click **Search > Search Features > Suggestions**.
2. Select a collection from the menu or accept All Collections.
3. Select a front end from the menu or accept All Front Ends.
4. Select a file format from the menu.
5. Click **Export**.
6. Open or save the file.

Exporting and Importing the Suggestions Blacklist

The search appliance also has a suggestions blacklist, which contains words that are excluded from query suggestions. If you want to change the blacklist, you can use the options on the **Search > Search Features > Suggestions** page to perform the following tasks:

- Exporting the suggestions blacklist.
- Adding, editing, or deleting entries.
- Importing the edited suggestions blacklist to the search appliance.

You can export the search appliance's suggestions blacklist by using the options under **Export Suggestions Blacklist**. To export a suggestions blacklist

1. Click **Export**.
2. Open or save the file.

You can import a suggestions blacklist by using the options under **Import Suggestions Blacklist**. To import a suggestions blacklist:

1. Enter the filename of a suggestions blacklist file or click **Choose File** and navigate to the correct file.
2. Click **Import**.

The imported suggestions blacklist overwrites the existing one. To remove the existing blacklist, upload an empty file.

A partial match is used for entries in the suggestions blacklist. For example, an entry for `ray` in the blacklist blocks `ray` and `raymond`. To block only `ray`, use regular expression `^ray$`.

Providing Document Previews

Document previews enable users to view a thumbnail image of a document in the search results. To view a document preview, the user hovers the pointer over the search result. The preview appears, as shown in the following figure.



The search appliance supports document previews for the following formats:

- Microsoft Word (doc, docx)
- Microsoft PowerPoint (ppt)
- Adobe Portable Document Format (pdf)

However, document previews are not available for .doc, .pdf and .ppt files in zip files.

To provide document previews to your users, perform the following tasks:

1. Enabling the document preview module by using the **Search > Search Features > Document Preview Module** page.
2. Showing document previews in a front end by using the **Page Layout Helper** on the **Output Format** tab of the **Search > Search Features > Front Ends** page.

For complete information about providing document previews, click **Admin Console Help > Search > Search Features > Document Preview Module** in the Admin Console.

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

Enabling Wildcard Search

Wildcard search enables your users to search by entering a word pattern rather than the exact spelling of a term. The search appliance supports two wildcard operators:

- *--Matches zero or more characters
- ?--Matches exactly 1 character

The search appliance is able to consider all words with * as wildcard terms, so users don't need to prepend the wildcard: special operator to a pattern that contains this operator. To enable the search appliance to do this, click the **Consider words with * as wildcards by default** checkbox on the **Search > Search**.

Using wildcards can simplify queries for long names, technical data, pharmaceutical information, or strings where the exact spelling varies or is unknown. A user can search for all words starting with a particular pattern, ending with a particular pattern, or having a particular substring pattern. A wildcard query term must satisfy at least one of the following conditions:

- A sequence to at least 2 characters at the start of a word, for example: go*
- A sequence to at least 2 characters at the end of a word,, for example: *le
- A sequence of at least 3 characters anywhere in the word, for example: *ear*

To search using a wildcard pattern, prepend the wildcard: special operator to a pattern. For example: wildcard:test* and wildcard:?nter.

For more information about the wildcard: special operator, see the Search Protocol Reference. Wildcard search is also supported for metadata queries, but the wildcard: special operator is omitted. For example: inmeta:name*. Also, metadata queries are %-encoded, which affects the form of an inmeta: wildcard query.

To enable wildcard search:

1. In the Admin Console, click **Search > Search Features > Front Ends > Filters**.
2. In the **Maximum expansion per wildcard term** text box enter a value from 0 (disable wildcard search) to 1000.
3. Optionally, click the **Consider words with * as wildcards by default** checkbox
4. Click **Save**.

Wildcard search is not supported for other common queries, including filetype, inurl, intext, and so on. Also, wildcard search is not supported with Chinese, Japanese, Korean, or Thai languages.

For complete information about enabling wildcard search, click **Admin Console Help > Search > Search Features > Front Ends > Filters** in the Admin Console.

Tuning Search Results

Serving quick and relevant search results helps ensure user satisfaction with the search experience. Ways that you, as an administrator, can help maintain results include:

- Optimizing the Speed of Results
- “Analyzing Searches that Do Not Return Results” on page 84

Optimizing the Speed of Results

Studies show that you can lose more users due to poor response time than to poor relevancy. If you find that performance is less than optimal, perform the tasks described in the following table.

Task	Comments
Make sure you have the right search appliance model for handling your query load.	If your search appliance model is not appropriate for your query volume, you might consider upgrading to a higher-capacity model.
Ensure that your user interface is uncluttered and free from large images or backgrounds, and that it does not contain non-search related server calls.	Any of these items can slow down response time. If any of these items are present, consider removing them from the user interface.
Check whether the volume of search queries is affecting response time.	If the volume of search queries is affecting response time, consider getting a second search appliance and load balancing the two search appliances.
Determine whether your search appliance is trying to crawl more documents than your license limit allows.	If your search appliance is trying to crawl more documents than your license limit allows, consider the following options: <ul style="list-style-type: none">• Whitelist URLs for crawling, as described in Using Only a Whitelist of URLs.• Increase the license limit of the search appliance. For information about increasing the license limit of your search appliance, contact Google for Work Sales.
Determine whether security checks are taking too long.	Before returning secure results to a user, the search appliance queries a secure server, such as an LDAP, SAML, or NTLM server, for user authorization. In some instances, the secure server's performance may compromise response time. If so, optimize the secure server.
Monitor your network traffic to find out if it is especially high.	Other devices on your network may be slowing down search appliance performance.

Analyzing Searches that Do Not Return Results

Sometimes, user search queries fail to return available information. In these instances, you can use front end configuration features such as KeyMatch (see “Using KeyMatches to Guide Users to URLs” on page 43), related queries (see “Using Related Queries to Suggest Alternative Searches” on page 45), and query expansion (see “Using Query Expansion to Widen Searches” on page 60) to provide better results.

To identify search terms that do not return results, generate a search report, as described in “Generating a Report of Unsuccessful Search Queries” on page 85. The report on searches that do not return results contains lists of top keywords and top queries that do not return results. The following example shows Top Keywords and Top Queries in a search report.

Top Keywords	#Occurrences
plans	920
retirement	812
savings	809
tax	667
pre	609
401K	511
options	332
employee	276
benefits	255

Top Queries	#Occurrences
401K	512
pre-tax savings plans	423
retirement	411
retirement plans	358
savings plans	358
retirement options	332
employee benefits	287
pre-tax plans	232
retirement savings	166

Notice that “pre-tax plans,” appears for 232 queries that did not return results. To suggest alternative search terms, create a related query, as shown in the following example:

You could also try: [401K Pre-Tax Plans](#)

To guide end users to the URL where the information resides, create a KeyMatch, as shown in the following example:

[Latest changes in company 401K options](http://www.montebre.com/2007Plans.html) KeyMatch
http://www.montebre.com/2007Plans.html

Generating a Report of Unsuccessful Search Queries

When you generate a report of unsuccessful search queries, you specify the following information:

- Collection for the report
- Name for the report
- Time frame for the report (optional)

For example, you might create a report of unsuccessful search queries called *Operations Zero Results Report* on the default_collection for the previous week.

To create a report of unsuccessful search queries:

1. Choose **Reports > Search Reports**.
2. From the **Show Search Reports for Collection** drop-down menu, select the collection whose search queries you want to include.
3. Under **Define Search Report**, type a name for the report.

Give the report a name of up to 20 alphanumeric characters, hyphens, and underscores. The name cannot start with a hyphen.
4. Under **Report type**, click **Searches that did not return results**.
5. Under **Report timeframe**, specify the period that you want the report to cover.
6. If you run scripts that create diagnostic data by generating test queries, exclude the diagnostic search terms by entering them under **Diagnostic terms to exclude**.
7. By default, the report shows the top 100 keywords and queries. At **Number of top queries and keywords to show**, change the number if you prefer more results or fewer results.
8. Click **Generate Report**.

The report appears under **List of search reports**, with its status set to Generating. The page does not automatically update when the report is complete, so refresh the browser page to check the report's status.

To view the report, click **View** under **List of search reports**.

Managing the Search Index

A search index provides the foundation for a positive search experience. As an administrator, you can perform the following tasks to ensure that the index continues to support search effectively:

- “Keeping the Search Index Clean” on page 86
- “Segmenting Data in the Search Index” on page 86

Keeping the Search Index Clean

You can help improve user searches by making sure that your index is as “clean” as possible. A clean index contains valuable documents and does not contain unnecessary documents. You can ensure a clean index when you set up the URLs for crawling your content on the **Content Sources > Web Crawl > Start and Block URLs** page in the Admin Console.

Generally, there are two methods for setting up URLs for crawling your content:

- Using a whitelist of URLs. List all possible URLs at your company in the **Start URLs** section.
- Using a blacklist of URLs. List unwanted URLs in the **Do Not Follow Patterns** section.

To keep the search index clean, you can use both methods together or simply use a whitelist. Generally, using only a whitelist is more effective than using only a blacklist. The following sections describe the advantages and disadvantages of both approaches. For information about setting up URLs for crawling your content, refer to *Preparing Data for a Crawl in Administering Crawl*.

If you have a search appliance that you use for testing, test your crawl patterns first and then deploy them if the quality of the search results has improved.

Using Both a Whitelist and a Blacklist of URLs


You can set up a crawl with a whitelist of URLs. After the search index has been created, you can prune unwanted URLs from the search index using a blacklist. Starting from the comprehensive list of start URLs and then pruning them has the benefit of ensuring that the crawler has found every document in your organization.

Using Only a Whitelist of URLs

An alternate approach is to set up a crawl with a whitelist that contains only URLs that you know to be valuable. This doesn't mean you should get too specific and place specific low-level folders and documents in the Crawl list, but it does mean you should be cautious of what a large root node might bring into the index.

The benefit of this approach is that the index will not be bloated to include documents that may be unnecessary to index. The disadvantage of this approach is you need to be especially careful to include every start URL that might be of value.

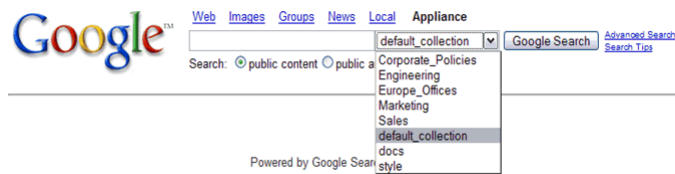
Segmenting Data in the Search Index

 User searches can be more efficient when they are restricted to subsets of the entire search index. As an administrator, you can help users to search more efficiently by using collections. A collection is a segment of the complete search index that you define by specifying URL patterns to include.

Using collections, you can show different results to different users. For example, suppose you want to create different collections, such as:

- “Engineering,” for technical users and other user who need to search for engineering documents
- “Sales,” for sales staff to search for sales documents
- “Marketing,” for marketing staff to search for marketing documents
- “Corporate Policies,” for any staff to search for policy documents
- “Europe Offices,” for users who are geographically located in the European offices

To search a collection, a user can select a collection from a pull-down menu on the search box, as illustrated in the following figure.



For more information about this topic, refer to [Adding a Menu to Search by Collection in Customizing the User Interface](#).

The maximum number of collections for a search appliance is 200. Having more than this number of collections can cause serving to fail. The solution for this problem is to reset the index.

The search appliance allows you to create an unlimited number of collections for a search index. To create a collection, use the **Index > Collections** page in the Admin Console. For information about using this page to create a collection, see **Admin Console Help > Index > Collections**.

Searching Collections

Individual collection search results have the same relevance ranking as full index searches. Only the content searched differs as it is restricted to the individual collection's content.

To restrict searches to a collection that you have defined, add the following to the URL of your search query:

```
&site=COLLECTION_NAME
```

The following examples show search queries for searching collections.

A search for “vacation” in the collection Human_Resources:

```
http://www.google.com/search?
q=vacation&output=xml&client=yoursite&site=human_resources
```

This search returns vacation results specifically from URLs in the Human_Resources collection.

A search for “product” in the collections Development and Marketing:

```
http://www.google.com/search?q=product&output=xml&client=yoursite&site=
(development)| (marketing)
```

This search for “product” returns results from either the Development or Marketing collections.

For more information, see the [Filtering](#) section of the *Search Protocol Reference*.

Enabling Users to Search Multiple Collections

Once you have created collections, determine if users will ever want to search multiple collections at the same time. For example, you might want to enable a sales person to search the “Marketing” collection and the “Sales” collection. The Google Search Appliance allows search within multiple collections by using the “|” (OR) or “.” (AND) operator.

To specify the collection that you want to search, you set the site parameter on the GET request. The following example shows a GET request where the collection specified is one named "all_content."

```
http://search.corp.mycompany.com/search?q=query+string
&site=all_content
&client=default_frontend
&output=xml_no_dtd
&proxystylesheet=default_frontend
```

To search for terms that appear in either the "sales" or "marketing" collection, use the boolean OR [|] operator on the site parameter. The following example shows a GET request that specifies both collections.

```
http://search.corp.mycompany.com/search?q=query+string
&site=sales|marketing
&client=default_frontend
&output=xml_no_dtd
&proxystylesheet=default_frontend
```

To search for terms that appear in both the "engineering" and "support" collections, use the boolean AND [.] operator on the site parameter. The following example shows a GET request that specifies both collections.

```
http://search.corp.mycompany.com/search?q=query+string
&site=engineering.support
&client=default_frontend
&output=xml_no_dtd
&proxystylesheet=default_frontend
```

Once you have exposed these multi-select collections in your UI, make sure that the users know that they can segment a search to get better relevancy.

For complete information about using search parameters, refer to the *Search Protocol Reference*.

Evaluating Search Performance

The most effective way for you to create an appropriate search experience is to focus on the end user. To gain insight into your user's opinions about search, you can solicit their feedback. Ways to solicit feedback from end users include:

- "Adding a Feedback Mechanism for Users" on page 88
- "Conducting a User Satisfaction Survey" on page 89

In addition to getting feedback from end users, you might want to exchange ideas with other search appliance administrators or members of the Google for Work team. For information on this topic, refer to "Exchanging Information on Google Groups" on page 90.

Adding a Feedback Mechanism for Users

The number of user searches tends to increase when users can provide feedback about the quality of results. User feedback is also an effective way to identify usability issues. When you solicit feedback, you let users know that their opinions about the search experience are important. As users see improvements based on their feedback, they search more.

Some easy ways you can enable users to give feedback might include:

- Adding an email link to the search results page. For example, you might label the link “Email us if you can’t find what you’re seeking.”
- Adding a button on the search results page. For example, you might label the button “I didn’t find what I was looking for.” Logging the percentage of queries that result in a click of this button provide an easy way for you to see whether you are improving the search experience.

Feedback that is sent with the user information is very useful. It enables you to contact users to get clarification and show that you are following up with them.

Conducting a User Satisfaction Survey

A survey can be an effective way of soliciting feedback from users about the search experience. A survey also enables you to establish a baseline of search quality metrics before beginning to implement best practices.

Some tips for creating a user satisfaction survey are:

- Keep it brief.
- Don't make the respondents fill out long essay questions.
- Keep questions as quantitative as possible so results are easily aggregated and compared to later survey results.
- Get enough participants to make the results meaningful and statistically significant.

The following example questions might help you develop your own survey questions.

1. How often do you find what you’re looking for?
 - 100% of the time
 - 80% of the time
 - 50% of the time
 - 20% of the time
 - Never
2. How often does your result show up in the top 10 (first page)?
 - 100% of the time
 - 80% of the time
 - 50% of the time
 - 20% of the time
 - Never
3. How often does your result show up as the first result?
 - 100% of the time
 - 80% of the time
 - 50% of the time
 - 20% of the time
 - Never

4. How often do you click on the shaded key matches at the very top of the results?
 - Whenever I see one
 - Sometimes
 - Never
5. How is the query response time?
 - Excellent
 - Sufficient
 - Unacceptable
6. Is the query result description (snippet) useful?
 - Definitely
 - Somewhat
 - No
7. How satisfied are you with your overall search experience?
 - Very satisfied
 - Somewhat satisfied
 - Not satisfied
8. What would you like to see added?

Finally, an effective way to evaluate the quality of search results is by using side-by-side evaluations. You could use this type of evaluation to test the following types of changes:

- Turning on/off certain collections (see “Segmenting Data in the Search Index” on page 86)
- Dynamic result clusters (see “Using Dynamic Result Clusters to Narrow Searches” on page 47)
- Query expansion (see “Using Query Expansion to Widen Searches” on page 60)
- Result biasing (see “Using Result Biasing to Influence Result Ranking” on page 71)
- Adding/removing host crowding (see “Experimenting with Host Crowding Options” on page 56)
- Testing deployment of a new version

Exchanging Information on Google Groups

Google wants you to get all possible value from your Google Search Appliance. An effective way to do this is to join the Google Search Appliance Discussion Forum, <http://groups.google.com/group/Google-Search-Appliance>. This group is a discussion forum where you can post questions, feedback, or advice for other users. The group also provides access to a knowledge base and useful files for administering a Google Search Appliance.

Members of the Google Search Appliance group includes other Google Search Appliance customers, administrators, and users. Members of the Google Search Appliance product, engineering, and support teams monitor the groups and occasionally provide assistance to other members.

Updating Your Search Appliance

There are usually many performance improvements, security enhancements, and/or features that are included in new Google Search Appliance software releases.

Unless Google for Work Support organization has advised you not to upgrade, you should always download the latest software release. If you are a current Google for Work customer, you can:

- Link to the latest software versions at the Google for Work Support site, <https://support.google.com/googleforwork/answer/142244#search>.
- Find more information on the new features in the *Guide to Software Release 7.4*.

Chapter 3


Customizing the User Interface

The Google Search Appliance has features that enable system administrators to enhance the search experience for end users. This chapter describes how system administrators can modify web pages in the Google Search Appliance user interface.

What Is the Google Search Appliance User Interface?

For an end user, the user interface is the means by which she interacts with the Google Search Appliance. The search appliance user interface is made up of the web pages that are listed in the following table.

Page	Description
Search page	Page where an end user enters terms and starts a search
Search Results page	Page where search results are returned to an end user
Advanced Search page	Page where an end user enters complex search criteria and starts a search
Search within Results page	Page where an end user enters terms to search within results
Cached page header	Message that appears at the top of a cached page to indicate that it's a cached version of the page

 By default, the search appliance offers a user interface that is simple and intuitive, like Google.com. As a search appliance administrator, you can use the default user interface as the basis for customizing one or more user interfaces that focus on your end users.

For example, suppose your organization plans to use a search appliance to serve both customers and employees in several countries in North and Latin America. You have identified several design goals for the user interface. For both types of end users, you want a user interface that reflects your organization's visual identity by using your logo and your color scheme.

For customers, you want a user interface that:

- Offers simple search options consisting of a search box and button
- Displays simple result listings that contain only a title and snippet

For employees, you want a user interface that:

- Offers multiple search options, such as searching secure content and searching within results
- Displays complex result listings that contain a title, a snippet, a date, a link URL, and a cached link

For both customers and employees, you want to offer user interfaces in four languages: English, Spanish, Brazilian Portuguese, and French. To accomplish all your design goals, you plan to create eight user interfaces.

For customers, you will create the following user interfaces:

- An English-language user interface for customers in the U.S. and Canada
- A Spanish-language user interface for customers in Mexico, Argentina, Uruguay, and the U.S.
- A Brazilian Portuguese-language user interface for customers in Brazil
- A French-language user interface for customers in Quebec, Canada

For employees, you will create the following user interfaces:

- An English-language user interface for employees in the U.S. and Canada
- A Spanish-language user interface for employees in Mexico, Argentina, Uruguay, and the U.S.
- A Brazilian Portuguese-language user interface for employees in Brazil
- A French-language user interface for employees in Quebec, Canada

This document describes how you can use Google Search Appliance features to accomplish these types of user interface customizations. The following table gives an overview of the major sections in this document.

Section	Describes
"Getting Started with Customizing the User Interface" on page 96	How you can customize the user interface using the Page Layout Helper
"Customizing the User Interface in the XSLT Stylesheet" on page 105	How you can customize the user interface using the eXtensible Stylesheet Language Transformations (XSLT) Stylesheet Editor
"Customization Process Overview" on page 113	What steps you should take to customize the user interface
"User Interface Design Principles" on page 116	What guidelines you should follow when customizing the user interface

What Is the Relationship Between a User Interface and a Front End?

A Google Search Appliance user interface is associated with a single front end. A front end is a framework that is composed of multiple elements that manage a single search experience. One of the elements of a front end is the format element, which determines how results are displayed to the user, or in other words, the user interface. The other elements of the front end define what results are available to present to the user. Each front end can have one user interface.

The search appliance has a default front end, which has a default user interface. You can use the default user interface without any customization. However, a search appliance can have multiple front ends, each with its own, customized user interface.

If you customize a user interface, you should create another front end for customization rather than customizing the default front end.

For more information about front ends, refer to “Managing the Search Experience” on page 14.

You can use a customized front end with a specific collection to help improve searches and enhance results, as described in “Using Collections with Front Ends” on page 14.

Creating a Front End

You create a front end using the **Search > Search Features > Front Ends** page in the Admin Console.

To create a front end:

1. Click **Search > Search Features > Front Ends**.

The **Front Ends** page appears.

2. Enter a name for the front end in the **Front End Name** box.
3. Click **Create New Front End**.

The new front end appears in the **Current Front Ends** column.

To edit the front end, click the **Edit** link for the front end.

Using Front End Search Parameters

When an end user searches for information using a front end, the Web browser sends the search request to the Google Search Appliance as a URL that contains a query string. In this query string, the front end's user interface is defined by the `&proxystylesheet` search parameter. Other elements of a front end are defined by the `&client` search parameter. You can mix and match the `&proxystylesheet` and `&client` search parameter values to accommodate different presentations of the same results. For complete information about using search parameters, refer to the *Search Protocol Reference*.

What Is the Relationship Between a User Interface and an XSLT Stylesheet?

When a search query is sent to the Google Search Appliance, the results are returned in XML. Results in XML format can be difficult for end users to read. To format results, the XML is parsed along with an XSLT stylesheet. The formatted results are more usable than raw XML results. An XSLT stylesheet contains information about which elements should appear in the user interface and how the elements should look. Each front end can use the same stylesheet or a different stylesheet. Each search appliance front end has a default XSLT stylesheet, which can be used with any front end.

For more information about how a search appliance uses XSLT stylesheets, refer to Search Experience Background.

What Tools Can I Use to Customize a User Interface?

You customize a user interface by editing variables in the XSLT stylesheet. There are two different ways that you can modify the XSLT stylesheet:

- Using the Page Layout Helper in the search appliance
- Using the XSLT Stylesheet Editor in the search appliance or an editor of your choice

The Page Layout Helper enables you to change only some of the elements in the XSLT stylesheet. The XSLT Stylesheet Editor enables you to make more extensive changes to the XSLT stylesheet. If the elements that you want to change are not available in the Page Layout Helper, you must use the XSLT Stylesheet Editor to change them. The search appliance supports XSLT 2.0 and XPath 2.0.

You may want to start by customizing the user interface using the Page Layout Helper. After you finish making and saving changes in the Page Layout Helper, you can, if you wish, make further changes in the XSLT Stylesheet Editor. Any changes that you make with the Page Layout Helper appear in the XSLT stylesheet.

However, after you save changes in the XSLT Stylesheet Editor, you cannot return to using the Page Layout Helper. It is automatically disabled. For more information about using both tools to customize a user interface, refer to “Customization Process Overview” on page 113.

For information about the Page Layout Helper, refer to “Working with the Page Layout Helper” on page 96. For information about the XSLT Stylesheet Editor, refer to “Working with the XSLT Stylesheet Editor” on page 106.

Changes that you make using the Page Layout Helper are fully supported by Google for Work Support. If you want to contact support about changes made using the Page Layout Helper, file a help ticket. You can also refer issues to the Google Search Appliance group on Google Groups. Changes that you make using the XSLT Stylesheet Editor are not supported by Google for Work Support. If you have issues about changes made using the XSLT Stylesheet Editor, you can refer them to the appropriate Google Group.

Another source of tools for user interface development is the Google Gadgets page (<https://developers.google.com/gadgets/>). You might consider creating a Google Gadget as a user interface for enterprise search. For example, you might create a Google Gadget search box for end users to add to their desktops. This gadget would enable end users to start a search from the search box on the desktop rather than the search page.

Google Gadgets are not supported.

What Knowledge Do I Need to Customize a User Interface?

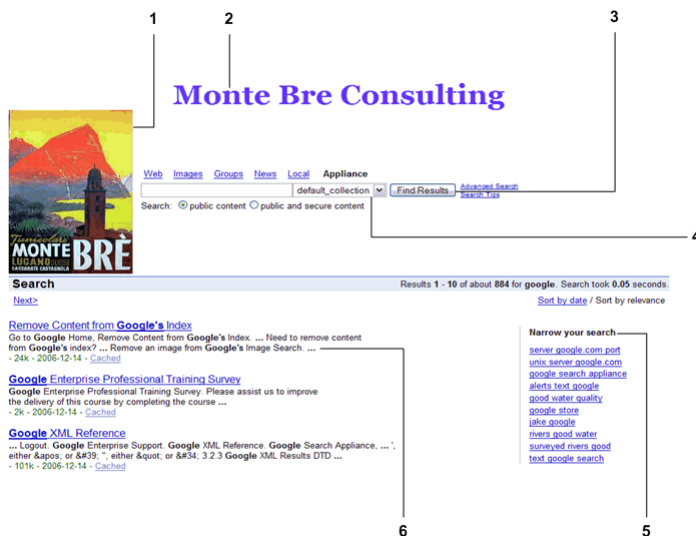
Even if you do not have any special knowledge of XSLT, you can effectively customize a Google Search Appliance user interface using the Page Layout Helper. However, if you want to add a custom header or footer (see “Adding a Header and Footer” on page 98) to your user interface using the Page Layout Helper, you add snippets of HTML code. In this instance, some knowledge of HTML is required.

If you want to make extensive changes to a search appliance user interface, you need to work directly in the XSLT Stylesheet. In this instance, knowledge of XSLT, XML, and HTML are required.

Getting Started with Customizing the User Interface

The default user interface for the search appliance includes Google-specific elements, such as the Google logo and the Google Search button. For illustrations of the default user interface, refer to “Starting with a Basic Search Experience” on page 9.

With a few simple changes, you can make a user interface that is specific to your organization by replacing Google elements with your own. The following figure illustrates a search results page that has been customized using the Page Layout Helper.



For descriptions of the customizations in this figure, refer to the key numbers in the following table.

Key	Customization
1	Replaced the Google logo with the organization's logo.
2	Added the header used on the organization's web site.
3	Changed search button text from Google Search to Find Results .
4	Added a menu to search by collection.
5	Added dynamic result clusters.
6	Removed URLs from result snippets.

Working with the Page Layout Helper

Without working directly with the XSLT stylesheet, you can use the Page Layout Helper to customize the user interface. All of the elements that you can change using the Page Layout Helper can also be changed using the XSLT Stylesheet Editor.

The Page Layout Helper has the following three sections:

- Global Attributes (see “Customizing Global Attributes” on page 97)
- Search Box (see “Customizing the Search Box” on page 99)
- Search Results (see “Modifying Search Results” on page 101)

The following procedures describe how to change each element in a user interface individually. However, you can also open one or all of the Page Layout Helper sections and make changes to all the elements at once.

Opening the Page Layout Helper

The Page Layout Helper appears on the **Search > Search Features > Front Ends > Output Format** page in the Admin Console. Before you can use the Page Layout Helper, you must create a front end. For information, refer to “Creating a Front End” on page 94.

To open the Page Layout Helper:

1. Choose **Search > Search Features > Front Ends**.
The **Front Ends** page appears.
2. Select a front end from the **Current Front Ends** list and click **Edit**.
The **Output Format** page appears.
3. In the **Page Layout Helper** box, select the section that you want to edit.
The section expands.

Previewing and Saving Changes

The Page Layout Helper **Preview** button opens a browser window to let you see how the page will look when you save your changes. Changes are not saved until you click the **Save** button.

A new window opens each time you click **Preview**. You can close these windows as you finish looking at them. When you click the **Save** button, the changes you made in any open section are saved to the XSLT stylesheet. All changes are optional.

You can also view your customizations in a browser window. For details, refer to “Viewing User Interface Changes in a Browser Window” on page 114.

Customizing Global Attributes

Global attributes are elements that appear on all user interface pages. The **Global Attributes** section of the Page Layout Helper enables you to make the following changes to user interface pages:

- “Changing the Logo” on page 98
- “Changing the Font Face” on page 98
- “Adding a Header and Footer” on page 98

Changing the Logo

By default, the Google logo appears on the search page, the search results page, the advanced search page, and the search within results page. As shown in the search result page figure (see “Getting Started with Customizing the User Interface” on page 96), you can replace the Google logo with your organization's logo. You can also remove any logo from user interface pages.

To change the logo:

1. Open the Page Layout Helper (see “Opening the Page Layout Helper” on page 97).
2. Click the right arrow next to **Global Attributes** to display the contents.
3. Enter the location and name of your company logo.

You may have to type the complete URL.

4. Enter the width and height in pixels of your logo image.
5. Click **Save**.

Changing the Font Face

The global font face is used for all text on all user interface pages. By default, the global font face is Arial, sans-serif, as shown in the search result page figure (see “Getting Started with Customizing the User Interface” on page 96). You can change the global font face using the Page Layout Helper. To change font sizes, font colors, or font faces for individual user interface elements, use the XSLT Stylesheet Editor (see “Customizing the User Interface in the XSLT Stylesheet” on page 105).

To change the global font face:

1. Open the Page Layout Helper (see “Opening the Page Layout Helper” on page 97).
2. Click the right arrow next to **Global Attributes** to display the contents.
3. Enter the name of the font family that your web site uses, for example, Times Roman serif.

The font face is case insensitive. If you enter a font that is not recognized, the page uses the Times font face.

4. Click **Save**.

Adding a Header and Footer

Your organization's web site may achieve a uniform look by using a standard header and footer on its pages. If so, you might want to maintain this look on the search appliance user interface. If pages on your organization's web site use a standard header and footer, you can add them to the search appliance user interface. The search result page figure (see “Getting Started with Customizing the User Interface” on page 96) includes a standard page header.

To add a header and/or footer:

1. Open the Page Layout Helper (see “Opening the Page Layout Helper” on page 97).
2. Click the right arrow next to **Global Attributes** to display the contents.
3. Paste your web site's header code in the **Header** area.
4. Paste your web site's footer code in the **Footer** area.
5. Click **Save**.

Customizing the Search Box

The **Search Box** section of the Page Layout Helper enables you to change the following elements:

- Search box and button (see “Changing the Search Box and Button” on page 99)
- Menu for searching by collection (see “Adding a Menu to Search by Collection” on page 100)
- Public and secure search radio buttons (see “Disabling Public and Secure Search Radio Buttons” on page 100)

The following figure shows a search box that has been customized using the Page Layout Helper.



For descriptions of the customizations illustrated in this figure, refer to the key numbers in the following table.

Key	Customization
1	Changed the length of the search box
2	Changed the search button text
3	Added a menu for searching by collection

Changing the Search Box and Button

By default, the search box length accommodates 32 characters, but it scrolls to allow longer queries. The search button label reads “Google Search.” As shown in the search box figure (see “Customizing the Search Box” on page 99), you can change the search box length and the search button text. You can also replace the search button with an image.

To change the search box and button:

1. Open the Page Layout Helper (see “Opening the Page Layout Helper” on page 97).
2. Click the right arrow next to **Search Box** to display the contents.
3. To lengthen or shorten the search box from 32 characters, type another number.
4. To replace the phrase **Google Search** on the button, type another word in the **Use text** box.
5. To use another image to replace the search button, click the **Use image** option and enter the complete URL to the image.
6. Click **Save**.

Adding a Menu to Search by Collection

A collection is a subset of the complete search index. If you include a menu to search by collection, you enable end users to narrow their searches. The search box figure (see “Customizing the Search Box” on page 99) includes a menu to search by collection. By default, the search box does not have a menu to search by collection.

To add a menu to search by collection:

1. Open the Page Layout Helper (see “Opening the Page Layout Helper” on page 97).
2. Click the right arrow next to **Search Box** to display the contents.
3. To display a menu of your collections so that your users can select which one to search, click the **Collections** checkbox.
4. Click **Save**.

To add more collections:

1. Open the Page Layout Helper (see “Opening the Page Layout Helper” on page 97).
2. Click the right arrow next to **Search Box** to display the contents.
3. Deselect the **Collections** checkbox.
4. Click **Save**.
5. Select the **Collections** checkbox.
6. Click **Save**.

If you add another collection, change the name of a collection, or remove a collection from a search box, you must regenerate your front end.

For more information about collections, refer to “Using Collections with Front Ends” on page 14.

Disabling Public and Secure Search Radio Buttons

By default, the **Secure Search** option is enabled, letting your users choose to search over public documents or both public and secure documents. By default, a search box contains radio buttons for public and secure search as shown in the search box figure (see “Customizing the Search Box” on page 99). If you do not want these radio buttons to appear in the user interface, you can disable them.

To disable public and secure search radio buttons:

1. Open the Page Layout Helper (see “Opening the Page Layout Helper” on page 97).
2. Click the right arrow next to **Search Box** to display the contents.
3. Click the checkbox to disable the display of the **Secure search** option.
4. Click **Save**.

Modifying Search Results

On the search results page, you can determine what data comes back, how it is arranged, and how it looks. As you select check boxes in each Search Results area, the sample page on the right shows your changes dynamically (for some browsers). Other browsers display a **Quick Preview** button.

Using the Page Layout Helper, you can change the following elements on the search results page:

- Page top elements (see “Changing Page Top Elements” on page 101)
- Page divider (see “Changing the Page Divider” on page 102)
- Navigation links (see “Changing Page Top Navigation Links” on page 102)
- Dynamic result clusters (see “Changing Dynamic Result Clusters” on page 102)
- Dynamic navigation (see “Changing Dynamic Navigation” on page 103)
- Sidebar elements (see “Changing Sidebar Elements” on page 103)
- Elements in result listings (see “Changing Result Listing Elements” on page 103)
- Page bottom elements (see “Changing Page Bottom Elements” on page 104)

Changing Page Top Elements

Elements that can appear at the top of a search results page include:

- Logo
- Advanced search link
- Search tips link
- Search box

By default, all these elements appear at the top of a search results page, as shown in the search result page figure (see “Getting Started with Customizing the User Interface” on page 96).

Using the Page Layout Helper, you can choose which page top elements you want to show or hide. To change text labels for the Advanced Search link or Search Tips link, use the XSLT Stylesheet Editor (see “Customizing the User Interface in the XSLT Stylesheet” on page 105).

To change page top elements:

1. Open the Page Layout Helper (see “Opening the Page Layout Helper” on page 97).
2. Click the right arrow next to **Search Results** to display the contents.
3. Click the check boxes to show or hide the **Logo, Advanced Search link, Search Tips link, or search box (top)**.
4. When finished, click **Save**.

Changing the Page Divider

A search results page can include a page divider. The divider can be a blue bar with search information or a gray line. By default, the page divider is a blue bar with search information, as shown in the search result page figure (see “Getting Started with Customizing the User Interface” on page 96).

Using the Page Layout Helper, you can choose whether to use a divider or not and which divider you want to use. To change text labels for Search Information, use the XSLT Stylesheet Editor (see “Customizing the User Interface in the XSLT Stylesheet” on page 105).

To change the search information and page divider:

1. Open the Page Layout Helper (see “Opening the Page Layout Helper” on page 97).
2. Click the right arrow next to **Search Results** to display the contents.
3. Click the **Search information** check box to show or hide search information.
4. Click a radio button to choose **Blue bar**, **Gray line**, or **No divider**.
5. When finished, click **Save**.

Changing Page Top Navigation Links

A search results page can include the following navigation links at the top of results listings:

- **Previous/Next** link
- **Sort by Date/Sort by Relevance** link

By default, the search results page includes both of these navigation links, as shown in the search result page figure (see “Getting Started with Customizing the User Interface” on page 96).

Using the Page Layout Helper, you can choose whether to show or hide one or both of these navigation links. To change text labels for navigation links, use the XSLT Stylesheet Editor (see “Customizing the User Interface in the XSLT Stylesheet” on page 105).

To change navigation links:

1. Open the Page Layout Helper (see “Opening the Page Layout Helper” on page 97).
2. Click the right arrow next to **Search Results** to display the contents.
3. Click the check boxes to show or hide the **Previous/Next** link or the **Sort by Date/Sort by Relevance** link.
4. When finished, click **Save**.

Changing Dynamic Result Clusters

Dynamic result clusters are keywords that are based on the results of each search query. Each keyword groups similar documents together. For information about dynamic result clusters, refer to “Narrowing Searches” on page 18.

By default, the search results page does not display dynamic result clusters. Using the Page Layout Helper, you can choose whether to hide or show dynamic results clusters, and where they should appear on the search results page, beside or at the top of results listings.

To change dynamic result clusters attributes:

1. Open the Page Layout Helper (see “Opening the Page Layout Helper” on page 97).

2. Click the right arrow next to **Search Results** to display the contents.
3. Click the check box to show or hide **Dynamic result clusters**.
4. Click a radio button to display dynamic result clusters at the **Side** or **Top** of search results.
5. When finished, click **Save**.

Changing Dynamic Navigation

Dynamic navigation helps users explore search results by using specific metadata attributes. For information about dynamic navigation, refer to “Using Dynamic Navigation to Help Users Explore Results” on page 49.

By default, the search results page does not display dynamic navigation. Using the Page Layout Helper, you can choose whether to hide or show dynamic navigation.

To change dynamic navigation:

1. Open the Page Layout Helper (see “Opening the Page Layout Helper” on page 97).
2. Click the right arrow next to **Search Results** to display the contents.
3. Click the check box to **Show Dynamic Navigation**.
4. When finished, click **Save**.

Changing Sidebar Elements

Sidebar elements include search results from **Google Apps** or **Google Site Search**. By default, the search results page does not display sidebar elements. Using the Page Layout Helper, you can choose whether to hide or show sidebar elements. You cannot show sidebar elements along with dynamic navigation or dynamic result clusters side view.

To show results from Google Apps, you must enable indexing public content or integrating personal content by using the **Content Sources > Google Apps** page. To show results from Google Site Search, you must provide a search engine identifier.

To change sidebar elements:

1. Open the Page Layout Helper (see “Opening the Page Layout Helper” on page 97).
2. Click the right arrow next to **Search Results** to display the contents.
3. Click the check boxes to show the selected elements.
4. When finished, click **Save**.

Changing Result Listing Elements

Each result listing on a search results page can include the following elements:

- Result title length
- Snippet
- URL
- Page size

- Modified date
- Cache link

By default, the search results page includes all of these elements. The search result page figure (see “Getting Started with Customizing the User Interface” on page 96) includes most of these elements, but omits the URL.

Using the Page Layout Helper, you can choose whether to show or hide any of these result listing elements. To change other aspects of these elements, such as font color, use the XSLT Stylesheet Editor (see “Customizing the User Interface in the XSLT Stylesheet” on page 105).

To change the Search Results attributes:

1. Open the Page Layout Helper (see “Opening the Page Layout Helper” on page 97).
2. Click the right arrow next to **Search Results** to display the contents.
3. Click check boxes to show or hide the result **Snippet, URL, Page size, Modified date, or Cache link**.
4. When finished, click **Save**.

Changing Page Bottom Elements

Elements that appear at the bottom of a search results page include:

- **Previous/Next** page navigation links
- Search box

By default, the search results page includes Google-style navigation links and a search box.

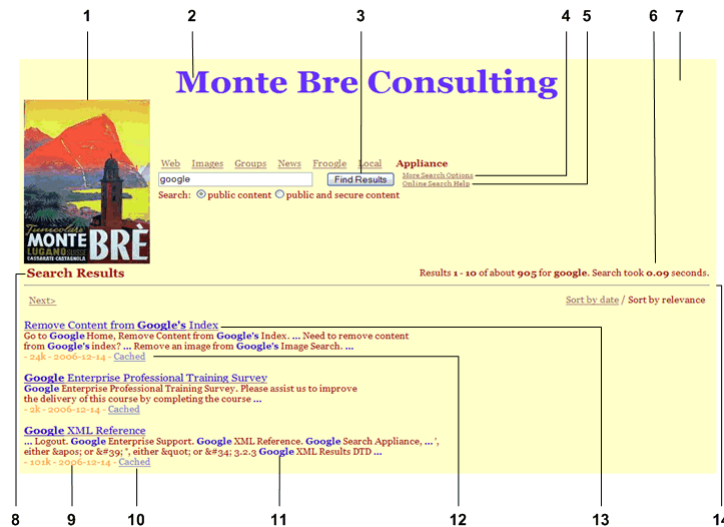
Using the Page Layout Helper, you can choose a different style of navigation link or hide the search box. To change other aspects of these elements, use the XSLT Stylesheet Editor (see “Customizing the User Interface in the XSLT Stylesheet” on page 105).

To change the page bottom elements:

1. Open the Page Layout Helper (see “Opening the Page Layout Helper” on page 97).
2. Click the right arrow next to **Search Results** to display the contents.
3. Click a radio button to choose a style of Previous/Next page navigation links.
4. Click the check box to show or hide the **Search box (bottom)**.
5. When finished, click **Save**.

Customizing the User Interface in the XSLT Stylesheet

You might want to customize a user interface in ways that are not supported in the Page Layout Helper. Such changes might include adding a background color and changing fonts for specific elements in the results listings. The following figure illustrates a search results page with extensive changes.



All the customizations in this figure were made using the XSLT Stylesheet Editor. For descriptions of the customizations in this figure, refer to the key numbers in the following table.

Key	Customization
1	Replaced the Google logo with the organization's logo.
2	Added the header used on the organization's web site.
3	Changed search button text from Google Search to Find Results .
4	Changed the advanced search anchor text from Advanced Search to More Search Options .
5	Changed search help anchor text from Search Tips to Online Search Help .
6	Changed global text color and font face from Arial, sans-serif to Georgia, serif.
7	Added a background color.
8	Added a separate font color for separation bar standard text.
9	Added a separate font color for result dates.
10	Added a separate font color for result cached links.
11	Added a separate font color for result keyword matches.
12	Removed URLs from result snippets.
13	Added a separate font color for result links.
14	Changed separation bar from blue bar to gray line.

Working with the XSLT Stylesheet Editor

Working directly with the XSLT stylesheet enables you to make extensive changes to the user interface. All of the elements that you can modify with the Page Layout Helper can be modified by editing the stylesheet directly. However, once you modify the stylesheet directly, you cannot make subsequent changes to the same stylesheet using the Page Layout Helper.

To customize the search and search results pages by editing the XSLT stylesheet, use the XSLT Stylesheet Editor or another editor of your choice. The following sections describe use of the XSLT Stylesheet Editor:

- “Opening the XSLT Stylesheet Editor” on page 106
- “Previewing XSLT Stylesheet Changes” on page 106
- “Exporting an XSLT Stylesheet” on page 107
- “Importing an XSLT Stylesheet” on page 107
- “Restoring the XSLT Stylesheet” on page 108

For detailed information about the contents of the XSLT stylesheet, refer to “Changing Variables in the XSLT Stylesheet” on page 108.

Opening the XSLT Stylesheet Editor

The XSLT Stylesheet Editor appears on the **Search > Search Features > Front Ends > Output Format** page in the Admin Console. Before you can use the XSLT Stylesheet Editor, you must create a front end. If you started customizing a user interface using the Page Layout Helper, you already have a front end. For information, refer to “Creating a Front End” on page 94.

To open the XSLT Stylesheet Editor:

1. Choose **Search > Search Features > Front Ends**.
The **Front Ends** page appears.
2. Select a front end from the **Current Front Ends** list and click **Edit**.
The **Output Format** page appears.
3. In the **XSLT Stylesheet Editor** box, click the **Edit Underlying XSLT Code** link.
The XSLT Stylesheet Editor expands.

Previewing XSLT Stylesheet Changes

When you are editing variables in the XSLT stylesheet, you should periodically preview the results of your changes and avoid saving changes until you are satisfied. When you click **Save** the corresponding front end is updated after 15-30 minutes. The preview pages display how the search or search results pages look with the changes you have made. However, the links and buttons on the preview page are not functional.

To see the actual search or search results pages, use the **Test Center** link in the blue bar at the top right of the page. To return the XSLT stylesheet to the state before it was edited, click **Restore Default**.

To preview changes that you make to the XSLT stylesheet:

1. Open the XSLT Stylesheet Editor (see “Opening the XSLT Stylesheet Editor” on page 106).
2. Enter any changes to the stylesheet and click **Preview** to review the changes.
3. When finished previewing your changes, close the preview window.
4. Correct errors or make more changes.
5. When finished, click **Save**.

You can also view your customizations in a browser window. For details, refer to “Viewing User Interface Changes in a Browser Window” on page 114.

Exporting an XSLT Stylesheet

If you work in the XSLT stylesheet, back up your changes periodically by exporting the stylesheet. You also need to export the XSLT stylesheet if you plan on editing it in an editor other than the XSLT Stylesheet Editor. You can export the XSLT stylesheet to work on in another location, then import it when you are satisfied with the changes. You can then preview the changes you've made.

To export the XSLT stylesheet:

1. Open the XSLT Stylesheet Editor (see “Opening the XSLT Stylesheet Editor” on page 106).
2. Click **Export**.
3. In the File Download wizard, click **OK** and navigate to a location for the file.

The default name for the stylesheet is *frontendname_frontend_stylesheet.en.xslt*. You can give the exported stylesheet any name you choose.

Importing an XSLT Stylesheet

If you develop an XSLT stylesheet outside the search appliance, or have a back-up version of the XSLT stylesheet, you can import it to the Google Search Appliance.

To import an XSLT stylesheet:

1. Open the XSLT Stylesheet Editor (see “Opening the XSLT Stylesheet Editor” on page 106).
2. Enter the filename of the edited XSLT stylesheet in the **Import stylesheet box**, or browse for the file.
3. Click the **Import** button.

A confirmation message warns that this will overwrite your page layout settings.

4. Click **OK**.

The edited XSLT stylesheet displays and is validated. Errors found during validation are displayed in red.

5. Fix errors in the file and repeat these Import steps.
6. When finished, click **Save**.

Restoring the XSLT Stylesheet

You can discard any changes to an XSLT stylesheet and restore the XSLT stylesheet to the state before it was edited.

To restore the XSLT stylesheet:

1. Open the XSLT Stylesheet Editor (see “Opening the XSLT Stylesheet Editor” on page 106).
2. Click **Restore Default**.
A confirmation message warns that this will overwrite your current stylesheet.
3. Click OK. The XSLT stylesheet before it was edited is restored.

Changing Variables in the XSLT Stylesheet

For ease of use, the XSLT stylesheet is divided into the following sections:

- Logo setup (see “Changing the Logo” on page 108)
- Global style variables (see “Changing Global Style Variables” on page 109)
- Result page components (see “Changing Result Page Components” on page 109)
- Results elements (see “Changing Result Elements” on page 110)
- Other variables (see “Changing Other Variables” on page 111)
- Other stylesheet sections (see “Changing Variables in Other XSLT Stylesheet Sections” on page 112)

Comments precede each section, so that you know whether a section can be customized. The following example presents the Global style variables section of the stylesheet.

```
<!-- *****  
Global Style variables (can be  
customized): ' ' for using browser's default  
***** -->  
<xsl:variable name="global_font">arial,sans-serif</xsl:variable>  
<xsl:variable name="global_font_size"></xsl:variable>  
<xsl:variable name="global_bg_color">#ffffcc</xsl:variable>  
<xsl:variable name="global_text_color">#990000</xsl:variable>  
<xsl:variable name="global_link_color">#996666</xsl:variable>  
<xsl:variable name="global_vlink_color">#551a8b</xsl:variable>  
<xsl:variable name="global_alink_color">#ff0000</xsl:variable>
```

Changing the Logo

The Logo setup section of the XSLT stylesheet contains variables that control the logo that appears on user interface pages. The search result page figure (see “Getting Started with Customizing the User Interface” on page 96) includes a custom logo.

Changing Global Style Variables

The Global Style variables section of the XSLT stylesheet contains variables that control the following elements of all search and search results pages:

- Font faces and sizes
- Page background color
- Text color
- Link colors

The search result page figure (see “Getting Started with Customizing the User Interface” on page 96) illustrates changes to the global font face, global font colors, global link colors, and page background color.

Changing Result Page Components

The Result page components section of the XSLT stylesheet contains variables that control the appearance of elements that appear at the top of the search results page. These elements include the page header, search box and button, separation bar, spelling suggestions, and so on. The search result page figure (see “Getting Started with Customizing the User Interface” on page 96) illustrates changes to the page header, search button, and separation bar.

The following table lists all result page elements that you can change and the types of changes you can make to each one.

User Interface Element	Customization
Result page header	Specify whether to use your own page header, the page header provided by Google, or both, with the provided header appearing under your page header.
Provided result page header	Specify links and labels that appear in the search results page header provided by Google.
Search boxes	Display a search box at the top or the bottom of the search results page. Specify the size of the search box.
Search buttons	Specify search button type (text or image).
Search info bars	Display search information bars.
Separation bar	Specify type of separation bar and label.
Navigation bars	Specify top and bottom navigation bars and attributes.
Sort by	Display Sort by link.
Spelling suggestions	Display spelling suggestions, change label text (Did you mean:), or color.
Synonyms suggestions	Display synonym suggestions, change label text (You could also try:), or color.
KeyMatches	Display KeyMatches, change label text (KeyMatch), text color, or background color.
Advanced Search Reporting	Specify whether to enable advanced search reporting.

Changing Result Elements

The Result elements section of the XSLT stylesheet contains variables that control the appearance of all search result listings. By editing variable values in this section, you can change the font face and color of result titles, snippets, and link URLs. The search result page figure (see “Getting Started with Customizing the User Interface” on page 96) illustrates changes to the result title color, keyword match color, and result cache link color.

The following table lists all elements of results listings that you can change and the types of changes you can make to each one.

User Interface Element	Customization
Result title	Display result title, change color of result title, change font size of result title, show or hide result snippet, change color of result snippet.
Keyword match	Change color of keyword match (the word or phrase that matches the search term) in result titles and snippets, change font size of keyword matches, use boldface font in keyword matches.
Link URL	Display result link URL, change color of result link URL, change font size of result link URL, truncate result link URL, or the length of truncated result link URL.
Meta tags	Display meta tags.
Results size	Display the size of result pages in kilobytes.
Results date	Display dates when pages were found.
Results cache	Display links to results cache versions of pages.
Color used in result cache link, similar pages, and description	Change the faint color used in result cache links, similar pages, and descriptions.
Secure results radio button	Display secure results radio button.

Changing Other Variables

The Other variables section of the XSLT stylesheet contains variables that control page titles, aspects of the Advanced Search Page, the Cached page header, error message text, and dynamic result clusters. The following table lists all elements of results listings that you can change and the types of changes you can make to each one.

User Interface Element	Customization
Page title	Change titles for the following pages: Search Home, Search Results, Advanced Search, and Search Within Results.
Advanced search page header	Specify whether to use your own page header, the provided page header, or both. The Advanced Search page contains expanded options for search queries, including Find Results keyword options, Language options, File Format options, and so on.
Advanced search page pane background color	Change the advanced search page pane background color.
Cached page header	Change the header text on the cached page. A cached page appears when a user clicks a cached link in search results. The header text explains that the document is a cached copy and not the current page.
Error message text	Change the error message text for a server error or an unknown XML result type.
Dynamic result clusters	Specify whether to include dynamic result clusters or not and where they should appear on the search results page.
Alerts	Specify whether to show the My Alerts link on the search and results pages.

Changing Variables in Other XSLT Stylesheet Sections

The remaining customizable sections of the XSLT stylesheet contain variables that enable you to change the elements listed in the following table.

User Interface Element	Customization	Stylesheet Section
My global page header/footer	Enter XML-compatible HTML code in this section to affect the header and footer of every search and search results page.	My global page header/footer
Logo template	Change the text that appears when a mouse hovers over the logo at the top of the search page, the search results page, and the advanced search page. The logo is also a link to the search page.	Logo template
Search results page header	Enter XML-compatible HTML code in this section to affect the search results page header. You can also change the font size of the header. The Search Within Results page enables the user to start a new search within the results of the last search.	Search result page header
Search within results page header	Specify whether to use your own page header, the page header provided by Google, or both, with the provided header appearing under your page header.	Search within results page header
Home search page header	Enter XML-compatible HTML code in this section to affect the home search results page header.	Home search page header
Separation bars	Change the color and background color of separation bars used in advanced search headers and search results pages. Separation bars divide the page header from the results.	Separation bar variables
Advanced search page header	Put whatever you like in the page's header. It appears under the Global headers on your pages.	Advanced search page header HTML
Cached page header	Enter XML-compatible HTML code in this section to affect the cached page header.	Cached page header
Search within results search input page	Enter XML-compatible HTML code in this section to change the look of the search page completely.	Search within Results search input page
Front door search input page	Enter XML-compatible HTML code in this section to change the look of the search page completely.	Front Door search input page
Empty result set	Change the look of the search results page that users see when there are no results to return.	Empty result set

Changing the Language of the User Interface

You can have your users' search page and search results pages in a language other than English, which is the default. You also can have several languages active for your users and the search appliance will present search results for an active language based on the settings in the user's computer.

The search appliance allows multiple stylesheets that present the search page, advanced search, and results pages in different languages, all associated with a single front end. The language-specific stylesheet is selected based on the Accept-language header sent from the user's browser. The stylesheet is selected from the set of languages marked "active"; if there is no match, the default language is used. A language-specific stylesheet is created when you make a language active. Each language's stylesheet can be edited and customized independently.

To change the language of the user interface, use the **Search > Search Features > Front Ends > Output Format** page in the Admin Console. For complete information about using the **Output Format** page, click **Admin Console Help > Search > Search Features > Front Ends > Output Format** in the Admin Console.

The search appliance displays search results in the following languages:

- Basque
- Catalan
- Chinese (Simplified)
- Chinese (Traditional)
- Czech
- Danish
- Dutch
- English (Default)
- English (UK)
- Finnish
- French
- Galician
- German
- Greek
- Hungarian
- Italian
- Japanese
- Korean
- Norwegian
- Polish
- Portuguese (Brazil)
- Portuguese (Portugal)
- Russian
- Spanish
- Swedish
- Turkish
- Vietnamese

Customization Process Overview

You may customize a user interface using both the Page Layout Helper and by editing the XSLT stylesheet directly. You must make all Page Layout changes in the boxes provided *before* editing the XSLT stylesheet directly. These changes are saved in the XSLT stylesheet when you click **Save**.

You cannot return to using the Page Layout Helper after you manually edit the XSLT stylesheet, unless you start over completely by clicking the **Restore Default** button. If you want to continue editing, you can do so in the XSLT Stylesheet Editor.

Here is the recommended sequence for customizing a user interface:

1. Create a new front end.
2. Use the Page Layout Helper to make changes to the user interface.
3. Click **Preview** to view each change you make.

A new browser window opens with each preview, so close the window each time you return to the Page Layout Helper. The **Preview** button lets you look at each change you make before you move on. It does not save your changes.

4. Continue making changes in the Page Layout Helper and previewing them.
5. When finished, click **Save**.
6. Click **Export** to save the XSLT stylesheet as a backup.

If you are satisfied with the page layout of your search and search results pages, go to step 13 now. If you want to make more changes, go to step 7. If you edit the XSLT stylesheet, those edits are made in addition to the Page Layout changes. You cannot return to the Page Layout Helper after editing the XSLT stylesheet itself; the Page Layout Helper is disabled.

7. Click the **Edit underlying XSLT code** link.

(The code now contains your page layout changes from using the Page Layout Helper.)

8. Following the commented instructions, make the changes you want.
9. Click **Preview** to see your changes.

A new browser window opens with each preview, so you may want to close the window each time you return to the XSLT stylesheet.

10. Continue making changes in the XSLT code and previewing the changes.
11. When finished and ready to serve the changed pages, click **Save**.
12. Click **Export** to save the XSLT stylesheet as a backup.
13. Test your changed search and search results pages by using the **Test Center** link at the upper right of the page.

Click the links and do some searches to make sure the pages look the way you want them to. Changed pages are served to users within 15-30 minutes.

Note: Later, you can use the **Import** button to use your edited XSLT stylesheet to make further changes.

Viewing User Interface Changes in a Browser Window

Creating a user interface is an iterative process. During this process, you may want to view your saved changes in a browser window. By default, the stylesheet cache is updated approximately every 15-30 minutes. To refresh the stylesheet cache and display your current changes immediately, include the search parameter `&proxyreload=1` in the search request URL. The following search request URL example includes the `proxyreload` parameter:

```
http://search_appliance_name/search? site=default_collection&client=
default_frontend&output=xml_no_dtd&proxystylesheet=
my_frontend&proxyreload=1&proxycustom=%3CHOME/%3E
```

For information about search parameters, refer to the *Search Protocol Reference*.

Managing Customized XSLT Stylesheets

If you customize XSLT stylesheets, make sure that you keep a backup copies offline. To back up an XSLT, export it as described in “Exporting an XSLT Stylesheet” on page 107. You can also export the entire configuration for a search appliance using the **Administration > Import/Export** page in the Admin Console.

To help manage ongoing customizations to XSLT stylesheets, you can use a source control system. This method of managing XSLT stylesheets is especially useful if you deploy an XSLT stylesheet in more than one search appliance, such as a master and a hot spare. The source control system can contain the authoritative version of the XSLT stylesheet.

Migrating a Customized XSLT Stylesheet to a New Software Release

In some instances, a new software release of the search appliance contains changes to the XSLT stylesheet. If you have a customized XSLT stylesheet from an earlier software release, you can migrate it to the new software release by adding any new XSLT code to it. Before you begin editing the XSLT stylesheet, ensure that you have an exported version of it. If not, export it, as described in “Exporting an XSLT Stylesheet” on page 107.

To migrate a customized stylesheet to a new software release:

1. Identify changes in the new XSLT stylesheet that you want to copy into the customized stylesheet.
2. Open both the exported, customized stylesheet and the new stylesheet in an XSLT compatible editor.
3. Scroll to the section of the XSLT stylesheets where the code that you want to copy appears.

For example, if the new code appears in the Result Page Components section, scroll to that section in both XSLT stylesheets.

4. Copy the lines of new or changed code from the XSLT stylesheet for the new release and paste them into the customized XSLT stylesheet.
5. Save your changes.
6. Create a new front end for the updated XSLT stylesheet as described in “Creating a Front End” on page 94.
7. Import the customized XSLT stylesheet into the new front end, as described in “Importing an XSLT Stylesheet” on page 107.

For more information about changes or new features in an XSLT stylesheet for a release, see the update instructions for the software release.

User Interface Design Principles

This section contains the following best practices tips and guidelines for designing an effective user interface:

- “Keep Search Pages Clean, Simple, and Fast” on page 116
- “Keep Advanced Search Separate” on page 116
- “Make Search Ubiquitous” on page 116
- “Make Sure the Search Box Is Big Enough” on page 116
- “Make Sure the User Knows what Documents Have Been Searched” on page 117
- “Make Help Easily Available” on page 117
- “Make Search Results Pages Useful” on page 117
- “Keep the Number of Results Reasonable” on page 117

Keep Search Pages Clean, Simple, and Fast

Searching should be fast, and faster search encourages users to search more often. Keep the main search page simple. Put advanced search features on a separate page. On search results pages, try to keep navigational elements that aren't search-related to a minimum.

Keep Advanced Search Separate

Over 95% of users do not use advanced search features. Make the primary search mechanism simple. At most the simple search area should invite keywords to be entered, and possibly the choice of key categories (such as collections) for searches. On the simple search page, place a link to an advanced search page for those users who want advanced features.

Make Search Ubiquitous

It should be easy for users to search for information from any page on your organization's sites. Every page should include a search box or at least a link to a search page. Every search results page should also include a search box to facilitate subsequent searches.

Make Sure the Search Box Is Big Enough

Search boxes should be big enough to promote multiple word entries. Users typically only enter what fits in the search box. If the search box is small, a user may enter only a single word. Because Google provides excellent support for multi-word, phrase, or natural-language queries, make the search box big enough for users to enter larger queries. The recommended minimum size for a search box is 20-30 characters.

Make Sure the User Knows what Documents Have Been Searched

Near the search box and on the search results page, be sure to state what document set is being searched over. It allows users to focus their searches and is especially important when using multiple collections. There should also be a link that enables a user to expand a search from a collection to the entire search index at any time from the search results page.

Make Help Easily Available

A link to a help page should be accessible near the search box.

Make Search Results Pages Useful

Search results pages should have all the information that a user needs to determine whether a given search result meets her information needs. Within a search results page, the following information about the results should appear:

- The original query within a search box
- Number of results for the query
- Total number of results
- Current retrieval set (for example, 21-30 of N, where N = total number of results)
- Time it took for the results to be returned
- State the query searched for and what collection was searched
- Title of a result document
- Dynamically generated document snippet
- URL of the result document
- Size of the result document

Also, the search appliance provides additional cached page features that may be provided with the results. Cache results allow the user to see the HTML version of the original page. This feature is especially useful if the URL is currently unavailable or in a format that the user's computer cannot read.

Keep the Number of Results Reasonable

Calculating and displaying lots of results is computing-intensive. Because most users find what they are looking for in the first 10 results, it is recommended that each search results page show only 10 results. You can always give the users an advanced search option to change the number of search results that appear on each search results page.

Chapter 4

Advanced Customization Topics

About This Document

This section describes the audience for this document, the organization of the material, and some additional sources of information.

Audience

This document is for search administrators and webmasters who want to customize the search interface. To perform most customization tasks, you need privileges to edit search appliance front ends and eXtensible Stylesheet Language Transformation (XSLT) stylesheets. For some customizations, you must also have privileges to edit your web portal and content pages.

This document provides detailed procedures to direct general users in the steps for editing HTML markup or XSLT code. Some background in HTML, XSLT, and search appliance query parameters is helpful for understanding the customization examples and replicating them in your environment.

The Customizations and Example Pages

This document uses examples to illustrate customizations. Each customization is described in a separate section with detailed explanation of the required markup or code. Each example is also illustrated in the context of example pages based on a fictional fishing fly-tying business.

The following customizations are described and illustrated in this document:

Customization	Description	Example Page
“Presenting the Search Interface in an Inline Frame” on page 121	Present the search and results pages in an inline frame within a static page.	“Inline Frame Example page” on page 120
“Creating an HTML Form Search Box” on page 122	Integrate an HTML-based search box directly into your web page instead of using the default search page.	“HTML Form Example Page” on page 119
“Combining an HTML Form Search Box with Inline Results” on page 124	Combine inline frame presentation of the results page with a custom search box.	“HTML Form Example Page” on page 119
“Replacing the Secure Search Radio Button with a Check Box” on page 124	Save screen space by using a checkbox for the secure search option in place of default radio buttons.	“Inline Frame Example page” on page 120
“Specifying Query Parameters with XSLT” on page 127	Make the search page submit selected query parameters with each search request.	“Inline Frame Example page” on page 120
“Including an Image Link for Results” on page 130	For each result with a product illustration or other image, present a thumbnail image in the result item on the results page.	“HTML Form Example Page” on page 119
“Replacing the Result Snippet with Custom Text” on page 131	Make each search result show a product description or other specific custom text in place of the standard description snippet.	“HTML Form Example Page” on page 119

HTML Form Example Page

The HTML form example page shown below illustrates a custom HTML form box, results image links, and custom description text.

The screenshot shows a search results page for 'ACME Flies'. The search bar contains the text 'nymph' and has a 'Search' button. The results list includes items like 'Prince Nymph', 'Dark Lord', 'AP Nymph', 'Copper John', and 'Fox Poopah', each with a small image and a custom description. Annotations on the left indicate: 'Search request is submitted from HTML form box.' pointing to the search bar; 'Each result includes an image link.' pointing to the thumbnails; and 'Some results use custom description text.' pointing to the text snippets under the thumbnails.

To view the source code for this page and its supporting XSLT stylesheet, see “Optional Example Materials” on page 138.

Inline Frame Example page

The inline frame example page shown below illustrates Google search presented in an inline frame with a custom secure search checkbox. The code for the search button in this example page has been customized to submit selected search parameters that limit the number of results displayed in the limited screen space of the inline frame.

The screenshot shows a web page titled "ACME Flies" with a subtitle "The ACME in quality nymphs, dries and streamers". The page content includes a welcome message, a search prompt, and a list of categories: Nymphs, Dry Flies, and Streamers. A search bar is present with a "Google Search" button and a "Secure Search" checkbox. The search results for "nymph" are displayed, showing two results: "Prince Nymph" and "AP Nymph". Annotations with arrows point to various elements: "The Google search interface is displayed in an inline frame within the page" points to the search bar; "This button submits selected parameters with each search request" points to the search button; "A secure search checkbox replaces the default radio buttons" points to the "Include secure content" checkbox; and "Meta tags in each result page are retrieved and displayed" points to the meta tags in the search results.

To view the source code for this page and its supporting XSLT stylesheet, see "Optional Example Materials" on page 138.

For More Information

Many basic customizations, such as displaying a custom logo or changing colors and fonts, are not included in the scope of this document. For information on performing basic customization work and an overview of the search process, see "Customizing the User Interface" on page 92.

Some of the advanced customizations require you to set or modify query parameters. For more information on query parameters, see the *Search Protocol Reference* and "Using Front End Search Parameters" on page 94.

Support Limitations

Keep in mind that the advanced customizations described in this document are not supported. Google Support cannot support or debug your customized HTML or XSLT code.

Customized XSLT stylesheets are not automatically upgraded when you update the search appliance software version. To enable new features in future versions, you may need to modify your customized stylesheets, or add your customized code to the default stylesheet for newer versions.

Integrating Search into Web Pages

If your goal is to integrate search into existing web pages that have a particular style and identity, you can modify or minimize the visual effect of the default search and results pages. This section describes three levels of integration:

- A basic approach that presents search pages in an inline frame.
- An HTML form search box that replaces the default search page.
- A combination of an HTML form search box and results pages in an inline frame within the page.

Presenting the Search Interface in an Inline Frame

Using inline frames, you can present the search interface within the context of an otherwise static web page as illustrated in the “Inline Frame Example page” on page 120.

HTML Code for Integrating Search with an Inline Frame

The integrated search interface in the example page is derived from an inline frame tag with a `src` attribute pointing to *iframe-example-frontend*. As shown below, the inline frame is presented within a table cell:

```
<table border="0" cellpadding="0" cellspacing="0" width="800">
<tr><td>
<p>Search our extensive database of flies:</p>
<tr><td>
<iframe id="iFrame" name="FlyFrame" frameborder="0" width="800" height="700"
scrolling="auto"
OnLoad="window.scroll(0, 0);" src="http://appliance_hostname/
search?site=default_collection
&client=iframe-example-frontend&output=xml_no_dtd&proxystylesheet=iframe-
example-frontend"></iframe>
</td></tr>
</table>
```

At a minimum, the source attribute for an inline frame tag must specify the protocol and search appliance hostname, `http://appliance_hostname`. If you provide only the hostname, the default front end search interface appears in the inline frame.

In the inline frame example, the source URL contains custom query parameters to load the front end named *iframe-example frontend*, which is customized for optimal display of the search interface in an inline frame. The front end includes XSLT code that makes the search button submit selected parameters that affect the display of the search results page. For more information, see “Specifying Query Parameters with XSLT” on page 127 and the *Search Protocol Reference*.

Adding Inline Frame Search to a Page

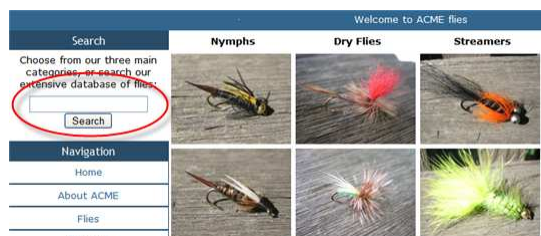
To add the search interface inside an inline frame in a web page:

1. Open the page to be modified in your preferred editor.
2. In the section of the page where you would like to display the search interface, add an inline frame tag.

- For the `src` attribute of the inline frame tag, specify `http://appliance_hostname`, where the host name is the correct value for your search appliance. This value points to the default front end. If you want to open a different front end, or customize any other query parameters for the search interface, add them to the URL as shown in the example code.
- Optionally, set the frame `width` and `height` attributes to make sure the inline frame is large enough to accommodate the search results page. Add a `noresize` attribute if you want to prevent end users from changing the size. You can set any optional attributes of the inline frame (scrolling, align, and so on.) depending on your design goals for the page.
- Save the page.
- Perform a test search from the page verify that the search page and results page are displayed inside the inline frame as expected.

Creating an HTML Form Search Box

With a simple HTML form such as the one depicted in the “HTML Form Example Page” on page 119, you can integrate a simple search box directly into your web page instead of using the default search page. This approach allows an end user to focus on the page content with minimal distraction from the search interface. However, if the user needs to perform a search, the search box is readily available, as shown below.



HTML Code for a Form Search Box

The HTML code for the search box used in the example is derived from a `form` tag in the side navigation of the page.

```
<form method="GET" target="FlyFrame" action="http://appliance_hostname/search">
  <input type="text" name="q" size="32" maxlength="256" value="" />
  <input type="submit" name="btnG" value="Search" />
  <input type="hidden" name="site" value="default_collection" />
  <input type="hidden" name="client" value="example-frontend" />
  <input type="hidden" name="output" value="xml_no_dtd" />
  <input type="hidden" name="proxystylesheet" value="example-frontend" />
  <input type="hidden" name="filter" value="p" />
  <input type="hidden" name="getfields" value="*" />
</form>
```

This form submits a `GET` request to the search engine, requesting results in the format specified by `example-frontend`. The example form includes optional input fields for a submit button and for query parameters that enhance the visual effect of the page. These parameters turn off duplicate directory filtering (`filter`), retrieve meta tag values (`getfields`), and specify the stylesheet associated with `example-frontend` (`proxystylesheet`).

The requirements for an HTML for search box are:

- Specify the `GET` method and the path `http://appliance_hostname/search`
- Capture the query terms in a text input field named `q`
- Specify the required query parameters `site`, `client`, and `output` as hidden input fields
- Unless you intend to retrieve raw XML results, specify `proxystylesheet`

In addition to the required fields, you can include any valid query parameter as a hidden input field in the HTML form. See the *Search Protocol Reference* for more information on query parameters.

Search boxes should be big enough to accommodate multiple word entries. Users typically enter queries that fit in the search box, so if the search box is small, they may enter only single words. Since Google supports multi-word, phrase, or natural language queries, make your search box big enough for your users to enter larger queries. The recommended minimum size for a search box is 20-30 characters.

Adding an HTML Form Search Box to a Page

To add an HTML form search box to a web page:

1. Open the page in your preferred editor.
2. In the section of the page where you want to display the search box, add a form tag like the following, where `appliance_hostname` is the correct value for your search appliance:

```
<form method="GET" action="http://appliance_hostname/search">
```

3. Inside the form tag, add a text input field named `q`, and provide values for size and maximum length (the absolute search appliance limit is 2048 characters):

```
<input type="text" name="q" size="32" maxlength="256" value="">
```

4. Inside the form tag, add the required query parameters `site`, `client`, and `output` as hidden input fields:

```
<input type="hidden" name="site" value="default_collection">
<input type="hidden" name="client" value="default_frontend">
<input type="hidden" name="output" value="xml_no_dtd" />
```

If you want to direct the search request to a non-default collection or front end, enter appropriate values for `site` and `client`.

5. Unless you intend to retrieve raw XML results, specify a valid value for `proxystylesheet`. Typically, this value is the name of the front end you specified for `client`.
6. Optionally, add a submit button field inside the form tag. If you do not add a button, end users must use the Enter key to submit searches.

```
<input type="submit" name="btnG" value="Search">
```

7. Save the page.
8. Perform a test search from the page to verify that the results page displayed as expected.

Combining an HTML Form Search Box with Inline Results

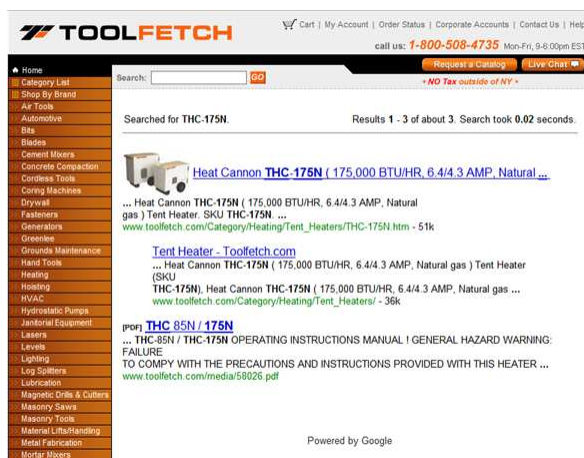
You can combine inline frame presentation of the results page with a custom search box. This combination of factors allows users to first view the frame populated with static content, and then replace that content with a list of search results if needed.

This approach is used in the “HTML Form Example Page” on page 119. The HTML form in this example page collects the search terms in a custom search box, submits them to the search engine, and then directs the results page to open in the inline frame “FlyFrame” as specified by the target attribute in the form tag:

```
<form method="GET" target="FlyFrame" action="http://appliance_hostname/search">
```

When using this approach, you may prefer to remove redundant search boxes from the results page by setting the XSLT stylesheet variables `show_top_search_box` and `show_bottom_search_box` to zero.

A production web site using a form box and inline frame at the time of writing is Toolfetch.com:



Modifying the Search Query

You may prefer to use the default Google search and results pages, but with a degree of customization. This section describes two modifications to the default search query experience: using a checkbox for secure search, and adding query parameters to a front end. Both of these customizations make only slight changes in the visual effect and behavior of the default search box.

Replacing the Secure Search Radio Button with a Check Box

Using a check box for secure search saves space on the search query page and gives it a cleaner look. The following screen capture from `iframe_index.html` depicts a customized search box in place of the default radio buttons:



XSLT Code for the Secure Search Checkbox

This customized search box is defined by XSLT rules in the stylesheet associated with *iframe-example-frontend*. In this stylesheet, the default code for the search box input form is overwritten with the following block of custom code.

```
<xsl:choose>
  <xsl:when test="$access='p'">
    <input type="checkbox" name="access" value="a" />
  </xsl:when>
  <xsl:otherwise>
    <input type="checkbox" name="access" value="a" checked="checked" />
  </xsl:otherwise>
</xsl:choose>
  Include secure content
```

This conditional clause with `<choose>` tests the value of the `access` parameter, and then defines the checkbox accordingly. When `access` is the default value of "p," the unchecked box sets the value to "a" if checked by the end user. If the value is already set to "a," the checkbox is displayed checked.

For more information on the `access` query parameter, see the *Search Protocol Reference*.

Adding Secure Search Checkbox Code to an XSLT Stylesheet

To modify a stylesheet to replace the secure search radio buttons with a checkbox:

1. Open the stylesheet in the Admin Console's XSLT Stylesheet Editor or your preferred editor.
2. Scroll to the following section heading:

```
<!-- *****
Search box input form (Types: std_top, std_bottom, home, swr)
***** -->
```

3. Scroll down within the search box input form section to locate the XSLT conditional clause beginning with the following line:

```
<xsl:if test="$show_secure_radio != '0'">
```

This conditional clause contains the default radio button code, shown below in boldface and bracketed by start/end comments (these comments are not found in your default stylesheet):

```
<xsl:if test="$show_secure_radio != '0'">
  <tr>
    <td colspan="2">
      <font size="-1">
<!-- Start cut. Between here and End cut is the code you
must replace with the secure search checkbox code. -->
        Search:
        <xsl:choose>
          <xsl:when test="$access='p'">
            <label><input type="radio" name="access" value="p" checked="checked"
              />public content</label>
          </xsl:when>
          <xsl:otherwise>
            <label><input type="radio" name="access" value="p"/>
              public content</label>
          </xsl:otherwise>
        </xsl:choose>
        <xsl:choose>
          <xsl:when test="$access='a'">
            <label><input type="radio" name="access" value="a"
              checked="checked" />public and secure content</label>
          </xsl:when>
          <xsl:otherwise>
            <label><input type="radio" name="access" value="a"/>
              public and secure content</label>
          </xsl:otherwise>
        </xsl:choose>
<!-- End cut. Between here and Start cut is the code you
must replace with the secure search checkbox code. -->
      </font>
    </td>
  </tr>
</xsl:if>
```

4. Replace the default code with the custom code shown below in boldface and bracketed by start/end comments below. This replaces the default radio button's `choose` clause with similar logic appropriate for a checkbox.

```
<xsl:if test="$show_secure_radio != '0'">
  <tr>
    <td colspan="2">
      <font size="-1">
        <!-- Start paste. Between here and End paste is the correct
        location for the secure search checkbox code. -->
          <xsl:choose>
            <xsl:when test="$access='p'">
              <input type="checkbox" name="access" value="a" />
            </xsl:when>
            <xsl:otherwise>
              <input type="checkbox" name="access" value="a" checked="checked" />
            </xsl:otherwise>
          </xsl:choose>
          Include secure content
        <!-- End paste. Between here and Start paste is the correct
        location for the secure search checkbox code. -->
      </font>
    </td>
  </tr>
</xsl:if>
```

5. Edit the text or font as you choose. Do not change any of the code inside the `choose` statement.
6. Save the stylesheet. If you are using an external editor other than the Admin Console's XSLT Stylesheet Editor, you must also import the stylesheet into the associated front end.
7. Verify the changes by opening the search query page and performing a test search on secure content.

Specifying Query Parameters with XSLT

The default search box for a front end is defined by the associated XSLT stylesheet. You can modify the XSLT stylesheet to include selected parameters in every query submitted from the query box in a front end. Valid parameters are documented in the *Search Protocol Reference*.

Though it is not immediately apparent, the search box in the "Inline Frame Example page" on page 120 is customized in this way. Each query submitted from the search box explicitly retrieves selected meta tag values and limits the number of results to optimize the limited page space.

XSLT Code for Specifying Query Parameters

The source attribute of the inline frame tag in the “Inline Frame Example page” on page 120 points to the customized front end, *iframe-example-frontend*. This front end submits selected parameters as defined by the XSLT code shown in bold at the bottom of the following code example.

```
<!-- *** form_params: parameters carried by the search input form *** -->
<xsl:template name="form_params">
  <input type="hidden" name="filter" value="0"/>
  <xsl:for-each
    select="PARAM[@name != 'q' and
      @name != 'ie' and
      not(contains(@name, 'as_')) and
      @name != 'btnG' and
      @name != 'btnI' and
      @name != 'site' and
      @name != 'filter' and
      @name != 'swrnum' and
      @name != 'start' and
      @name != 'access' and
      @name != 'ip' and
      (@name != 'epoch' or $is_test_search != '') and
      not(starts-with(@name, 'metabased_'))]">
    <input type="hidden" name="{@name}" value="{@value}" />
    <xsl:if test="@name = 'oe'">
      <input type="hidden" name="ie" value="{@value}" />
    </xsl:if>
    <xsl:text>
  </xsl:text>
  </xsl:for-each>
  <xsl:if test="$search_collections_xslt = '' and PARAM[@name='site']">
    <input type="hidden" name="site" value="{PARAM[@name='site']/@value}"/>
  </xsl:if>

  <!--Add these fields to every query string submitted from this front end-->
  <input type="hidden" name="getfields" value="rank.price.fly-type" />
  <input type="hidden" name="num" value="3" />
</xsl:template>
```

When users click the search button in this front end, `getfields="rank.price.fly-type"` and `num="3"` are included in the search request along with the standard parameters. Consequently, the result set will be limited to three results per page, and the specified meta tags will be retrieved for each result.

Adding Query Parameters to a Search Form

To add query parameters to a front end by modifying the associated stylesheet:

1. Open the stylesheet in the Admin Console's XSLT Stylesheet Editor or your preferred editor.
2. Scroll to the following section heading:

```
<!-- *****
Search Parameters (do not customize)
***** -->
```

3. Within the Search Parameters section, find the section beginning with `<!-- *** form_params: parameters carried by the search input form *** -->`. This section contains an XSLT template named *form_params*.

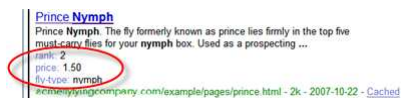
4. Scroll to the end of the form parameters template, `<xsl:template name="form_params">`.
5. Between the template's closing tag and the preceding `</xsl:if>` tag, add hidden input fields for all parameters you want to submit from the search form. For example, to retrieve all meta tag values in the results, add this line:

```
<input type="hidden" name="getfields" value="*" />
```

6. Save the stylesheet. If you are using an external editor other than the Admin Console's XSLT Stylesheet Editor, you must also import the stylesheet into the associated front end.
7. Verify the changes by submitting a search from the modified search box and checking the URL string for the specified parameters.

Displaying Meta Tag Values in Results

The custom front end for the “Inline Frame Example page” on page 120 is configured to show meta tags in the results. Each result displays the three meta tags, *rank*, *price*, and *fly-type*, that are explicitly retrieved by the custom search form:



To modify a front end to show meta tags in results:

1. Open the stylesheet in the Admin Console's XSLT Stylesheet Editor or your preferred editor.
2. Scroll to the section heading, `<!-- *** misc elements *** -->`.
3. Inside this section, set the variable `show_meta_tags` to a non-zero value:

```
<xsl:variable name="show_meta_tags">1</xsl:variable>
```

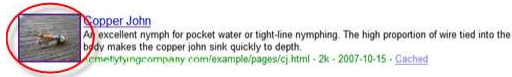
4. Save the stylesheet. If you are using an external editor other than the Admin Console's XSLT Stylesheet Editor, you must also import the stylesheet into the associated front end.
5. Make sure that the search parameter `getfields` is included in search requests, and that it is assigned the correct values to retrieve the desired meta tags. For more information on the `getfields` query parameter, see the *Search Protocol Reference*.
6. Verify the changes by submitting a search that returns results that contain meta tags, and check that the meta tag values are displayed.

Enhancing Search Results

Customization can serve your goals for the visual effect and content of results pages. This section describes two ways to enhance results pages: including image links in results, and replacing the result snippet with customized text.

Including an Image Link for Results

Placing small images in the search results enhances the visual appeal of the results page. Because experience has trained many users to click icons and pictures, it is helpful to make each image a link to the search result, just as the result title is a link to the result. The “HTML Form Example Page” on page 119 shows image links presented in the results pages.



XSLT Code for Image Links

The thumbnail-sized image links in the example pages are derived from a small block of XSLT code added after the variable declarations in the “single result” section of *example-frontend_stylesheet.en.xslt*. To find this block in the example stylesheet, search for the comment text “<!--Include image” and look for the following code:

```
<!--Include image from meta tag in results list-->
<xsl:for-each select="MT">
  <xsl:if test="@N='product-image' and @V!="">
    <a href="{ $protocol } : // { $escaped_url } ">
      
    </a>
  </xsl:if>
</xsl:for-each>
```

The XSLT image block iterates through each `MT` (meta tag) element in the result set and tests for non-empty values for the tag `product-image`. Then it includes the meta tag value inside a link to the results URL as represented by the concatenated variables `protocol` and `escaped_url`.

It is important to locate this XSLT block after the variable declarations located at the beginning of the “single result” XSLT block. Because the image block uses two of these variables, it will throw an error in the XSLT stylesheet editor if it is located before the variable is declared.

Image Meta Tag

Each product description page in this example includes a meta tag named `product-image` containing an absolute link to a thumbnail image. Because relative links do not work in the search results page, the meta tag values must be absolute links. For example, on *mywebserver* the prince nymph example page would include this meta tag:

```
<meta name="product-image" content="http://www.mywebserver.com/example/images/
prince_small.jpg" />
```

The exact meta tag name is important, because it is required as a value for the search request parameter `getfields` and also it is tested in a conditional clause in the XSLT code. All pages for which you want to present a results image must include a meta tag like this one, with the name properly referenced in the stylesheet and in `getfields`.

You must explicitly retrieve the meta tags you want from the result set in order for them to be transformed by the XSLT code. You can do this by adding `getfields="meta_tag_name"` to your search request, or use `getfields="*"` to retrieve all meta tags. See the *Search Protocol Reference* for more information on adding query parameters to the search request.

Adding Image Link Code to your XSLT Stylesheet

To modify a stylesheet to include image links in the results:

1. Open the stylesheet in the Admin Console's XSLT Stylesheet Editor or your preferred editor.
2. Scroll to the following section heading:

```
<!-- *****  
A single result (do not customize)  
***** -->
```

3. Within the single result section, scroll down to the following line:

```
<!-- *** Indent as required (only supports 2 levels) *** -->
```

4. Directly above this line, insert the following XSLT code:

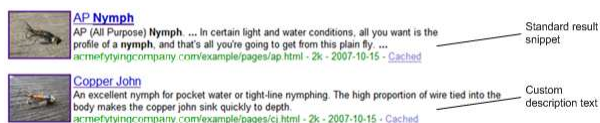
```
<!--Include image from meta tag in results list-->  
<xsl:for-each select="MT">  
  <xsl:if test="@N='meta_tag_name' and @V!=''">  
    <a href="{ $protocol }://{ $escaped_url }">  
        
    </a>  
  </xsl:if>  
</xsl:for-each>
```

5. Replace *meta_tag_name* with the name you have assigned to the meta tags storing your image link data. See “Image Meta Tag” on page 130 for more information.
6. Set the height and width attributes of the image tag to values appropriate for your thumbnail images.
7. Save the stylesheet. If you are using an external editor other than the Admin Console's XSLT Stylesheet Editor, you must also import the stylesheet into the associated front end.
8. Verify the changes by opening the search query page and performing a test search on pages that include image meta tags.

Replacing the Result Snippet with Custom Text

Google's automated selection of a result snippet is a useful feature that helps users evaluate the relevance of a result. However, it only generates snippet from the first 300K of the content and you may prefer to force the result to show a product description or other specific custom text related to the result.

For the fictional fishing fly business of the example pages, imagine that writers in the marketing department create succinct product descriptions for each item. The short descriptions are stored in meta tags and the longer descriptions are stored in the product page text. If the product page for a fly doesn't have a description tag, the results page uses the automated snippet derived from the page text. This approach is illustrated in the “HTML Form Example Page” on page 119, in which some results include a custom description and some results includes a standard result snippet with ellipses and search terms displayed in boldface.



XSLT Code for Custom Text

The product descriptions in the results for the example are derived from a small block of XSLT code added to the “Snippet Box” section of the stylesheet. This custom block includes and augments the default code for displaying the snippet. Note the differences and similarities in the default snippet box and the customized version in these two examples:

Default Snippet Box

```
<!-- *** Snippet Box *** -->
<table cellpadding="0" cellspacing="0" border="0">
  <tr>
    <td class="s">
      <xsl:if test="$show_res_snippet != '0'">
        <xsl:call-template name="reformat_keyword">
          <xsl:with-param name="orig_string" select="S"/>
        </xsl:call-template>
      </xsl:if>
    </td>
  </tr>
</table>
```

Custom Description Snippet Box

```
<!-- *** Snippet Box *** -->
<table cellpadding="0" cellspacing="0" border="0">
  <tr>
    <td class="s">
      <!--Use description meta tag, if non-empty, in place of snippet-->
      <xsl:choose>
        <xsl:when test="MT/@N='description' and MT/@V!=''">
          <xsl:value-of select="MT[@N = 'description']/@V"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:call-template name="reformat_keyword">
            <xsl:with-param name="orig_string" select="S"/>
          </xsl:call-template>
        </xsl:otherwise>
      </xsl:choose>
    </td>
  </tr>
</table>
```

The custom description snippet block displays the description for each result that has a description meta tag, or `MT`. If the XSLT engine does not find a valid description tag, it calls the standard template rules to display the auto-generated snippet element `s`.

The custom version includes the rule to display a standard snippet, but applies that template only if there is no valid description tag in the result. If you use this approach, make sure you have only one instance of the `reformat_keyword` template that selects the snippet element.

Description Meta Tag

Most of the product pages in this example includes a meta tag named `description` that contains a brief description of the product. For example, the copper john nymph page includes this meta tag:

```
<meta name="description" content="An excellent nymph for pocket water or
tight-line nymphing. The high proportion of wire tied into the body makes
the copper john sink quickly to depth." />
```

All pages for which you want to present custom text must include a meta tag like the above example, with the name properly referenced in the stylesheet and retrieved in the search request using the query parameter `getfields`. For pages that do not contain valid description tags, the stylesheet uses the automated snippet derived from page text.

You must explicitly retrieve the meta tags in the result set in order for them to be transformed by the XSLT code. Do this by adding `getfields=description` to your search request. For more information on `getfields`, see the *Search Protocol Reference*.

Adding Custom Text Code to an XSLT Stylesheet

To modify a stylesheet to replace the result snippet with custom text:

1. Open the stylesheet in the Admin Console's XSLT Stylesheet Editor or your preferred editor.
2. Scroll to the section marked by the comment `<!-- *** Snippet Box *** -->`.
3. Replace the fourth line of this section, `<xsl:if test="$show_res_snippet != '0'">`, with the following text:

```
<xsl:choose>
  <xsl:when test="MT/@N='meta_tag_name' and MT/@V!=''">
    <xsl:value-of select="MT[@N = 'meta_tag_name']/@V"/>
  </xsl:when>
  <xsl:otherwise>
```

4. Replace the last line of this section, `</xsl:if>`, with the following text:

```
</xsl:otherwise>
</xsl:choose>
```

5. Replace `meta_tag_name` with the name you have assigned to the meta tags storing your custom description text. See "Description Meta Tag" on page 132 for more information.
6. Save the stylesheet. If you are using an external editor other than the Admin Console's XSLT Stylesheet Editor, you must also import the stylesheet into the associated front end.
7. Verify the changes by opening the search query page and performing a test search on pages that include description meta tags.

Display Links in Dynamic Navigation to Sort Results by Metadata

You may want to provide a link to allow users to sort the results automatically by some metadata value. If you also have Dynamic Navigation enabled, you can modify the XSLT code to include a sort link, as shown in the following figure.



Note: If you provide Dynamic Navigation along with sorted results, you may see an apparent inconsistency whereby values in the Dynamic Navigation pane do not appear to correspond to any document when sorted. This is because the Dynamic Navigation pane considers more documents than the search results. For this reason, this integration is not provided in the default XSLT.

To enable this link, you must make the following changes to the default XSLT, by adding the bolded text to the stylesheet:

1. Add the bolded text between the two unbolded fragments:

```
.dn-block {
  display: block;
}
.dn-sort-txt {
  background-color: #E5ECF9;
  font-weight: normal;
  font-size: 90%;
  width: 100%;
}
.ac-renderer {
  background: white;
  border-bottom: 1px solid #558BE3;
```

2. Move the line `<xsl:template name="sort_by">` from above the `<xsl:variable name="sort_by_url">` definition to below it.

3. Add the bolded text between the two unbolded fragments:

```
<xsl:variable name="sort_by_relevance_url">
  <xsl:value-of select="$sort_by_url"
  />&amp;sort=date%3AD%3AL%3Ad1</xsl:variable>

<xsl:variable name="sort_by_date_url">
  <xsl:value-of select="$sort_by_url"
  />&amp;sort=date%3AD%3AS%3Ad1</xsl:variable>

<table><tr valign='top'><td>
<span class="s">
<xsl:choose>
  <xsl:when test="/GSP/PARAM[@name = 'sort' and starts-with(@value, 'meta')]">
    <a ctype="sort"
href="{ $gsa_search_root_path_prefix }?{$sort_by_date_url}">{{MSG_sort_by_date}}
</a>
    <xsl:text> / </xsl:text>
    <a ctype="sort"
href="{ $gsa_search_root_path_prefix }?{$sort_by_relevance_url}">{{MSG_sort_by_relevance}}</a>
  </xsl:when>
  <xsl:when test="/GSP/PARAM[@name = 'sort' and starts-with(@value, 'date:D:S')]">
```

4. Add the bolded text above the unbolded fragment:

```

<xsl:template name="dn_meta_sort_link">
  <xsl:param name="meta_name"/>
  <xsl:param name="display_name"/>
  <xsl:param name="tag_type"/>
  <xsl:param name="direction"/>
  <xsl:param name="language"/>

  <xsl:variable name="numeric">

    <xsl:choose>
      <xsl:when test="$tag_type = 1">N</xsl:when>
      <xsl:when test="$tag_type = 2">N</xsl:when>
      <xsl:when test="$tag_type = 3">F</xsl:when>
      <xsl:otherwise>D</xsl:otherwise>
    </xsl:choose>
  </xsl:variable>

  <a ctype="sort"
href="{ $gsa_search_root_path_prefix }?{ $sort_by_url } & sort=meta%3A
{ $meta_name }%3A{ $direction }%3A%3A{ $language }%3AD%3A{ $numeric }">
Sort Results by <xsl:value-of select="$display_name"/></a>
</xsl:template>

<xsl:template match="PMT">
  <xsl:param name="dn_tokens"/>
  <xsl:param name="partial_count"/>

```

5. Add the bolded text between the two unbolded fragments:

```

<xsl:choose>
  <xsl:when test="@IR = 1">
    <ul class="dn-attr">
      <li class="dn-attr-hdr"><span class="dn-attr-hdr-txt"><xsl:attribute
name="title"><xsl:value-of select="@DN" disable-output-
escaping="yes"/>
</xsl:attribute><xsl:value-of select="@DN"/></span>
<div><span class="dn-sort-txt">
  <xsl:call-template name="dn_meta_sort_link">
    <xsl:with-param name="meta_name" select="$pmt_name"/>
    <xsl:with-param name="display_name" select="@DN"/>
    <xsl:with-param name="tag_type" select="@T"/>
    <xsl:with-param name="direction">A</xsl:with-param>
    <xsl:with-param name="language" select="/GSP/PARAM[@name='ulang']/
@value"/>
  </xsl:call-template>
</span></div></li>
      <xsl:apply-templates select="PV">
        <xsl:with-param name="pmt_name" select="$pmt_name"/>
        <xsl:with-param name="dn_tokens" select="$dn_tokens"/>
        <xsl:with-param name="partial_count" select="$partial_count"/>
      </xsl:apply-templates>
    </ul>
  </xsl:when>

```

6. Add the bolded text between the two unbolded fragments:

```
<span class="dn-attr-hdr-txt"><xsl:attribute name="title">
  <xsl:value-of select="@DN" disable-output-escaping="yes"/>
</xsl:attribute><xsl:value-of select="@DN"/></span></div>
<div><span class="dn-sort-txt">
  <xsl:call-template name="dn_meta_sort_link">
    <xsl:with-param name="meta_name" select="$pmt_name"/>
    <xsl:with-param name="display_name" select="@DN"/>
    <xsl:with-param name="tag_type" select="@T"/>
    <xsl:with-param name="direction">A</xsl:with-param>
    <xsl:with-param name="language" select="/GSP/PARAM[@name='ulang']/
@value"/>
  </xsl:call-template>
</span></div></li>
</xsl:otherwise>
</xsl:choose>

<xsl:apply-templates select="PV[position() < $dyn_nav_max_rows + 1]">
  <xsl:with-param name="pmt_name" select="$pmt_name"/>
  <xsl:with-param name="header" select="@DN"/>
```

Configuring a Front End to Serve Secure and Public Results

By default a front end is configured to serve only public results. You can configure a front end to serve both secure and public results by editing the XSLT stylesheet.

To configure a front end to serve both secure and public results:

1. In the admin console, click **Search > Search Features > Front Ends**.
2. Click the **Edit** link next to a front end name in the list of front ends.
3. Scroll down to see the XSLT Stylesheet Editor.
4. Click the **Edit Underlying XSLT Code** link.
5. In the XSLT stylesheet, find the following code:

```
<xsl:variable name="access">
  ...
</xsl:otherwise>p</xsl:otherwise>
```

6. Make the following change to the code:

```
<xsl:otherwise>a</xsl:otherwise>
```

7. Click **Save**.

The modified XSLT stylesheet takes effect after 15 minutes. To see the result immediately, you can add `&proxyreload=1` at the end of the search string and resubmit the search.

Disabling Filtering for a Front End

Google uses automatic duplicate snippet filtering and duplicate directory filtering in search results. Duplicate snippet filtering is used to display only the most relevant document when multiple documents contain identical titles as well as the same information in their snippets in response to a query. Duplicate directory filtering is used to display only the two most relevant results for a directory when there are many results in a single web directory.

You can disable duplicate snippet filtering and duplicate directory filtering for a search query by using the query parameter `filter=0`, as described in *Search Protocol Reference*. In addition, the search appliance supports disabling filtering for a front end by editing the XSLT stylesheet. See the following sections for steps for:

- “Disabling Filtering for a Search Results Page” on page 137
- “Disabling Filtering for an Advanced Search Results Page” on page 137

After editing the XSLT stylesheet, you might have to reload the stylesheet to see your changes. See “Reloading the XSLT Stylesheet” on page 138 for more information.

Disabling Filtering for a Search Results Page

To disable duplicate snippet filtering and duplicate directory filtering for a search results page:

1. Open the XSLT stylesheet in an editor.
2. Locate the `<xsl:template name="form_params">` element.
3. Add the following XSLT code right before the closing tag `</xsl:template>`.

```
<input type="hidden" name="filter" value="0"/>
```

The following example shows the modified code.

```
<xsl:template name="form_params">
...
<input type="hidden" name="filter" value="0"/>
</xsl:template>
```

4. Save your changes.

Disabling Filtering for an Advanced Search Results Page

To disable duplicate snippet filtering and duplicate directory filtering for a search results page:

1. Open the XSLT stylesheet in an editor.
2. Locate the following comment.

```
<!--====Advanced Search Options====-->
```

3. Add the following line before the comment.

```
<input type="hidden" name="filter" value="0"/>
The following example shows the modified code.
<input type="hidden" name="filter" value="0"/>
<!--====Advanced Search Options====-->
```

4. Save your changes.

Reloading the XSLT Stylesheet

After saving the stylesheet, you might need to add `&proxyreload=1` to the search URL to reload the XSLT stylesheet.

View the source of the search page to see that it includes the following tag: `<input type="hidden" name="filter" value="0">`.

Reference

This section contains reference materials, including information on setting up and comparing the example pages and resources.

Optional Example Materials

You do not need to set up the example materials in order to read and understand the custom code, but it may be helpful to see the customizations in the same context described in this document. Depending on whether you begin with *index.html* or *iframe_index.html*, the example site demonstrates the customizations depicted in the “HTML Form Example Page” on page 119 or the “Inline Frame Example page” on page 120.

Setting up the Example

To view the search functionality of the example site, you must deploy the example web pages in a web server and implement the example stylesheets in a search appliance.

To set up the materials for testing in your search appliance:

1. Copy the example web pages, images and CSS stylesheet to a web server.
2. In the example files, search and replace all references to *appliance_hostname* with the host name of your search appliance.
3. In the example files, search and replace all references to *webserver_hostname* with the host name of your web server.
4. Crawl the folder containing these materials.
5. Create copies of the example stylesheets, and remove the `.txt` extension from the filenames. For example, *example-frontend_stylesheet.en.xslt.txt* becomes *example-frontend_stylesheet.en.xslt*.
6. Create a front end named “example-frontend” and import the stylesheet *example-frontend_stylesheet.en.xslt*. See “Customizing the User Interface” on page 92 for more information on creating front ends.
7. Create a front end named “iframe-example-frontend” and import the stylesheet *iframe-example-frontend_stylesheet.en.xslt*.

After you have set up the example materials, run a test query on them by opening *index.html* and typing “nymph” or other key words from the example pages into the search box. The expected result looks like the “HTML Form Example Page” on page 119. To view the inline frame example, use the search box displayed in */pages/iframe_index.html*.

Example Contents

The example consists of web pages and stylesheets designed for a fictional e-commerce site dedicated to selling fishing flies. It includes files described below.

Example index page and stylesheets:

- index.html
- example-frontend_stylesheet.en.xslt.txt
- iframe-example-frontend_stylesheet.en.xslt.txt

Example CSS:

- flystyle.css

Example image files:

- adams_large.jpg
- adams_small.jpg
- ak_bugger_large.jpg
- ak_bugger_small.jpg
- ap_large.jpg
- ap_small.jpg
- boss_large.jpg
- boss_small.jpg
- bugger_large.jpg
- bugger_small.jpg
- caddis_large.jpg
- caddis_small.jpg
- cj_large.jpg
- cj_small.jpg
- dark_lord_large.jpg
- dark_lord_small.jpg
- fox_poopah_large.jpg
- fox_poopal_small.jpg
- prince_large.jpg
- prince_small.jpg
- royal_large.jpg
- royal_small.jpg
- sculpin_large.jpg
- sculpin_small.jpg

Example pages:

- adams.html
- ak_bugger.html
- ap.html
- boss.html
- bugger.html
- caddis.html
- cj.html
- dark_lord.html
- flyframe.html
- fox_poopah.html
- prince.html
- royal.html
- sculpin.html

Comparisons of Default and Customized Stylesheets

To locate and review customized XSLT code, it can be helpful to have a side-by-side comparison of the default and customized stylesheets. Google provides sxse, a side-by-side comparison tool at <http://code.google.com/p/sxse/>. You can also readily find other freeware tools for preparing such comparisons.

To view comparisons of the example XSLT stylesheets and the default search appliance stylesheet on which they are based, see the following pages:

- example-frontend_changes.html
- iframe-example-frontend_changes.html

Appendix A

Quick Reference

The Google Search Appliance has features that enable system administrators to enhance or personalize the search experience for end users. This chapter provides a quick reference to features, best practices, and Admin Console pages you can use to enhance or personalize the search experience.

Search Experience Features

The following table lists Google Search Appliance features you can use to enhance or personalize the search experience. 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
Alerts, enabling	Index > Alerts	"Enabling Alerts" on page 76
Advanced Search Reporting	Search > Search Features > Front Ends > Output Format	"Gathering Information about the Search Experience" on page 36
Date biasing	Search > Search Features > Result Biasing > edit	"Using Date Biasing" on page 73
Document previews	Search > Search Features > Document Preview Module	"Providing Document Previews" on page 81
Domain filters	Search > Search Features > Front Ends > Filters	"Restricting Search Results by Domain Name" on page 57
Dynamic navigation	Search > Search Features > Dynamic Navigation	"Using Dynamic Navigation to Help Users Explore Results" on page 49
Expert search	Search > Search Features > Expert Search	"Providing Expert Search for Users" on page 53
File type filters	Search > Search Features > Front Ends > Filters	"Restricting Search Results by File Type" on page 58
Filters	Search > Search Features > Front Ends > Filters	"Using Filters to Restrict Search Results" on page 56
Front ends	Search > Search Features > Front Ends	"Creating a Front End" on page 94

Feature	Admin Console Page	Reference
KeyMatch	Search > Search Features > Front Ends > KeyMatch	“Using KeyMatches to Guide Users to URLs” on page 43
Language bundles	Search > Search Features > Language Bundles	“Changing Languages for Query Expansion and Spelling Suggestions” on page 66
Language filters	Search > Search Features > Front Ends > Filters	“Restricting Search Results by Language” on page 58
Meta tag filters	Search > Search Features > Front Ends > Filters	“Restricting Search Results by Meta Tag” on page 58
Metadata biasing	Search > Search Features > Result Biasing > edit	“Using Metadata and Entity Biasing” on page 74
My Alerts link, showing	Search > Search Features > Front Ends > Output Format	“Showing the My Alerts Link” on page 76
OneBox modules, defining or importing	Content Sources > OneBox Modules	“Using OneBox Modules to Integrate Structured Results” on page 70
OneBox modules, selecting	Search > Search Features > Front Ends > OneBox Modules	“Using OneBox Modules to Integrate Structured Results” on page 70
Page layout helper	Search > Search Features > Front Ends > Output Format	“Working with the Page Layout Helper” on page 96
People search (deprecated)	Search > Search Features > Expert Search	“Using People Search” on page 54
Query expansion policy	Search > Search Features > Front Ends > Filters	“Setting the Query Expansion Policy for a Front End” on page 65
Query expansion, configuring	Search > Search Features > Query Settings	“Using Query Expansion to Widen Searches” on page 60
Query suggestions, providing	Search > Search Features > Front Ends > Output Format	“Providing Query Suggestions” on page 79
Query suggestions, managing	Search > Search Features > Suggestions	
Related queries	Search > Search Features > Front Ends > Related Queries	“Using Related Queries to Suggest Alternative Searches” on page 45
Remove URLs	Search > Search Features > Front Ends > Remove URLs	“Removing URLs from Search Results” on page 59
Result biasing	Search > Search Features > Front Ends > Filters	“Working with Result Biasing” on page 72
Result biasing policy, creating	Search > Search Features > Result Biasing	“Working with Result Biasing” on page 72
Source biasing	Search > Search Features > Result Biasing > edit	“Using Source Biasing” on page 72
Translation	Search > Search Features > Front Ends > Output Format	“Enabling Translation of Search Results” on page 66
User Results	Search > Search Features > User Results	“Providing User Results” on page 77

Feature	Admin Console Page	Reference
XSLT stylesheet editor	Search > Search Features > Front Ends > Output Format	"Working with the XSLT Stylesheet Editor" on page 106

Search Experience Administration Best Practices

The following table lists best practices for creating the search experience. For each best practice, 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 it.

Best Practice	Reference	Admin Console Page
Enable advanced search reporting	"Gathering Information about the Search Experience" on page 36	Search > Search Features > Front Ends > Output Format
Allow users to choose search results by topic	"Using Dynamic Result Clusters to Narrow Searches" on page 47	Search > Search Features > Front Ends > Output Format
Help user explore search results by using specific metadata attributes	"Using Dynamic Navigation to Help Users Explore Results" on page 49	Search > Search Features > Dynamic Navigation
Restrict search results by domain name, language, file type, or meta tag	"Using Filters to Restrict Search Results" on page 56	Search > Search Features > Front Ends > Filters
Present recommended links above search results	"Using KeyMatches to Guide Users to URLs" on page 43	Search > Search Features > Front Ends > KeyMatch
Add structured, real-time application data to search results	"Using OneBox Modules to Integrate Structured Results" on page 70	Content Sources > OneBox Modules and Search > Search Features > Front Ends > OneBox Modules
Enable searching for experts in your organization	"Providing Expert Search for Users" on page 53	Search > Search Features > Expert Search
Provide your own synonyms for use in query expansion	"Using Query Expansion to Widen Searches" on page 60	Search > Search Features > Query Settings
Set the query expansion policy for a front end	"Setting the Query Expansion Policy for a Front End" on page 65	Search > Search Features > Front Ends > Filters
Install and activate a language bundle	"Changing Languages for Query Expansion and Spelling Suggestions" on page 66	Search > Search Features > Language Bundles
Present alternative search terms above search results	"Using Related Queries to Suggest Alternative Searches" on page 45	Search > Search Features > Front Ends > Related Queries
Translate search results into the user's language in real time	"Enabling Translation of Search Results" on page 66	Search > Search Features > Front Ends > Output Format
Show thumbnail images of documents in search results	"Providing Document Previews" on page 81	Search > Search Features > Document Preview Module

Best Practice	Reference	Admin Console Page
Prevent specific URLs from appearing in search results	"Removing URLs from Search Results" on page 59	Search > Search Features > Front Ends > Remove URLs
Influence the order of documents in search results	"Using Result Biasing to Influence Result Ranking" on page 71	Search > Search Features > Result Biasing
Enable or disable result biasing for a front end	"Working with Result Biasing" on page 72	Search > Search Features > Front Ends > Filters
Identify problematic query topics for related queries, KeyMatches, and query expansion synonyms	"Analyzing Searches that Do Not Return Results" on page 84	Reports > Search Report
Create a user interface that focuses on your users	"Customizing the User Interface" on page 92	Search > Search Features > Front Ends > Output Format
Divide the search index into meaningful partitions	"Segmenting Data in the Search Index" on page 86	Index > Collections
Enable users to set up alerts	"Providing Alerts for End Users" on page 75	Index > Alerts
Enable users to add search results for certain keywords in a specific front end	"Providing User Results" on page 77	Search > Search Features > User Results
Enable search queries to auto-complete and query suggestions to appear as a user types in the search box.	"Providing Query Suggestions" on page 79	Search > Search Features > Front Ends > Output Format
Prevent click-jacking web attacks	"Enabling/Disabling Click-Jacking Defense" on page 70	Search > Search Features > Query Settings

Admin Console Pages

The following table lists search appliance Admin Console pages that are used to create the search experience. 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
Index > Collections	"Segmenting Data in the Search Index" on page 86
Search > Search Features > Front Ends > Filters	"Using Filters to Restrict Search Results" on page 56
Search > Search Features > Front Ends > Filters	"Setting the Query Expansion Policy for a Front End" on page 65
Search > Search Features > Front Ends > Filters	"Working with Result Biasing" on page 72
Search > Search Features > Front Ends > KeyMatch	"Using KeyMatches to Guide Users to URLs" on page 43
Search > Search Features > Front Ends > OneBox Modules	"Using OneBox Modules to Integrate Structured Results" on page 70

Admin Console Page	Reference
Search > Search Features > Front Ends > Output Format	"Gathering Information about the Search Experience" on page 36
Search > Search Features > Front Ends > Output Format	"Customizing the User Interface" on page 92
Search > Search Features > Front Ends > Output Format	"Using Dynamic Result Clusters to Narrow Searches" on page 47
Search > Search Features > Front Ends > Related Queries	"Using Related Queries to Suggest Alternative Searches" on page 45
Search > Search Features > Front Ends > Remove URLs	"Removing URLs from Search Results" on page 59
Search > Search Features > Query Settings	"Using Query Expansion to Widen Searches" on page 60
Content Sources > OneBox Modules	"Using OneBox Modules to Integrate Structured Results" on page 70
Search > Search Features > Document Preview Module	"Providing Document Previews" on page 81
Search > Search Features > Result Biasing	"Using Result Biasing to Influence Result Ranking" on page 71
Search > Search Features > Dynamic Navigation	"Using Dynamic Navigation to Help Users Explore Results" on page 49
Search > Search Features > Suggestions	"Providing Query Suggestions" on page 79
Index > Alerts	"Providing Alerts for End Users" on page 75
Search > Search Features > Language Bundles	"Changing Languages for Query Expansion and Spelling Suggestions" on page 66
Reports > Search Reports	"Analyzing Searches that Do Not Return Results" on page 84
Search > Search Features > User Results	"Providing User Results" on page 77
Search > Search Features > Expert Search	"Providing Expert Search for Users" on page 53

Index

Symbols

&client search parameter 94
&proxystylesheet search parameter 94
&site parameter 87

A

Advanced Search link 10
advanced search page 92
advanced search reporting

- baseline measurements 40
- click types 38
- custom click types 39
- enabling for a front end 42
- exporting 42
- generating reports 41
- method 40
- using 36–42
- values 37

alerts

- description 24
- enabling 76
- My Alerts link 76
- providing 24
- using 75–77

all authorization mode 50
alternative search terms 16, 45
authentication

- alerts 75
- user results 78

authorization

- all authorization mode 50
- dynamic navigation 50
- dynamic result clusters 47
- fast authorization mode 50
- performance 83

automatic self-learning scorer 42
average click rank 40

B

best practices checklist 32

blacklists

- end-of-line characters 65
- query expansion 63
- query suggestions 80
- synonyms 19
- URLs in the index 86

built-in features 28

C

cached page header 92
checklist of best practices 32
click-jacking defense 70
clicks

- analyzing 36–42
- average click rank 40
- interpreting data 39
- types 38

collections

- description 14
- menu for searching 100
- searching 87
- searching multiple 87
- setting up 86–88
- using with front ends 14

Content Sources > Web Crawl > Start and Block
URLs page 86
customizing

- process 113–114
- search experience 11
- user interface 96–115

D

date biasing

- dates used 74
- description 23
- using 73

default front end 10, 14
directories, duplicates 20
diverse end users 11
document dates 74

- document previews
 - description 21
 - using 81
- domain filter 57
- duplicate directories filter 20, 56, 137
- duplicate snippet filter 20, 56, 137
- dynamic navigation
 - all authorization mode 50
 - authorization 50
 - configuration 51
 - description 18
 - enabling 51
 - fast authorization mode 50
 - query expansion 52
 - secure search 50
 - show or hide 103
 - showing 52
 - sort results links 133
 - using 49–52
- dynamic result clusters
 - description 18
 - enabling 18
 - enabling and disabling 47
 - show or hide 102
 - using 47–48

E

- end users
 - alerts 77
 - diverse search experiences 11
 - focusing on 9
 - satisfaction survey 88
- entity biasing 74
- entity recognition 75
- expert search
 - configuration 53
 - data sources 53
 - description 21
 - enabling 53
 - using 53
- external OneBox modules 21
- external provider 71

F

- fast authorization mode 50
- feedback from users 88
- file type filter 57, 58
- filters
 - description 22
 - disabling 137
 - domain 57
 - file type 57
 - language 57
 - meta tag 57
 - using 56–59
- Filters tab 57, 66
- forum, Google Search Appliance 90

- front ends
 - advanced search reporting 42
 - alerts 75
 - and collections 14
 - creating 94
 - default 14
 - description 14
 - dynamic navigation 49, 52
 - dynamic result clusters 47
 - elements defined in 27
 - expert search 53
 - filters 56
 - KeyMatch 43
 - multiple 36
 - number of 15
 - OneBox modules 71
 - personalization 34
 - query expansion policy 65
 - query suggestions 79
 - related queries 45
 - result biasing 72
 - search experience 35
 - search parameters 94
 - translation of search results 66
 - user interface 93
 - user results 77

G

- Global Attributes, Page Layout Helper 25, 97–98
- Google logo 98

H

- host crowding 56

I

- indented search results 56
- index
 - clean 86
 - expert search 53
 - languages 58
 - managing 85–88
 - related queries 45
 - removing URLs 60
 - segmenting 86
- Index > Alerts page 76
- Index > Collections page 15, 87
- Index > Document Dates page 74
- internal OneBox modules 21
- internal provider 71

K

- KeyMatch tab 44
- KeyMatches
 - changing appearance 44
 - creating 17
 - description 17
 - identifying 44
 - using 43–45

L

- language bundles 66–69
- language filter 58
- languages
 - filters 57
 - query expansion 61
 - supporting multiple 13
 - user interface 113

M

- Macintosh style end-of-line characters 65
- meta tags
 - biasing 74
 - dynamic navigation 49
 - expert search 53
 - filters 57, 58
 - query expansion 59
- metadata
 - biasing 74
 - dynamic navigation 49
 - dynamic result clusters 49
 - expert search 53
 - filters 58
 - query expansion 59
- metadata and entity biasing
 - description 23
 - using 74
- metadata, sort links 133
- multiple search experiences 11
- My Alerts link 76

N

- narrowing searches 18, 47

O

- OneBox modules
 - adding click types 39
 - description 20
 - implementing 21
 - using 70–71
- Output Format tab 97

P

- Page Layout Helper 66, 97–104
 - description 25, 95
 - dynamic result clusters 48
 - My Alerts link 76
 - using 96–97
- partialfields search parameter 58
- people search 54
- percentage of satisfied clicks 40
- performance, evaluating 88–90
- personalization 9, 34

Q

- query expansion
 - blacklist 63
 - built-in files 60
 - contextual 69
 - customized 62
 - description 19
 - dynamic navigation 52
 - file formats 62
 - front ends 65
 - meta tags 59
 - non-contextual 69
 - policy settings 65
 - preconfigured local files 60, 61
 - stopwords 64
 - using 60–66
- query suggestions
 - blacklist 80
 - exporting 80
 - using 79–81

R

- real-time data, integrating 20
- related queries
 - changing appearance 46
 - creating 17
 - description 16
 - identifying 46
 - using 45–46
- Related Queries tab 46
- relevance
 - automatic self-learning scorer 37
 - collections 87
 - fine-tuning 42
 - result biasing 71
 - search results 19
- remove URLs 22, 59
- Reports > Search Log page 42
- Reports > Search Reports page 42, 44, 85
- requiredfields search parameter 58
- result biasing
 - configuring 23
 - date biasing 73
 - description 23
 - metadata and entity biasing 74
 - policy 72
 - source biasing 72
 - using 71–75
- results rankings
 - influencing 23, 71

S

- scoring, fine-tuning 42
- Search > Search Features > Document Preview Module page 81
- Search > Search Features > Dynamic Navigation page 49, 51, 59
- Search > Search Features > Expert Search page 53
- Search > Search Features > Front Ends > Filters page 72

- Search > Search Features > Front Ends page 14, 42, 44, 46, 48, 76, 79, 94, 97
- Search > Search Features > Language Bundles page 69
- Search > Search Features > Query Settings page 52, 59, 61, 70
- Search > Search Features > Result Biasing page 72
- Search > Search Features > Suggestions page 80
- Search > Search Features > User Results page 77, 78
- Search Box, Page Layout Helper 25, 99–100
- search experience
 - basic 9–10
 - creating multiple 11
 - customization 11
 - description 8
 - managing 14
 - process 29–31
- search page 92
- search parameters 127
- search query 29
- search results
 - custom snippets 131
 - enhancing 19–24
 - image link 130
 - indented 56
 - listing elements 103
 - optimizing speed 83
 - page 92
 - page bottom elements 104
 - page divider 102
 - sidebar elements 103
 - snippets 56
 - top elements 101
 - top navigation links 102
 - tuning 83–85
- Search Results, Page Layout Helper 25, 101–104
- Search Tips link 10
- Search Within Results link 10
- search within results page 92
- searches
 - collections 87
 - improving 16
 - multiple collections 87
 - narrowing 18
 - no results returned 84
 - refining 49
 - unsuccessful 85
 - widening 19, 60
- site search parameter 87
- snippets
 - duplicates 20, 56
 - translation 66
- sort results by metadata 133
- source biasing
 - description 23
 - using 72
- spell checker 16
- spelling suggestions 16
- stopwords for synonyms 19

- structured data, integrating 20, 70
- survey, user satisfaction 89
- synonyms
 - blacklist 19
 - built-in dictionaries 19, 60
 - customized 19, 60
 - stopwords 19
 - uploading lists 19

T
translation of search results 66

U
updating a search appliance 91

URLs

- guiding users to 17, 43
- removing from results 22, 59
- removing from the index 23

user alerts

- description 24

user interface

- advanced customization 118–140
- customizing 24–26
- default 96
- description 92–95
- design principles 116–117
- font face 98
- front ends 93
- Google logo 98
- header and footer 98
- language 113
- radio buttons 100
- viewing changes 114

user results

- authentication 78
- description 21
- moderation 78
- using 77–79

users

- diverse search experiences 11
- feedback 88
- focusing on 9
- satisfaction survey 89

W
web pages, integrating search 121
whitelist of URLs in the index 86
widening searches 19, 60
wildcard search 82

X
XML search results 31, 94

- XSLT stylesheet
 - applying 31
 - customizing 105-112
 - description 95
 - exporting 107
 - importing 107
 - managing 115
 - migrating 115
 - reloading 138
 - restoring 108
 - user interface 94
 - variables 108-112
- XSLT Stylesheet Editor 25
 - dynamic result clusters 48
 - My Alerts link 77
 - using 106