

The Oracle logo is displayed in a bold, red, sans-serif font. A horizontal line is positioned directly below the logo.

ORACLE®

ATG WEB COMMERCE

Search

Version 10.0.2

Administration Guide

**Oracle ATG
One Main Street
Cambridge, MA 02142
USA**

ATG Search Administration Guide

Document Version

SEARCH10.0.2 SEARCHADMINv1 4/15/2011

Copyright

Copyright © 1997, 2011, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark licensed through X/Open Company, Ltd.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/us/corporate/accessibility/index.html>.

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/support/contact.html> or visit <http://www.oracle.com/accessibility/support.html> if you are hearing impaired.



Contents

1	Introduction	1
	Audience	1
	Document Conventions	1
	More Information	2
2	Overview	3
	How ATG Search Works	3
	Search Request Life Cycle	4
	ATG Search Components	7
	Project Administration	7
	Search Workbench	8
	Security in ATG Search	9
	ATG Search for Multisite	9
	Accessing Search Administration	12
	Accessing the Search Workbench	12
3	Managing Search Projects	15
	Search Project Overview	15
	Before You Begin—Global Settings	16
	Creating Search Projects	17
	Copying and Linking Search Projects	18
	Viewing Search Projects	19
	Deleting Search Projects	20
	Adding Content to Search Projects	20
	Creating New Content Sets	21
	Deleting Content Sets	22
	Adding File System Content to Content Sets	22
	Adding Repository Content to Content Sets	27
	Viewing Content	28
	Adding Customizations to Content	28
	Removing Customizations from Content	28
	Adding Language Support and Dictionary Adaptors	29
	Associating Content with Sites in Site Administration	31
4	Working with Indexes in Search Administration	33
	Estimating Index Size	33

Applying Customization Data	35
Building Indexes	35
Creating Automatic Indexing Rules	39
Using Index Review	40
Content Summary	41
Index Audit	41
Content Inspection	42
IndexingOutputConfig Analysis	44
Viewing Indexing Histories	46
Deleting Indexes and Histories	46
Deleting Index Histories Manually	46
Configuring Index Cleanup Rules	47
Redeploying Old Indexes	47
5 Indexing Repository Data	49
Creating XHTML Documents from Repository Items	49
Specifying the Definition File and the Repository	50
Definition File Format	50
XHTML Document Format	51
Specifying Properties for Indexing	53
Specifying Multi-Value Properties	53
Specifying Map Properties	54
Specifying Properties of Item Subtypes	55
Specifying a Default Property Value	56
Specifying Non-Repository Properties	56
Suppressing Properties	57
Renaming an Output Property	57
Translating Property Values	58
Loading the XHTML Documents into ATG Search	59
Configuring Document Loading	60
Viewing the XHTML Output	61
6 Customizing the XHTML Output	65
Using Property Accessors	65
FirstWithLocalePropertyAccessor	66
GenerativePropertyAccessor	66
Using Variant Producers	67
LocaleVariantProducer	67
Accessing the Context Object	67
Using Property Formatters	68
Using Property Value Filters	69
UniqueFilter	70
ConcatFilter	70
UniqueWordFilter	71
HtmlFilter	72



7	Indexing ATG Commerce Catalog Data	73
	Indexing the Product Catalog Repository	73
	SKU-Based versus Product-Based Indexing	73
	Handling Catalog-Specific Properties	75
	Determining Which Properties to Index	76
	Excluding Uncategorized Products from Indexing	78
	Indexing a Multisite Catalog Repository	79
	Indexing Price Data in Price Lists	80
	Creating the Price Property	80
	Indexing Price List Data	81
8	Managing Search Environments	83
	Creating Search Environments	83
	Configuring Content Adapters for New Search Environments	86
	Creating Content Adapters	87
	Adding New Adapters to Adapter Registry	88
	Attaching Environments to Search Projects	88
	Starting and Stopping Search Environments and Engines	89
	Viewing Search Environments	89
	Moving Search Environments Between Projects	90
	Deleting Search Environments	91
9	Topic Sets	93
	Adding Topic Sets	93
	Creating New Topic Sets in the Workbench	94
	Importing Topic Sets as XML Files	94
	Adding Topic Sets to Projects	95
	Modifying Topic Sets	95
	Exporting Topic Sets	95
	Adding Topics	95
	Moving Topics	96
	Renaming Topic Sets and Topics	97
	Deleting Topic Sets and Topics	97
	Automatically Associating Content with Topics	98
	Adding Rule Patterns	98
	Understanding Topic Weights	99
	Understanding Rule Pattern Syntax	100
	Example Patterns for Topic Rules	101
	Enabling and Disabling Rule Patterns	102
	Deleting Rules	102
	Using Macros in Patterns	102
	Adding Global Macros	102
	Adding Local Macros	103
	Deleting Macros	103

10	Dictionaries	105
	Why Use Dictionaries	105
	Why Create a Dictionary	106
	Why Add Terms	106
	Why Add Synonyms and Child Terms	107
	Using Dictionary Inspection Tools	109
	Looking Up Dictionary Terms	109
	Evaluating Content Processing	111
	Analyzing Search Text	113
	Managing Dictionaries	117
	Adding Dictionaries	117
	Editing Dictionaries	118
	Duplicating Dictionaries	118
	Exporting Dictionary Data	118
	Importing Dictionary Data	119
	Deleting Dictionaries	119
	Managing Terms	119
	Adding Terms	119
	Adding Child Terms	120
	Modifying Terms	121
	Moving Terms	122
	Deleting Terms	122
	Managing Synonyms	122
	Adding Synonyms to Terms	122
	Modifying Synonyms	124
	Deleting Synonyms	124
	Understanding Term Weights	125
	Managing Term Weight Sets	126
	Default Term Weight Sets	126
	Adding Weight Sets	126
	Copying Weight Sets	126
	Importing Weight Sets	127
	Adding Terms to Weight Sets	127
	Deleting Terms from Weight Sets	127
	Deleting Weight Sets	128
11	Text Processing Option Sets	129
	Adding Text Processing Option Sets	130
	Adding Project TPO Sets	130
	Importing Project TPO Sets	131
	Adding Content TPO Sets	131
	Importing Content TPO Sets	132
	Using Table Structure Parsing	133
	Duplicating Text Processing Option Sets	135
	Modifying Text Processing Option Sets	136
	Exporting Text Processing Option Sets	136



Deleting Text Processing Option Sets	136
12 Facet Sets	139
Creating Facet Sets	140
Adding Child Facets	140
Moving Facets	143
Modifying Facet Sets and Facets	143
Deleting Facet Sets and Facets	143
Importing Facet Sets	144
Exporting Facet Sets	144
Duplicating Facet Sets	145
Viewing Facet Set Adaptors	145
13 Query Rules	147
Creating Query Rule Sets	147
Creating a Rule Set	148
Adding Rule Groups to Rule Sets and Rule Groups	148
Adding Rules to Rule Groups	149
Rule Example: Adding a Property Constraint	159
Rule Example: Redirecting Users	160
Rule Example: Promoting Preferred Results	160
Using Macros in Rules	161
Enabling and Disabling Query Rules, Patterns, and Actions	161
Importing Query Rules	162
Exporting Query Rules	163
Moving Query Rule Sets and Rule Groups	163
Deleting Query Rule Sets	163
Deleting Query Rule Groups	164
Deleting Query Rules	164
Deleting Patterns and Actions	164
14 Reporting	165
Accessing Reports	165
Running Reports	167
Data Logging for Reports	167
Search Logging Components and Messages	167
Enabling and Disabling Search Logging	171
Configuring Log File Location	172
Search Loader Components	172
Scheduling Data Loading	173
Search Reports	173
All Selected Topics	173
Content Views by Topic	174
Most Frequent Search Terms	174
Most, Least, and Last Selected Topics	175
Most, Least, and Last Viewed Content	176



Search Activity Over Time	177
Search Activity Summary	178
Search Response Time Trends	179
Search Terms Not in the Dictionary	180
Search Terms Grouped by Topic	181
Search Terms Grouped by Type	181
Search Terms Grouped by Word Count	182
Search Terms Consisting of Custom Terminology	183
Search Terms with No Results	183
Search Terms with Numbers	184
Search Terms with Special Compound Terms	185
Search Terms Existing only as Thesaurus Entries	185
Search Terms Not in the Indexed Content	186
Searches Trending Up and Down	187
Site Search Usage Overview	187
15 Troubleshooting and Performance Tuning	189
Finding Debugging Information	189
ATG Search Administration	189
SearchEngineService	190
Enabling Search Logging	191
Enabling Auxiliary Data Error Reporting	194
Search Engine Performance Logging	195
Using Diagnostics Capture	195
Viewing Index Data	197
Troubleshooting Repository Indexing	198
Troubleshooting Connection Problems	198
Check Data Source Configuration	198
Check Lock Management Configuration	199
Check Event Server Connections	200
Troubleshooting Search Results	200
Search Returns No Results	201
Search Returns Unexpected Results	202
Search Error Messages	202
Connection Refused	202
No ManagedConnections Available Within Configured Blocking Timeout	203
Could Not Prepare Environment	203
Cannot Find Remote IndexingOutputConfig	203
Locale Errors	204
Not Enough Space	204
Out of Memory Errors	204
This Server is Under Too Much Load for the Multiplexed...	204
ConfigurationClaimedException or IndexingException	205
Unable to Suggest Port for Engine	206
Search Performance Tuning	206
Configuring Multi-Threaded Indexing	206
Configuring the Number of Threads per Engine	206



Adjusting TCP Settings for Heavy Load on Windows	207
Disabling Index Compression	208
Adjusting Physical Partition Size	208
Configuring Indexing for Large Content Sets	208
Updating Search Settings on a Live Site	209
Appendix A: Indexable File Types	211
Word Processing	211
Text	214
Spreadsheet	214
Database	215
Raster Image	216
Vector Image	218
Presentation	219
Compressed Formats	220
Email Formats	220
Multimedia	221
Other	222
Appendix B: Using WGet for Web Site Indexing	223
Appendix C: Loading Customizations by Reference	225
Auxiliary Data by Reference	225
Merchandising Search Configurations by Reference	227
Appendix D: Implementing Type-Ahead in ATG Search	229
Creating a Type-Ahead Value List	229
Making Type-Ahead Available to Your Site	230
Glossary	233
Index	236





1 Introduction

This guide explains how to index content so that it can be searched on your site, and how to maintain your ATG Search system. It assumes that you have already installed and configured ATG Search as described in the [ATG Search Installation and Configuration Guide](#).

This chapter includes the following sections:

Audience

Document Conventions

More Information

Note: This book does not address topics that are specific to using ATG Search with ATG Merchandising. See the [ATG Merchandising Guide for Business Users](#) and the [ATG Merchandising Administration Guide](#) for information on that product and its specialized search features.

Audience

This guide is intended for administrative users responsible for maintaining the ATG Search installation and for managing the end-user search experience, including:

- Determining what content can be searched
- Enhancing searching with features such as topic sets and custom dictionaries
- Setting up rules to ensure that content is indexed appropriately
- Using system maintenance features such as reporting
- Viewing and analyzing search reports

If you are using the multisite feature, search administrators may also be involved in preparing search indexes for use with Site Administration (see the [ATG Multisite Administration Guide](#)).

Document Conventions

This guide uses the following conventions:

- <ATG10dir> refers to your ATG installation directory.



- *Server* or *ATG server* refers to an ATG application instance that is deployed to your application server in the form of an EAR file. See the [ATG Installation and Configuration Guide](#) for information. ATG documentation commonly refers to the following server types:
 - Asset management server—Where Content Administration, Search Administration, and other such applications run. Sometimes called just the management server.
 - Agent-facing server—Used for ATG Knowledge and other agent-oriented applications
 - Customer-facing server or production server—Where your ATG application runs.

More Information

See the following guides for more information on ATG Search:

- [ATG Search Installation and Configuration Guide](#)—Installing ATG Search.
- [ATG Search Query Guide](#)— Constructing search queries and handling the results.



2 Overview

This chapter describes the features and components of ATG Search.

ATG Search is an industry-leading search engine that lets users zero in on the information they need, regardless of format, location or language. Multiple levels of analysis pinpoint answers and deliver them with sub-second response times.

This chapter includes the following sections:

[How ATG Search Works](#)

[Search Request Life Cycle](#)

[ATG Search Components](#)

[Security in ATG Search](#)

[ATG Search for Multisite](#)

[Accessing Search Administration](#)

[Accessing the Search Workbench](#)

How ATG Search Works

Searching means retrieving stored information using an input query. *Indexing* is the process of making documents and other content searchable. This searchable collection is then called the *index*. During indexing, ATG Search generalizes documents and other content into an abstraction called a *content item*. A content item consists of two main parts: *searchable text content* and *metadata*. Metadata includes the title, summary, and other associated properties. Examples of content items are products in an ATG Commerce catalog repository, HTML files, or structured data from a traditional database.

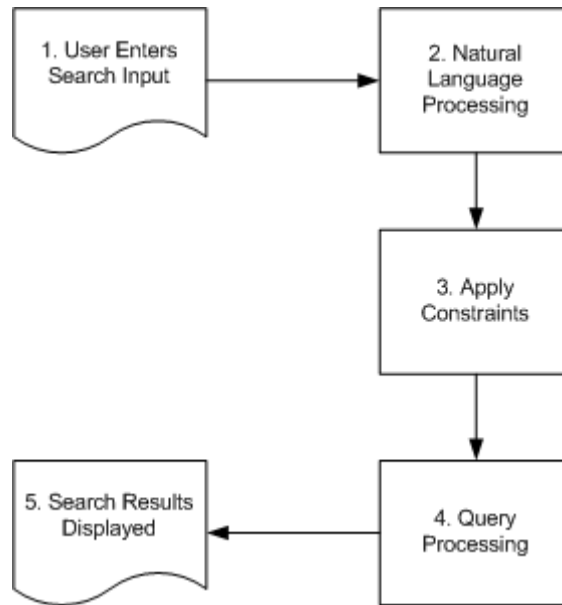
Within the index, content items are organized into document sets. You can think of these as virtual directory structures, which can offer a variety of routes to reach a particular content item. ATG Search creates some document sets from the physical location of the items, some sets from item metadata, and other sets using categorization results. Document sets allow end-users to search subsets of content and to browse without query input.

In ATG Search, end-user text queries are embedded by the client application within *requests*. A request may include parameters, processing options, constraints, security settings, and other information. The two primary request types are the Search Query and the View Item request. After receiving a request, ATG Search returns *responses*, which contain information that varies depending on the type of request. For a

Search Query, the response contains a list of results plus other information to drive the search application or user interface.

Search Request Life Cycle

This diagram provides a high-level overview of the processes used in answering search requests:



Search Request Processing

The sections that follow provide detailed information on each request processing step.

Note: This section does not include processing information specific to the Search Merchandising feature of ATG Merchandising. See the [ATG Merchandising Guide for Business Users](#) for information on Search Merchandising.

User Enters Search Input

ATG Search can accept several different types of search input. Two common types are:

- **Keyword**—The user enters one or more terms and clicks a submit button.
- **Browse**—The user clicks a link that contains search request information encoded in a URL. This information could include a product category.

The examples in this section use keyword searches, because that is familiar to most people and the most friendly to human readers.



Along with keywords or phrases to search on, the user can enter operators such as quote marks, + and ! symbols. The search form handlers (the programmatic part of the page on which the user enters search terms) can also add parameters behind the scenes, for example to identify the language the user is searching in, or to restrict the search to certain parts of the index.

The text the user enters is packaged up into a query request. The request is a Java object that is sent to ATG Search when the user clicks a submit button, follows a link, etc., then translated into XML by the search engine.

See....	Troubleshooting
<p>ATG Search Query Guide for information on the search form handlers.</p>	<p>You can use Search Reports to see what search terms your users enter. See the Reporting chapter in this guide.</p> <p>You can view the entire XML of a search query. See Enabling Search Logging in the Troubleshooting and Performance Tuning chapter of this guide.</p>

Natural Language Processing

Once it receives the submitted request, ATG Search analyzes it to determine what terms it has to search for. To do this, it performs a series of analyses on the text the user entered. Note that it is the final result of *all* of these steps that is used by the search engine to conduct the search.

This is essentially the same process used to index content, and it relies on a combination of the core search dictionary, any supplemental dictionaries you have created, and language-specific information.

This step includes the following processes:

1. Identify the words in the request. This step is necessary before ATG Search can perform any further processing. For example, “running the New York Marathon” can be broken down into the following words:

running
the
New York
marathon

Note that if it is specified as such in the dictionary, ATG Search can index and search for compound terms such as “New York” as if they were a single word.

2. Identify the root form of each of the identified words.
In the marathon example, the root form of *running* would be identified as *run*.
3. Map variant word forms and common misspellings to a single form of the word.



For example, if the user had mistakenly typed *marathan*, ATG Search could correct it to *marathon* before proceeding. (Note, however, that if the user puts the term in quotes, "marathan," ATG Search attempts to match on the misspelled version.)

4. Identify synonyms for each of the words, if any.

For example, your dictionary could include *race* as a synonym for *marathon*. Searches that include the term *race* would find results that include the term *marathon*.

5. The final step is to apply a weight to each of the terms in the request. This is a complex, proprietary process by which ATG Search decides how important each term in the request is.

For example, words like *the* and *it* are probably not important to the search, but a product name is.

The final result of natural language processing is a list of terms that includes the spell-corrected root forms of the original search terms, any synonyms for these terms, and weighting information.

See....	Troubleshooting
The Dictionaries chapter of this guide for steps 1-4.	See Using Dictionary Inspection Tools and Troubleshooting Search Results in this guide.
The Understanding Term Weights section of this guide for step 5.	
The Why Use Dictionaries section of this guide for information on when and how to use custom dictionaries.	

Apply Constraints

Now that it has identified the terms that it will compare to the index, ATG Search applies any constraints that were entered along with the request. Constraints can include site, catalog, or category information; operators such as quote marks and the ! symbol; query rules; and rank configurations.

The constraints are processed and added to the query information.

See....	Troubleshooting
For operators, see <i>User-Entered Query Operators</i> in the ATG Search Query Reference Guide .	Query and response XML; see Troubleshooting Search Results in this guide
For query rules, see the Query Rules chapter of this guide.	
For Merchandising Search Configurations, see the ATG Merchandising Guide for Business Users	



Query Processing

The compiled query information is compared against the index. The engine returns XML that includes the resulting matching products or statements, taking constraints into account, along with categorization, spelling, and query feedback information specified in the request.

The results are sorted in relevance order by default, but this behavior can be changed using request parameters. The results can be grouped to reduce duplication, and you can apply paging to prevent slowing down your site with large result lists.

The final results are displayed to the user.

See....	Troubleshooting
<p>The docSort attribute in the ATG Search Query Reference Guide for ordering results.</p> <p>The <i>Refining Search Results</i> section of the ATG Search Query Reference Guide for categorization, spelling, and query feedback.</p> <p>The sorting attribute in the ATG Search Query Reference Guide for grouping.</p> <p>The pageNum and pageSize attributes in the ATG Search Query Reference Guide for paging.</p>	<p>Response XML</p>

ATG Search Components

The ATG Search Administration interface consists of the following main components:

- Project Administration
- Search Workbench

Project Administration

The *search project* contains information about a set of searchable content you define, and the configuration information for that content, such as where it comes from, what rules are followed for its indexing, and which customization data is applied. *Project Administration* is the part of the Search Administration interface dedicated to project-related tasks.

Note: A search project is not a Content Administration project. Search projects do not contain deployable assets and do not have an associated workflow, but are a means to organize and configure a given set of indexable content.

The center of the project is an *index*, the searchable content produced by ATG Search processing. Each project has one index. The index is created from the *content sets*, which organize the content you want to



include in the index. *Content* refers to the original, unprocessed repositories, documents, etc. that you want to make searchable.

The project links the index with search *environments*, which perform the tasks of indexing content and answering end-user queries.

Projects also link an index with *customization data*, such as topic sets or specialized dictionaries. This customization data is created using the Workbench (see next section) or through other ATG applications such as Merchandising.

For information, see the [Managing Search Projects](#) and [Managing Search Environments](#) chapters of this guide.

Search Workbench

The *Search Workbench* is used for creating and working with customization data. Customizations are used to enhance the out-of-the-box configuration for features such as languages, dictionaries, and topic sets.

Customization is the process of optimizing your ATG Search implementation for more accurate search results. Customizations typically add information to the index (such as additional dictionaries or term weights) or affect the way your content is processed or accessed by end-users (for example, query rules, topic sets, or facet sets).

Note: For information on adding customization data to your search project, see [Adding Customizations to Content](#) in the [Managing Search Projects](#) chapter.

ATG Search Administration allows you to work with the following types of customization data:

- Topic Sets – Hierarchical organizations of content items, which can help your users search or browse content. See the [Topic Sets](#) chapter.
Note: This feature is used by the ATG Service applications.
- Dictionaries – Customized terminology that augments the supplied ATG Search dictionaries. See the [Dictionaries](#) chapter.
- Term Weights – Allow you to identify “stop words” or “noise words” that can reduce search effectiveness and should not be considered when determining a content item’s relevance. See the [Managing Term Weight Sets](#) section of the [Dictionaries](#) chapter.
- Query Rules – Allows specified query strings to trigger actions that can alter search behavior or return specialized results, such as promotional or sponsored links. ATG Advisor rules are implemented in ATG Search through query rules, which return Advisor resources within ATG Search’s results. See the [Query Rules](#) chapter.
- Facet Sets – Allow you to select the properties that will be used to return faceting information for faceted navigation. See the [Facet Sets](#) chapter for non-Merchandising uses of faceting, or the [ATG Merchandising Guide for Business Users](#) if you are using Merchandising.
- Text Processing Option Sets – Collections of parameter settings that control a wide variety of options related to content indexing and query processing. See the [Text Processing Option Sets](#) chapter.



- Merchandising Search Configuration – Used in conjunction with ATG Merchandising, allows you to customize the way search results are ranked.

Note: Merchandising Search Configurations can only be loaded by reference, and are not available in the Workbench. See [Appendix C: Loading Customizations by Reference](#).

- Auxiliary Data – Provides a way to change properties on items that have already been indexed, or to add type-ahead data to your index (see [Appendix D: Implementing Type-Ahead in ATG Search](#)).

Note: Auxiliary Data can only be loaded by reference, and is not available in the Workbench. See [Appendix C: Loading Customizations by Reference](#).

Security in ATG Search

ATG Search includes the Search Access role, which allows a user assigned that role to access Search Administration and the Workbench.

Roles are administered in the ATG Business Control Center; for information, see the [ATG Business Control Center Administration and Development Guide](#).

ATG Search for Multisite

If you are using ATG's multisite feature, you can associate a site with a particular content set within a search project, using one of the strategies outlined in this section:

- Single project, single content set—Use this strategy if you do not need to target searches for different sites.

Note: If you have data for multiple sites grouped into a single content set, you can still target searches to a particular site's data by using a constraint on the query request (see the [ATG Search Query Guide](#) for information). Keep in mind that this can only be used to narrow searching within a content set, not to combine data from separate content sets.

- Single project, multiple content sets—Each content set can contain content for one or more sites.

This approach is useful if you want to allow users to choose a site to search, or if you need to display both site-specific content and content that has been combined from multiple sites in your search results. You may want to separate your content into contents sets to segregate data that is always searched together from other site data.

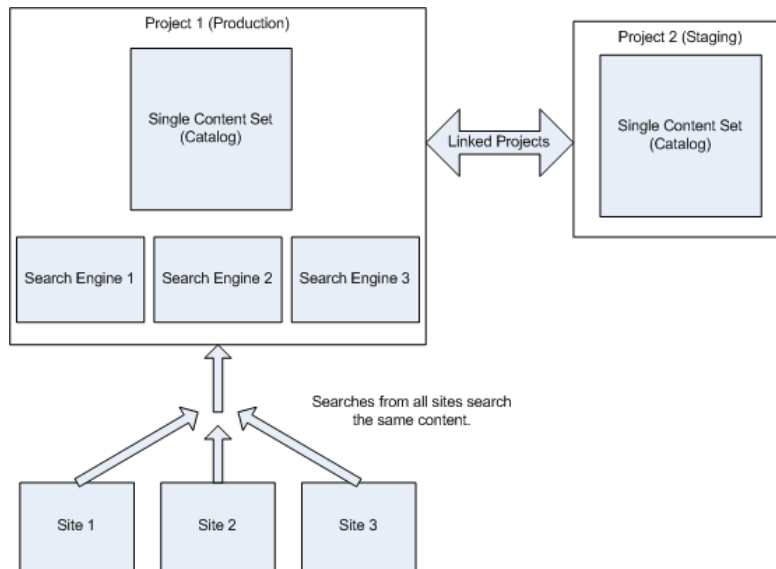
- Separate project per site—Use this approach only if you want to keep content for each site entirely separate. This may result in faster searches, since no constraints need to be applied to the searches, but keep in mind that search results from different projects cannot be combined.



The diagrams that follow show each of these configuration possibilities. For information on search projects, linked projects, etc., see the [Managing Search Projects](#) chapter.

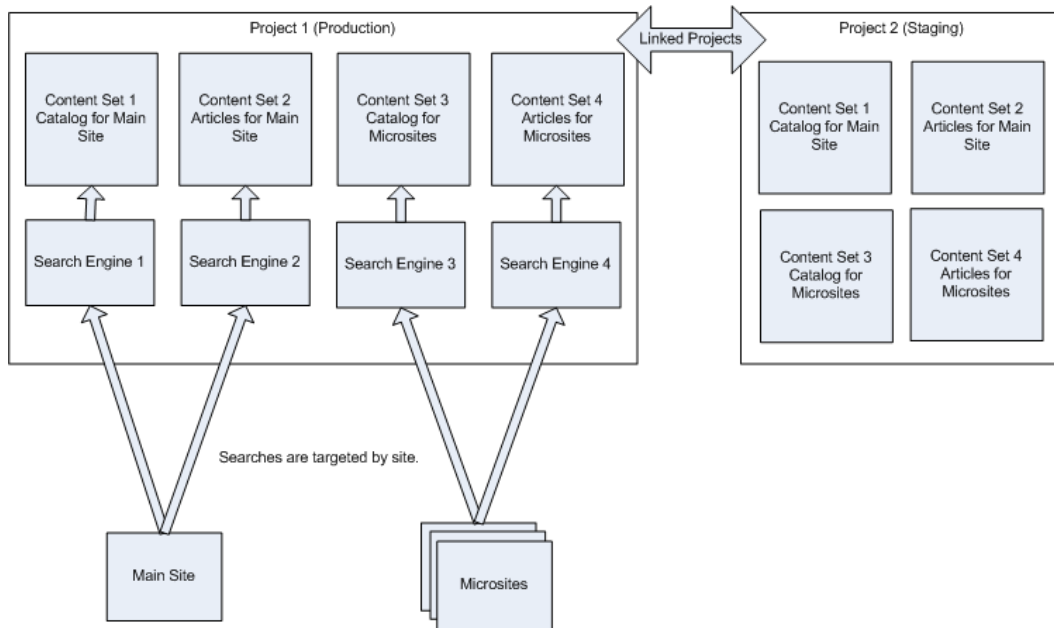
Note that multisite considerations are not the only factor in determining how your projects and content are arranged. For example, you may want to separate out data that you expect to be indexed or searched frequently from data that is updated or searched less often. Hardware considerations may also be a factor.

The simplest multisite arrangement is to have all content in a single content set and a single project, which is searched the same way by all sites. This works best if you have a small amount of content.



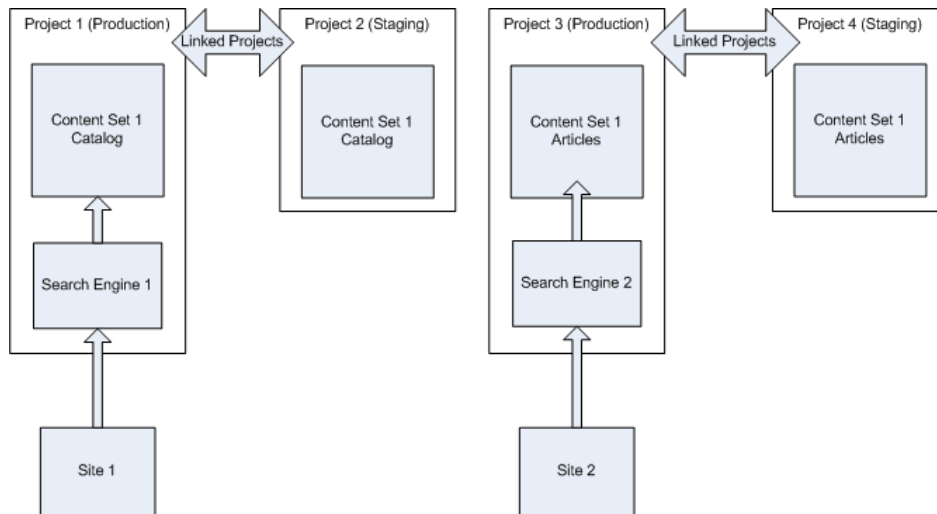
Multisite with One Content Set

A more complicated arrangement separates out your content into content sets, which can be individually searched and updated.



Multisite with Segregated Content

Finally, you can completely wall off content for separate sites in separate projects.



You configure certain ATG Search multisite settings in Site Administration, but the majority of your search project work is still done in Search Administration.

Important: When you are creating sites in Site Administration, in order to link those sites to search content, you must first create projects and content sets in Search Administration. See the [Associating Content with Sites in Site Administration](#) section in this guide for further information.



Accessing Search Administration

To access Search Administration:

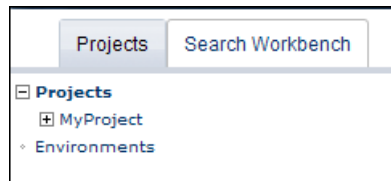
1. On the server running Search Administration, access the ATG Business Control Center. The default location is `http://hostname:portnumber/atg/bcc`.
2. Enter a login and password that includes the Search Access role.
3. In the left navigation area, click Search Administration.
4. Click Browse Search Projects to begin using Search Administration.

Accessing the Search Workbench

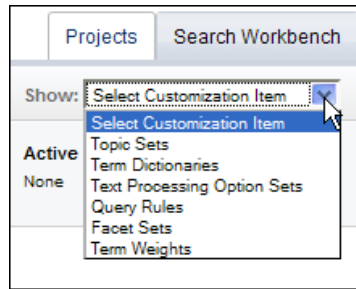
To access Search Administration from the Business Control Center:

1. On the server running Search Administration, access the ATG Business Control Center. The default location is `http://hostname:portnumber/atg/bcc`.
2. Enter a login and password that includes the Search Access role.
3. In the left navigation area, click Search Administration.
4. Click Open Search Workbench.

To access the Workbench from within Search Administration, click the Search Workbench tab in the left navigation area.

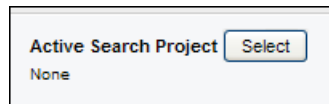


You can then select a customization item to configure in the Workbench.

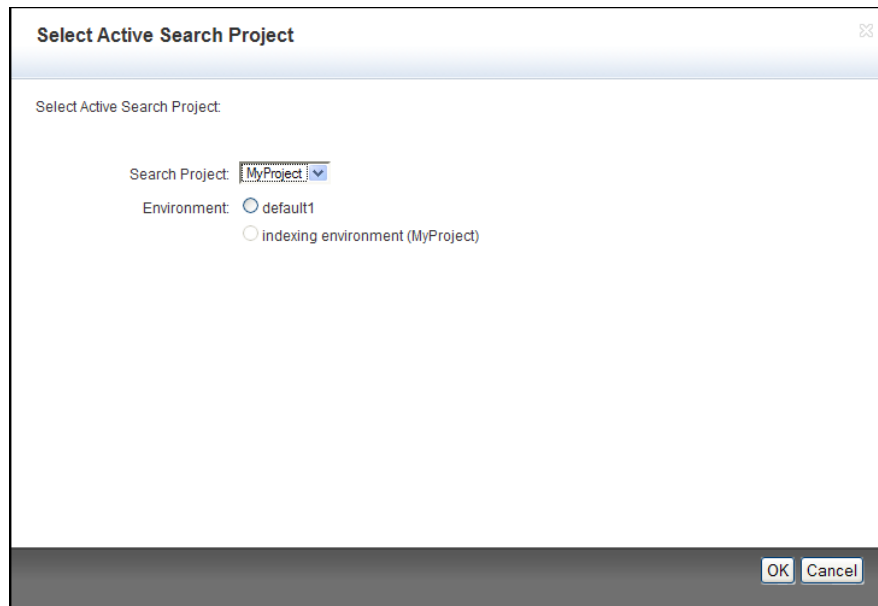


When you select a customization item, the right-hand area displays a page for working with that data type. The left navigation area changes to provide information specific to that customization type.

The left navigation area also includes the Active Search Project display.



To view customization data in the context of a particular search project with which it is associated, use the Select link to choose a project.



Some Workbench tools, such as Dictionary Inspection, require an active project to work on. You can change the active project at any time.





3 Managing Search Projects

A *search project* is the core of ATG Search Administration. In it, you define what content you want to make searchable as a unit, and all of the associated configuration information: where the content is stored, what rules are followed for its indexing, what languages are used in it, and so on.

Note: A search project is not a Content Administration project. Search projects do not contain deployable assets, do not have an associated workflow, etc., but are instead a means to organize and configure a given set of indexable content.

This chapter provides all of the information necessary to create a search project, and add content and customizations.

This chapter contains the following sections:

[Search Project Overview](#)

[Before You Begin—Global Settings](#)

[Creating Search Projects](#)

[Copying and Linking Search Projects](#)

[Viewing Search Projects](#)

[Deleting Search Projects](#)

[Adding Content to Search Projects](#)

[Associating Content with Sites in Site Administration](#)

Search Project Overview

A search project collects the content that you want to index, settings that control how that content is indexed, and what resources are used to create the index and answer end-user queries. The *index* is your searchable content. Each index is composed of one or more content sets. A search project should represent a body of content that you want to deploy as one entity. Each search project automatically has one content set when created, as well as a default environment for creating and deploying indexes.

You can also create projects that do not contain searchable content, but which serve as holders for related information such as custom dictionaries or type-ahead information.

Note: Index size does not bear any direct relationship to the size of the raw information being indexed—dictionaries, topics, and other customizations all add to the size of the index, as can the nature of the



content itself. For example, content consisting mostly of pictures with some metadata might form a very small index relative to the raw content size, while a product catalog with many small, unique pieces of information might be relatively large.

Before You Begin—Global Settings

Global settings allow you to target searches to specific content sets within your search project. You may need to create global settings before you create projects. Global settings consist of two parts, a *target type* and a *content label*.

The target type identifies the general purpose of the search project. If you are using ATG Content Administration, the target types should correspond to your Content Administration targets (Search Administration does not enforce this; it is a good practice, however). ATG Search includes two default target types, Production and Staging.

The content label identifies the purpose of a content set, such as a catalog or a collection of articles. ATG Search includes a single default content label, Catalog. Note that unless you actually need to separate out your content by type in order to perform targeted searches, you won't need additional content labels.

One use of global settings is for ATG's multisite feature. In multisite, global settings allow you to direct searches to content associated with a particular site. Even if you are not using multisite, however, you may find it useful to create global settings beyond the defaults. For example, if your search project includes different types of content that you want to search individually from a single site, you can use content labels to differentiate the content types. If you are using multisite and your content model makes all content searchable from all sites, you may not need more than the default global settings.

The combinations of content label and target type must be unique for each search project. For example, you can have two projects, each with two content sets, using the following combinations:

- Project1: Production/Catalog
- Project1: Production/Articles
- Project2: Staging/Catalog
- Project2: Staging/Articles

If you create a third project that includes content sets with the Catalog content label, you must also create a new target type for that project.

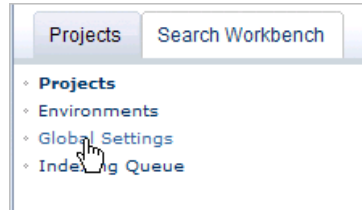
Also note that you can only directly create projects that have a target type of Production; to create all other target types, you must copy a Production project (see [Copying and Linking Search Projects](#) for details). Only projects with the Production target type can have sites associated with them. Site associations are propagated to linked projects.

The target type/content label combination is used to locate specific content sets within a particular environment. For example, the Production/Catalog combination equates to the environment associated with the production project and the content sets that index catalog data. This feature also allows you to use the same form handlers in your production and staging servers without alteration (see the [ATG Search Query Guide](#) for form handler information).



To create global settings:

1. Go to the Projects tab in the left navigation area.
2. Click the Global Settings link.



3. Add Content Labels or Target Types.

Global Settings

Content Labels

Content Labels are used to define the type of content to be indexed within a content set. For example, if catalog and article content are indexed in separate content sets, add an "article" Content. Content Labels are unique within a search project for a given Target Type. The same label may be applied to multiple content sets.

Content Label ?	Associated Content Sets
Catalog	<input type="checkbox"/>
<input type="text"/>	<input type="checkbox"/>

Target Types

Target Types are used to define the type of ATG instance to which an index will be deployed.

Target Type ?	Associated Projects
Production	
Staging	<input type="checkbox"/>
<input type="text"/>	<input type="checkbox"/>

Note: The production Target Type may not be deleted. Other settings may be deleted when there are no associated content sets/projects.

4. Click Save.

Creating Search Projects

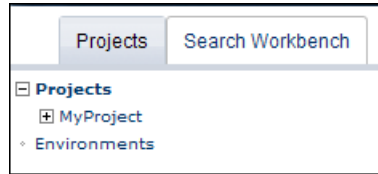
This section describes the overall process of search project creation. For details on creating and configuring a project's components, see the additional sections referred to.

Note: Only Production projects can be created using New Search Project. You cannot change the target type of a Production project. For all other project types, see [Copying and Linking Search Projects](#). Site Administration uses the Production projects to determine what content labels are used.

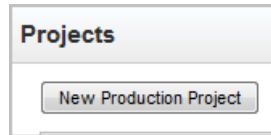


To create a search project:

1. Go to the Projects tab in the left navigation area.
2. Click the Projects link.



3. On the Projects page, click the New Production Project button.



4. In the General section of the New Project page, enter the project's name (required) and description (optional).
5. If the new project does not contain searchable content, check the No Content Items in this Search Project option. An example of a project with no content is a type-ahead project; see [Appendix D: Implementing Type-Ahead in ATG Search](#).
6. Provide a name (required) and description (optional) for the project's first content set, or use the defaults.
7. If more than one content label is available for the first content set, select the Content Label to use; if only one label is available, it is assigned to the content set automatically. See the [Before You Begin—Global Settings](#) section for information on content labels.
8. Click the Create Search Project button.
9. Click the Add Content button to specify content to index in this project. See the [Adding Content to Search Projects](#) section for information.
10. Click the Build Index button to create an initial index from the specified content using the default environment. See the [Building Indexes](#) section for information.

Copying and Linking Search Projects

You may want to duplicate a project if you need to set up another index using similar content and customizations. Linked search projects use the same content sets as the original project. Linked projects are automatically updated for any changes in content set names or labels you make in the Production project. However, note that changes to the content items within the sets are not automatically carried over; this allows you to experiment with changes to your content during testing.



Note: ATG strongly recommends linking if you are setting up a staging environment. Linking ensures that the two projects are configured identically, which reduces the chance of configuration errors.

When you copy a production project that includes indexed repository content, before indexing you **must** do the following:

- Update the name of the IndexingOutputConfig component in the linked project to point to the staging version of the component. The copied version of the project uses your initial IndexingOutputConfig component but appends _COPY to the name.
- Update the host name in the linked project. The copied version of the project uses your initial host name but appends _COPY to the name.
- Remove or update any remote post-indexing customization adapters that refer to the production version.

To copy a search project:

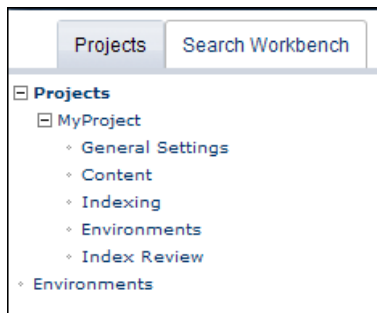
1. On the Projects tab, click Projects.
2. Click the Copy & Link link for the project.
3. Provide a name and description for the project.
4. Select the project’s target type. (See [Before You Begin—Global Settings](#) for target type information.)
5. Click OK.

Customizations associated with your production project are not duplicated, but the same customizations are associated with the new project as with the original.

Viewing Search Projects

To view a search project, do one of the following:

- On the Projects tab, expand the Projects menu, then click the project name.



- On the Projects tab, click the Projects link, then select the project from the list displayed on the Projects page.



Projects			
New Search Project			
Search Project	Description	Target Type	
MyProject		Production	Copy & Link

Deleting Search Projects

When you delete a search project, you can either delete the environments and customization data associated with that project, or detach them from the search project.

Note that you cannot delete a Production project if it has linked projects. You must first delete the linked projects, then the Production project.

1. On the Projects tab, click Projects.
2. Click the project’s delete icon ().
3. In the dialog box, select items you want to delete along with the project.

Note: Any items that you do not delete can be reused by other search projects.
4. Click OK.

Adding Content to Search Projects

Content is the material you want to make available to be searched, and can take many forms. Examples include:

- Sales literature in PDF or Word document form
- A product catalog, including descriptions and prices
- Metadata describing images

Note: For a list of supported formats for file system content, see [Appendix A: Indexable File Types](#). Indexing PDF, HTML, and rich content involves separate installations and licensing; see the [ATG Search Installation and Configuration Guide](#) for information.

ATG Search can index content found in either a file system or an ATG repository. Either type of content can in principle be considered to be either *structured* or *unstructured*. In usual practice, repository data is structured, and file system data is unstructured. A common exception is XHTML documents, which may be stored in a file system but are structured (see the [ATG Search Query Guide](#) for information on the XHTML structure required by ATG Search).



When you index content as structured, the client application through which end-users access ATG Search can constrain searches to specific text fields (in what is called *fielded search*). When using fielded search, a search is performed as usual, but only over the text contained in the specified fields; for example, a user might want to search only across product descriptions in a commerce catalog, or only for symptoms in an ATG Service solution. This differs from constraining by property, which is analogous to “search only items that have brand=BrandX”. (Note that you can achieve a similar effect in unstructured documents using index-only metatags. See the instructions in [Adding File System Content to Content Sets](#).)

Structured content exists in the form of ATG Commerce catalog data and ATG Knowledge solutions. The data loaders and `IndexingOutputConfig` component generate XHTML representations of this data, which ATG Search then indexes. You may want to index file system content as structured if your data already exists as XHTML files. Otherwise, unstructured content usually consists of text documents in various forms.

In the index, unstructured content is rooted at the `/Documents` document set. All structured content is rooted at the `/Solutions` document set. The document set organization allows constraints to be applied at a high level (see the [Topic Sets](#) and [Query Rules](#) chapters for information on features that use the document set, also referred to as *docset*).

Once you have identified the content you want to index, you can add it to your search project’s content set. Each search project automatically has one content set when created. If you plan to have only one content set, go to [Adding File System Content to Content Sets](#) or [Adding Repository Content to Content Sets](#), depending on your content type.

You can use additional content sets to organize your content if you want to update portions of your index independently, associate content sets with specific sites in a multisite configuration, or allow client software that uses ATG Search to constrain end-user searches by content set.

Creating New Content Sets

You can add content sets only to production projects. Content sets you add to a production project are automatically copied to any linked projects.

To add a new content set:

1. On the Projects tab, go to `Projects > Your Project > Content`.
2. Click the Add Content Set button.



Content

[Add Content Set](#)

Content Set 1 [\[Edit \]](#) [\[Add Content \]](#)

Content Label: Catalog

Content Name ^

(None)

[Add Content Set](#)

3. Enter the content set name (required), description (optional). If you have created additional content labels other than the default Catalog label, select a content label.

The content set is added:

Content Set 2 [\[Edit \]](#) [\[Delete \]](#) [\[Add Content \]](#)

Content Label: Catalog

Content Name ^	Location
(None)	

You can now add content to the content set. See [Adding File System Content to Content Sets](#) or [Adding Repository Content to Content Sets](#).

Deleting Content Sets

You can delete content sets only from production projects. Content sets you delete from a production project are automatically deleted from any linked projects.

Note that you cannot delete a content set if sites are associated with that content set through Site Administrations; you must remove the association in Site Administration before deleting the content set.

To delete a content set:

1. On the Projects tab, go to Projects > *Your Project* > Content.
2. Click the delete link next to the content set name.

Content Set 2 [\[Edit \]](#) [\[Delete \]](#) [\[Add Content \]](#)

Content Label: Catalog

3. Click OK to confirm the deletion.

Adding File System Content to Content Sets

To add content from a file system to a content set:



1. On the Projects tab, go to Projects > *Your Project* > Content.
2. Click the Add Content link under the content set to which you want to add the content.



3. On the Add Content Set page, enter a name for the new content.
4. Select File System as the content type. Additional fields appear on the page.
5. Enter or browse to the Content File Path. This is the root directory where the content is located.



6. Under Indexing Options, configure the following optional settings:
 - Select a content-level Text Processing Option Set to apply to this content, if any. The default is to use the project-level set. See the [Text Processing Option Sets](#) chapter for information.
 - Enter a document set name. A document set name creates a sub-path within the final index, which can be used by the client UI to restrict end-user searches.

For example, if there is no Document Set Name, then unstructured documents in this content are indexed under the root /Documents. If you provide a Document Set Name of MyContent, the content is indexed under /Documents/MyContent.
 - Select whether the content should be treated as structured or unstructured during indexing. Examples of structured documents include data in repositories, while unstructured content consists of documents in a file system.
 - If you are using ATG's multisite feature, set the value of the Sites metatag to either All Sites or Selected Sites. All Sites means that unstructured content cannot be constrained by site. Selected Sites allows you to associate your unstructured content with the specified sites.

For example, consider the ATG Commerce Reference Application, which includes three sites: ATG Store US, ATG Store Germany, and ATG Home. If you create a content set called Countries, set the Sites metatag to include US and Germany, and your search request specifies the Germany and Home sites, the search returns index items from the Countries content set. If your search request specifies only the Home site, it does not return index items from the Countries content set.



To select a site, click the Select button. Check a site to select it. Click OK when finished selecting sites.

Note: If you later change the sites associated with your content, you must run a full index to ensure that site associations are correctly represented in search results.

- Identify and create additional index-only metatags as necessary. Index-only metatags allow you to add metatags to unstructured content (structured content is already tagged), allowing users to perform fielded searches. To create a metatag, specify the property's type, then provide a name and a value. Note that in order to use this feature, your content should be organized to reflect the metatags you want to apply. For example, you may have a directory of text files representing news articles. You could further organize them into subdirectories by topic, then apply metatags to allow constraints. In this example, the metatags would be of type String, have name Subject, and values of Local, National, Global, etc.

Check the Create Document Set box if you want ATG Search to create a document set based on this metatag. Document sets allow end-users to search subsets of content and to browse without query input.

- Optionally, for each of the file types supported by ATG Search, you can provide file extensions that should be mapped to that type for this content (optional).



File Type Mappings and File Suppression [Hide]	
Document Type	Default file extensions
Plain Text N/A.	<input type="text" value=".txt"/>
HTML Uses HTML structure.	<input type="text" value=".html .htm .shtml"/>
Rich Content (Includes MS Office Documents, RTF, and hundreds of other formats) INSO/Stellent third party product provides conversion to HTML.	<input type="text" value=".doc .ppt .rtf"/>
Adobe PDF ATG extraction client application using Adobe Library.	<input type="text" value=".pdf"/>
XML XML processing.	<input type="text" value=".xml"/>
XHTML ATG Search-specific XML processing for our published structured content format.	<input type="text" value=".xhtml"/>

8. Configure additional optional file suppression features:

Additional Feature	Default Value
Blank Extension Map files with no extension to one of the six ATG Search file types or to "Suppress"	<input type="text" value="HTML"/>
Unspecified Extensions Map all file extensions not otherwise explicitly mapped to one of the six ATG Search file types or to "Suppress"	<input type="text" value="Suppress"/>
Suppression Map file extensions to "Suppress"	<input type="text"/>
General File Suppression Map files that match wildcard expressions, or explicit file paths, to "Suppress"	<input type="text"/>

- **Blank Extension**—Select a file type to which files with blank extensions should be mapped. If you select Suppress, files with no extension are not indexed.
- **Unspecified Extensions**—Select a file type to which files with unmapped extensions should be mapped. If you select Suppress, files with unknown extensions are not indexed.
- **Suppression**—Identify file extensions that should never be indexed in this content.
- **General file suppression**—Identify specific files in this content to suppress. Separate file name with a space. File names can include asterisks as wildcards. For example: MyDoc* or *Document . doc.

Note that if the filename includes a space, you must enclose that name in double quotes. For example:

MyDoc1 "My Doc2" MyDoc3

9. Configure Advanced Settings (optional) for this content.



Advanced Settings [Hide]

External Access URL:

Converted Document Output Directory:

Converted Document Access URL:

Default Encoding:

Additional Setting:

- External Access URL—Base URL for the document repository folder containing unstructured documents to be indexed.

When ATG Search finds a sentence in the index that answers a question, it sends that sentence to the end user along with a link to the source of the answer. Some document types, such as PDFs, must be retrieved from a URL external to the index in order for the user to view them. This is the URL provided to the end-user viewing the document. For example:

`http://www.somestore.com/product_manuals`

- Converted Document Output Directory—Path to the directory where you want to store image files that are created during the indexing of rich data files.
- Converted Document Access URL—Virtual directory of the Converted Document Output Directory. This allows someone to view images (via a URL link) from answers that are derived from your rich data content.
- Default encoding. This optional value can be used to change the default encoding assumed for content that is not tagged with an encoding, and where the encoding differs from the platform default encoding.

For example, the platform default encoding on Windows is Windows-1252. You may have documents encoded as Shift_JIS, which is used for Japanese, but these documents are not tagged with their encoding. Specify Shift_JIS in this field, and the documents will be decoded properly.

Note: If a document is properly tagged with an encoding, ATG Search uses that encoding, ignoring both the system default and the Default Encoding value for this field in the content definition.

- The Additional Setting area is for any other property=value settings you may need to apply.

10. Click Add Content.

Repeat for any linked projects to which you want to add the content.



Adding Repository Content to Content Sets

To add content from an ATG repository to a content set:

1. On the Projects tab, go to Projects > *Your Project* > Content.
2. Click the Add Content link for the content set to which you want to add the content.
3. On the Add Content Set page, enter a name for the new content.
4. Select Repository as the content type.
5. Provide the following information:

The screenshot shows a form with the following fields and options:

- ★ IndexingOutputConfig Path:
- Location: Local, Remote
- ★ Host Machine:
- ★ RMI Port:

- IndexingOutputConfig Path—Nucleus path to this component, which is used for converting repository items to XHTML for indexing. See [Indexing Repository Data](#) for information.
 - Whether the IndexingOutputConfig component is running locally or on a remote server.
 - If remote, provide the host name and the port used to access IndexingOutputConfig. This should be the RMI port used by the remote ATG installation, which is typically 8860.
6. Under Indexing Options, set the following optional configurations:
 - Select a Text Processing Option Set to apply, if any. See the [Text Processing Option Sets](#) chapter for information.
 - Enter a document set name. A document set name creates a sub-path within the final index, which can be used by the client UI to restrict end-user searches.

For example, if there is no Document Set Name, then structured documents in this content are indexed under the root /Sol uti ons. If you provide a Document Set Name MyContent, the content is indexed under /Sol uti ons/MyContent.
 - Select whether this content should be treated as structured or unstructured during indexing. Examples of structured documents include databases and repositories, while unstructured content consists of documents in a file system.
 7. Click Add Content.

Repeat for any linked projects to which you want to add the content.



Viewing Content

To view content used in a specific search project, on the Projects tab, go to Projects > *Your Project* > Content.

This page allows you to view content as it is organized into content sets, to filter by content type, or to move content from one content set to another within the search project.

Adding Customizations to Content

Customizations are collections of accessory data that can enhance out of the box ATG Search behavior, depending on the nature of your content and the needs of your users.

To add customization data created in the Workbench to a project:

1. Do one of the following:
 - On the Projects tab, go to Projects and click on *Your Project Name*. In the Content for Indexing section of the project page, click the Manage Content link.



- On the Projects tab, go to *Your Project Name* > Content.
2. Customizations are divided into two sets.
 - Language Customizations (Pre-Index) include Core Language Support, which identifies which languages your content uses, and which dictionaries ATG Search requires to index the content; language selection; [Dictionaries](#); Term Weights (see [Understanding Term Weights](#) in the [Dictionaries](#) chapter); and [Text Processing Option Sets](#). These are applied to your content before indexing.
 - Post-Index Customizations include [Topic Sets](#), [Query Rules](#), [Facet Sets](#), search configurations you create in ATG Merchandising (see the [ATG Merchandising Guide for Business Users](#)), and auxiliary data.

To add a customization, select the customization type from the appropriate Add Item list.

3. Check the specific customizations you want to associate with the project.
4. Click Save to save your selections and return to the Content page.

Repeat for any linked projects to which you want to add the customization.

Removing Customizations from Content

To remove customization data created in the Workbench from a project:

1. Do one of the following:



- On the Projects tab, go to Projects and click on *Your Project Name*. In the Content for Indexing section of the project page, click the Manage Content link.



- On the Projects tab, go to *Your Project Name* > Content.

2. Click the Remove icon (🗑️) for the customization.

Repeat for any linked projects from which you want to delete the customization.

Note that removing a customization from a project does not delete the customization, but disassociates it from the project. The customization can still be used with other projects.

Adding Language Support and Dictionary Adaptors

To index content in languages other than English, or to add dictionary adaptors for English:

1. Do one of the following:
 - On the Projects tab, go to Projects and click on *Your Project Name*. In the Content for Indexing section of the project page, click the Manage Content link.



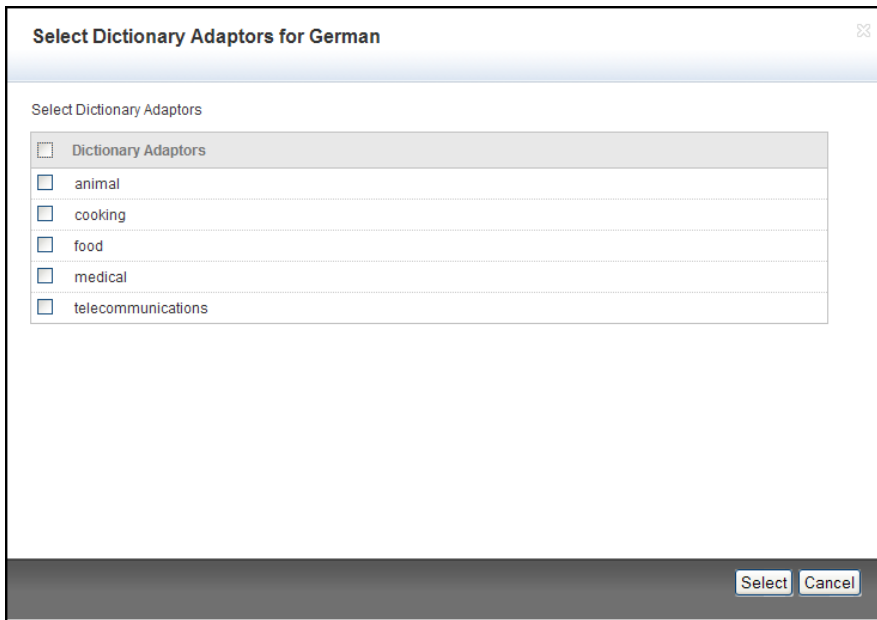
- On the Projects tab, go to *Your Project Name* > Content.

2. In the Language Customizations section, select Core Language Support from the Add Item drop-down.
3. Check the language(s) you want to enable indexing for. ATG Search supports the following languages:
 - Catalan
 - Chinese (traditional and simple)
 - Dutch
 - English
 - English (UK)
 - French
 - French (Canada)
 - German
 - Hungarian
 - Italian



- Japanese
 - Korean
 - Polish
 - Portuguese
 - Portuguese (Brazil)
 - Russian
 - Spanish
4. Each selected language (including English, which is selected by default) has a Dictionary Adaptors link. Dictionary adaptors contain industry-specific terms that are not part of the main ATG Search dictionary. Click the link to select dictionary adaptors that should be included for the language. Different adaptors are available for different languages; the example that follows is for German.

Check any adaptors you want to include, and click Select.



5. In the main window, click Save.

Note: If you have no English-language content in your project, you must create a custom Text Processing Option Set that indicates the languages you are using and associate it with your project. The default Text Processing Option Set assumes English as the content language.



Associating Content with Sites in Site Administration

In a multisite environment, you associate specific sites with specific content sets as a way of constraining indexing and querying. For example, suppose you have three sites, A, B, and C, and a content set that you want to make available for searching on sites A and C, but not on site B. To do this, you could associate sites A and C with this content set in ATG Site Administration. When the index is generated, the logical partition associated with this content set does not include data for site B.

When a query is issued, ATG Search looks at the sites the query applies to, determines which content sets are associated with those sites, and directs the query only to those content sets. So in this example, if a customer is searching site B, the A-C content set is not searched, because that content set is not associated with site B. This makes searching more efficient, and allows you to create user interfaces that let customers search specific sites.

Note that if a disabled site is associated with a content set, ATG Search ignores this association during querying.

To ensure that your site data is correctly indexed and available when your site goes live, use the following procedure:

1. Create your site and catalog data.
2. Publish the new site in a disabled state.
3. Index the site data.
4. Republish the site in an enabled state.

For each content label you have created in Search Administration, in Site Administration you can select a content set that has that label to be used for that site.

To associate a site with a content set:

1. On the server running Search Administration, access the ATG Business Control Center. The default location is `http://hostname:portnumber/atg/bcc`.
2. Enter your login and password.
3. In the Operations menu, click Site Administration, then Manage Site Assets.
4. Enter a name and description for the project. The name is used to identify the project in various places in the UI, for example in the To Do list on the Home page. Making this name distinctive is recommended so that you and other users can distinguish it from other projects in your system. Project names do not have to be unique, however. The description is optional but can be helpful where many projects have similar names.
5. Click Continue when you have finished. At this point, the project's Author task becomes active.
6. Select an existing site to configure, or click the Create New icon on the Actions menu, then select Site.
7. Select a site template.



8. The Search Content table shows the content labels you created in Global Settings. Select a content set to associate with each label. If you do not want a content label to be available on the site you are configuring, you can leave the content set blank.

Site	
Search Content	
Content Label	Content Set
Catalog	Catalog Content Set

Note that any time you change a site's content set associations in Site Administration, you'll need to reindex the affected content sets. This is because the site information can be used to pick the right content set.



4 Working with Indexes in Search Administration

Building the index turns your search project's content into searchable data, associates all of the relevant customizations with the index, and makes it available for searching.

This chapter includes the following topics:

[Estimating Index Size](#)

[Applying Customization Data](#)

[Building Indexes](#)

[Creating Automatic Indexing Rules](#)

[Using Index Review](#)

[Viewing Indexing Histories](#)

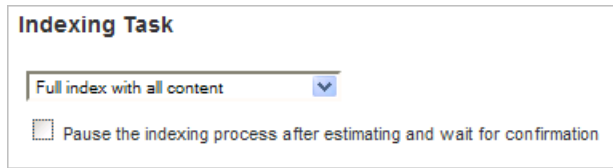
[Deleting Indexes and Histories](#)

[Redeploying Old Indexes](#)

Estimating Index Size

To estimate the size of your index before building it:

1. On the Projects tab, go to Projects > *Your Project*.
2. Click the Manage Indexing link.
3. On the Indexing Setup tab, check the Pause the indexing process after estimating option.



Note: You may need to wait a moment for the page to refresh before seeing estimation begin. Depending on your content, estimation can take a significant length of time.

4. Click the Build Index button.
5. When estimation is complete, review the data to ensure that you have sufficient resources to index and deploy the content. The estimation results include:
 - Estimated size of the index in MB
 - Approximation of how long it will take to build the index
 - A partitioning plan. A *physical partition* represents the amount of content that can be served by one search engine, approximately 1.5 GB. You may want to split your content into more than the minimum number of partitions if the estimation is close to the limit.

The deployment plan displays the search engines available for these partitions. See the [ATG Search Installation and Configuration Guide](#) for information about adding search engines.

You may also want to review any errors that occurred during index estimation (see below for details).

6. To continue and create the index, click Continue Indexing. To cancel the process, click Cancel Indexing, then Yes to confirm.

Indexing errors can occur due to problems with the content (unsupported file type, corrupted or otherwise unreadable file, etc.) or with the customization data (duplicate definitions in a dictionary, references to content items not in the index, invalid terms in a term weight file, etc.). If you encounter customization data errors, you may be able to address them by editing your customization data.

Some examples of customization data error messages are:

```
-- Rank config missing pattern:
Content Item: cust-100002: Rank
Config: /atg/searchadmin/adapters/customizations/RankConfigVFS, rankConfig_
1174904401681.xml
Errors for Merchandising Search Configuration
"D:\p4\cascade\platform\dynamo\home\localconfig\atg\searchadmin\customizations\
rank_configs\rankConfig_1174904401681.xml "
Identifier      Type      Description      Other Info
cfg7002_rule21002_0      error      Failed to extract pattern      path:
/Ranking/cfg7002/rule21002
```



```
-- Facet Set (refinements) missing range information:
Content Item: cust-100008: Refinement
Configuration: /atg/searchadmin/adapters/customization/RefinementVFS, facet_dsw.xml
Errors for Facet Set
"D: \p4\cascade\platform\Dynamo\home\local config\atg\searchadmin\customizations\
refinements\facet_dsw.xml "
Identifier      Type      Description      Other Info
300004          element   Missing explicit ranges
300006          element   Missing explicit ranges
300012          element   Missing explicit ranges
```

Applying Customization Data

Some customization data types can only be applied when you fully index content from scratch, and any changes to them require reindexing. These are known as “pre-index customizations,” and include the following types:

- Dictionaries
- Term Weight Sets
- Text Processing Option Sets

Note that the text processing option sets are not loaded themselves, but direct the indexing of other content.

Other customization data types (“post-index customizations”) can be applied after the content is indexed, and do not require that you reindex the content to apply changes, though you do need to update the index with the new customization data. The following types of customization data can be changed without reindexing your content:

- Query Rules
- Topic Sets
- Facet Sets
- Auxiliary Data
- Merchandising Search Configurations

Building Indexes

This section describes the steps involved in creating a search index.

More than one content set can be indexed at the same time, if enough indexing engines are available (see the [ATG Search Installation and Configuration Guide](#) for architecture recommendations), however, the indexes must be for different search projects. Additional indexing requests that involve the same content



are queued until the content is available. You can access the queue from the Status Monitor tab during indexing, as shown:



When indexing is complete, the resulting indexes can be automatically copied to search environments. If sufficient engines are not available, indexing tasks are placed in a queue and processed in the order in which they are submitted.

Note: If you later change the sites associated with your content, you must run a full index to ensure that site associations are correctly represented in search results.

To build an index using the project’s default settings:

1. On the Projects tab, go to Projects > *Your Project*.
2. Click the Manage Indexing link.
3. Click the Build Index button.

The window changes to the Status Monitor tab.

4. If you have selected to stop indexing after estimation, when the estimation is finished, click Continue Indexing.

To build an index using settings other than the project defaults:

1. On the Projects tab, go to Projects > *Your Project* > Indexing > Index Setup tab.
2. Click the Manage Indexing link.
3. Select the Indexing Task.

Each task can provide additional options depending on your selections. The following example shows the Full Index with All Content task.



Indexing

Index Setup
Automatic Indexing Rules
Status Monitor
History

Indexing Task

Full index with all content

▼

Pause the indexing process after estimating and wait for confirmation

1. Index to be used
Use clean index

2. Pre-Index Customizations
Load latest pre-index customizations

3. Content
Index all content

4. Post-Index Customizations
Load latest post-index customizations

5. Deployment

Don't Deploy

Deploy index to selected environments

<input checked="" type="checkbox"/> Environment ^	Primary
<input checked="" type="checkbox"/> default1	<input checked="" type="radio"/>

- Full index with all content—Builds your entire index from scratch, using the content sets you select and the most up-to-date customization data, then deploys to the environments you select.
- Full index with selected content—Specify particular content sets to be reindexed.
- Incremental index—Only indexes new or modified content in an existing index. You can select whether or not to update all content sets, and whether to apply post-indexing customization data.

Note: Incremental indexes involve comparing old to new content to determine changes, and can take longer to build than a full index. You need to do at least one full index before using incremental indexing.

Note: If you try to do an incremental index of selected content sets when you have changed a pre-index customization that applies to that index, such as adding a core language, ATG Search performs a full reindex, not an incremental



index. This prevents you from having out of date customization data for the non-selected content sets.

Similarly, if you have configured the `maxIncrementalChanges` property of your `ProductCatalogOutputConfig` component (this property is rarely used), and if the number of updates exceeds the configured limit, a full index is performed.

- **Load Post-Index Customizations Only**—Updates the index by reapplying post-index customization data (such as topics) but makes no other changes. You can either apply all post-index customizations, or specify customizations to apply.
 - **Deploy Only**—Deploys your existing index to the selected environments without making any changes.
4. Select content to index. Some indexing tasks allow you to specify which content sets are indexed.

3. Content

Index selected content sets

<input type="checkbox"/> Content Set	Content	Location
<input type="checkbox"/> Content Set 1	MyContent	File System

Click the checkboxes to select those content sets you want to index.

5. Select customizations to update. The indexing task you select may allow you to choose whether to apply all customizations, or only selected post-indexing customizations.

Note: If you are using Search Merchandising facets, make sure to add your Facet Sets and Search Configurations to the index as post-indexing customizations.

4. Post-Index Customizations

Load all customizations

Load selected post-index customizations

Content/Data Source

(None)

6. Select deployment environments. Some indexing options allow you to deploy the indexed content when indexing is concluded. Check the environments to which you want to deploy the index.
7. Click **Build Index**. The status monitor shows the index build progress. After indexing, the page shows success/failure metrics for each step, and provides access to any error messages logged.



Note: The Time Remaining estimate reflects only index creation. File deployment may take additional time.

Note: You can build indexes for separate projects in parallel. You can run multiple indexing tasks for a single project in parallel only if the updates do not involve the same content sets. If the same content sets are involved, indexing tasks are queued.

Creating Automatic Indexing Rules

Any set of indexing configuration choices you make can be saved as an auto-indexing rule for the project.

You can also use automatic indexing rules to trigger index updates every time you deploy content using Content Administration.

Warning: Make sure you set the schedules for your automatic indexing rules so that a given project does not try to run two indexing rules at the same time; if it does, unpredictable system behavior may result. If scheduled tasks overlap, ATG Search will finish the first indexing task before starting the second, which could mean long delays in updating your index. Two *different* projects can be indexed concurrently, however, if you have enough indexing resources.

To create an Auto-Indexing Rule:

1. Make your indexing configuration selections as described in [Building Indexes](#).
2. Click Save as Auto-Indexing Rule. The page switches to the Automatic Indexing Rules tab. An example follows.

Indexing task

Status: Enable Disable

Indexing Task Name:

Comment:

Steps:

- Use existing index
- Use existing pre-index customizations
- Do not index content*
- Load latest post-index customizations
- Index the processed content and customizations
- Deploy index to selected Environments**

Content Sets: (None)

Deploy to: staging_MyProject

3. Select Enable or Disable.
4. Provide an Indexing Task Name (required).
5. Add a descriptive comment (optional).



6. Select an indexing trigger.

- CA Deployment—The indexing process runs every time the specified target receives a Content Administration deployment. The target name you provide **must** be exactly the same as it appears in Content Administration.
- Schedule—Set up a schedule on which the indexing rule should run.

- Request—Use this setting for integration with ATG Knowledge.

7. Click Create Rule.

Using Index Review

The Index Review tabs allow you to examine your index after it has been created. They provide statistics about the index as a whole and about individual content items. You can use this information for troubleshooting; for example, if a search does not return the expected results, has the index size changed? Are all expected languages and file formats included? Has the customization data changed? Are the expected content items included in the index? This information helps locate and identify indexing problems before they can affect your production site.



Content Summary

The content summary tab provides high-level information about your index and the content items it includes. You must have created at least one index before this page can be used.

To view the Content Summary:

1. On the Projects tab, go to Projects > *Your Project* > Index Review. Content Summary is the default tab.

Current Index

Indexing Task: Full index with all content	Start Time: 06/16/2010 2:14 PM
Indexing Status: Finished with Error(s)	End Time: 06/16/2010 2:15 PM
Errors: 79 [View Indexing Log]	Deployed to: default1

	Content Set 1	Total
Overview	View Details	
Content Configured	1	1
Size of Indexable Content	71.6 MB	71.6 MB
Index Size	12 MB	12 MB
Percentage of Max Size	0%	
Index Scheme	--	compressed
Content Items		
Total Items	1	1
Structured	0	0
Unstructured	1	1
In English	1	1

2. Click View Details for additional information on a particular content set.

Index Audit

The Index Audit page allows you to compare metrics from your current index with a previous version of the index. You can display only changed metrics, or all metrics in the audit. By default, the audit compares the current index to the previous index, and displays only the metrics that have changed.

You must have completed and deployed at least one index before you can use this tab, and two indexes before you can do a comparison.

To use Index Audit:

1. On the Projects tab, go to Projects > *Your Project* > Index Review.
2. Click the Index Audit tab.



3. Click the Select link to choose an index to compare to the current index. By default, the next most recent index is selected for comparison. If you only have one index, metrics for that index are displayed, but there is no comparison.
4. By default, only changed metrics are displayed. Click the All Elements button to show all metrics for both indexes, or click Changes Only to view only metrics that differ.

Note: The *Topics loaded* metric does not include default topics supplied in Search Administration, but only topics you explicitly add.

Content Inspection

Content Inspection allows you to view individual content items in your index, along with information about that item, such as associated metadata or text properties, how the item would be displayed on a site, and the XHTML representation (for structured data only). Note that this feature does *not* replicate how a search would be performed by a user on your site (for information on how you can do this, see the [ATG Merchandising Guide for Business Users](#)).

To view items, enter one or more search terms and apply constraints as needed. For example, if you want to view catalog content items that include the Widgetizer product, but do not want to see text documents that mention the product, you could constrain your search to include only structured documents.

To use Content Inspection:

1. On the Projects tab, go to Projects > *Your Project* > Index Review.
2. Click the Content Inspection tab.
3. Enter one or more constraints for your search. Click the More Options link to display more constraints. If you enter no constraints, you can browse all content.

You can uncheck any Enable Constraints box to disable a particular constraint; this allows you to see your search results with and without that constraint without having to re-enter the information later.

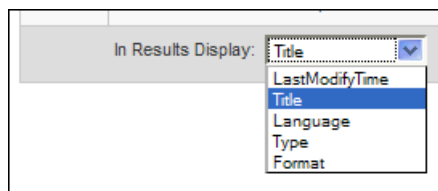
The constraints include:

- Term—Enter a single word for which to inspect content. To add more terms, click the + icon. To remove terms, click the – icon. Uncheck the All Forms box to search only for literal matches on the term you enter.

- Language—Constrain your inspection to documents in the selected language.
- Property—Select a metadata property and enter a value to constrain results to content items with that value for the property. You can use this constraint to filter by site if you are using ATG’s multisite feature.



- Unstructured—Constrain your inspection to either structured or unstructured documents. By default, both formats are included.
 - Document Set—Constrain your results to a particular document set, such as Topics/National or Topics/si tel d/News.
 - URL—Enter an exact document URL. Since this constraint returns only one document, it is generally not used in combination with other constraints.
 - Document Format—Constrain your inspection to HTML documents, text documents, etc.
4. Select a display option for the second results column. This property is included in the results along with the item URL. The default is Title.



5. Click Search. Ten results at a time are displayed as a table.

Items: 1-10 of 3,349

URL	Title
...lacoadmin\ACQAdminDevGuide.1.1.html	atg
...lacoadmin\ACQAdminDevGuide.1.24.html	Running ATG Outreach with ATG Commerce
...lacoadmin\ACQAdminDevGuide.1.6.html	Overview of ATG Outreach Profile Repositories
...lacoadmin\ACQAdminDevGuide.1.60.html	Opt-in Forms
...lacobusiness\ACOBUSUserGuide.1.15.html	Segments
...lacobusiness\ACOBUSUserGuide.1.46.html	Adding Child Stages
...lacobusiness\ACOBUSUserGuide.1.62.html	Browsing Campaign Templates
...lacobusiness\ACOBUSUserGuide.1.9.html	ATG Outreach Overview
...datawarehouse\ACDataWarehouse.1.73.html	KDW_CONTROL
...reportinginstall\ACInstall.1.34.html	Running ATG Customer Intelligence Modules

Items: 1-10 of 3,349

6. Click a result URL to view information about that content item.



Basics	Metadata	Text Properties	Preview
Basics			
Property	Value		
ID:	18		
URL:	C:\Docs\Outreach20071docs\acobusiness\ACOBUSUserGuide.1.9.html		
Title:	ATG Outreach Overview		
Summary:	2 ATG Outreach Overview This chapter introduces you to important terms used in ATG Outreach and ATG Content Administration and includes an example of a simple campaign. ATG Outreach Overview		
Type:	Unstructured		
Format:	HTML		
Language:	English		
Physical directory:	/Documents/Outreach20071docs/acobusiness		

The popup includes the following tabs:

- Basics—General information about the content item.
- Metadata—If you have any indexed metadata for this content item, the tab displays the metadata.
- Text Properties—If you have configured any text properties for this content item, the tab displays the properties and their values.
- Preview—Shows how the content item might appear in a Web page.
- Structured Source (not shown)—This tab is shown only for structured content. It displays the XHTML into which the content is converted.

IndexingOutputConfig Analysis

The IndexingOutputConfig component is used to generate XHTML documents from ATG repository items (see the [Indexing Repository Data](#) chapter for detailed information). If your project includes repository data, you can use this page to identify properties you are indexing that may not be adding value to your index. Indexing unnecessary properties increases the size of your index and slows response times. This page can provide guidance on ways you might improve structured data processing and therefore ATG Search performance.

Note, however, that the data represented on this page does not indicate errors, but consists of informational feedback. Your application may have unusual needs, for which these recommendations do not apply. The following image shows an example of information that may be included in this tab:



Index Review

Content Summary
Index Audit
Content Inspection
Index Output Config Analysis

Current Index: 06/07/2010 3:24 PM (Full index with all content)
 Environment: commerce
 Total number of content items: 632
 Content items analyzed: 632
 Number of properties in analyzed items: 21972
 Total size of text in analyzed items: 279370

Per Item Analysis (meta/text) [Hide 1 Checks] ?

Type	Description	Count	Average	Name	Value
meta	average properties per item	632	34.77	--	--

Properties with a Single Value (meta) [Hide 4 Checks] ?

Type	Description	Count	Average	Name	Value
meta	single string value	632	1.0	\$repository.repositoryName	ProductCatalog
meta	single string value	632	1.0	\$itemDescriptor.itemDescriptorName	product
meta	single integer value	632	1.0	LastModifyTime	0
meta	single integer value	632	1.0	endDate	0

Text Fields with Small Contents (text) [Show 11 Checks] ?

Text Fields with Multiple Instances (text) [Show 6 Checks] ?

The analysis page performs the following checks. See the [Determining Which Properties to Index](#) section for guidance on applying this information to your IndexingOutputConfig settings.

- Per Item Analysis (meta/text)—Number of meta-properties indexed per content item.
- Properties with a Single Value (meta)—Meta-properties that have the same value for all content items with that property.
- Properties with Long String Values (meta)—Identifies meta-properties that contain strings more than 50 characters long on average.
- Text Fields with only Numbers (text)—Identifies text properties that contain only numbers. Such fields can be useful to index in the case of part numbers and other likely search terms, but are likely to be less useful in the case of dates or Boolean values.
- Text Fields with Small Contents (text)—Identifies text properties that contain fewer than 10 characters on average.
- Text Fields with Multiple Instances (text)—Identifies text fields that occur more than once per content item; depending on the contents, it may be possible and useful to combine such fields into one field.



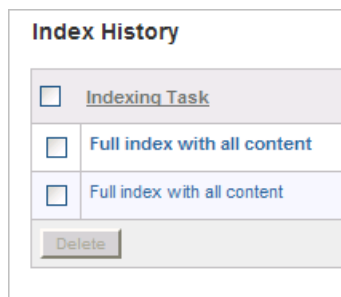
Viewing Indexing Histories

The History tab provides a record of previous index builds. It allows you to review the detailed log for a particular build, to deploy the index produced, or to delete the index and event record of the build.

To view indexing history, in the Projects tab go to Projects > *Your Project* > Indexing > History tab.

For each item in the history, the table shows the Indexing Task (Full, Incremental, Custom, etc.), start and end times, status, number of errors, number of content items, size of the index file, environments to which the index was deployed, and whether it is currently deployed to a search environment.

Click the Indexing Task column entry to view details about the content indexed.



Deleting Indexes and Histories

To conserve disk space, you can delete indexes that are not currently active. The History page allows you to delete the index and event record of the task. The page also features options for automating the deletion of previous indexes.

Deleting Index Histories Manually

To manually delete an index history:

1. On the Projects tab, go to Projects > *Your Project* > Indexing > History tab.
2. Click the delete icon (✖) for the index to delete.
3. Click OK to confirm the deletion.

If the index file itself has not been deleted, you can either keep the event history but delete the index file, or delete both.

If the index files have already been deleted or the indexing task was cancelled, but the event history entry has not been deleted, then the delete applies only to the event history.



Configuring Index Cleanup Rules

If used, the automatic index cleanup rules are run once per day. To configure automatic history deletion:

1. On the Projects tab, go to Projects > *Your Project* > Indexing > History tab.

2. Set the following:
 - Minimum inactive indexes—You must have at least the specified number of inactive indexes before any index is automatically deleted.

Note: Active indexes are never automatically deleted.
 - X most recently indexed—Retain this number from the most recently created indexes.
 - Indexed in last X days—Retain indexes created after the specified number of days in the past.

If neither the “X most recently indexed” nor the “Indexed in last X days” rule is selected, no indexes are deleted.

3. Click the Save Rules button.

When indexes are automatically removed, both the index files and the related event histories are deleted.

An index will not be deleted if an existing indexing task has been scheduled that has a reference to that index, for instance an incremental update via automatic indexing rule.


By default, cleanup rules are run 30 minutes after ATG Search starts up, and every 24 hours after that. You can configure this behavior in the schedule property of the `/atg/searchadmin/repository/service/IndexCleanupService` Nucleus component in the ATG Control Center. You can also run the cleanup manually at any time using the ATG Dynamo Server Admin. See the [ATG Installation and Configuration Guide](#) for information on both the ACC and the ATG Dynamo Server Admin. Go to `/atg/searchadmin/repository/service/IndexCleanupService`, and invoke the `cleanIndexes` method.

Redeploying Old Indexes

If a problem has been found in an active index, or you want to diagnose issues, you can use the History tab to redeploy an old index.

To deploy an old index:



1. On the Projects tab, go to Projects > *Your Project* > Indexing > History tab.
2. Click the Redeploy icon () for the index.
3. Select the environment to which to deploy the index.
4. Click OK to confirm.

Note: The next index you perform must be full. You cannot index incrementally from a redeployed index.



5 Indexing Repository Data

To make repository data, such as your product catalog, available for searching, the ATG platform must transform the data into a format that ATG Search can index, and then load this data into ATG Search for indexing. This chapter describes the ATG classes and components that perform these functions, and explains how to configure or override these classes and components to work with the repositories in your ATG environment.

This chapter contains the following sections:

[Creating XHTML Documents from Repository Items](#)

[Specifying Properties for Indexing](#)

[Loading the XHTML Documents into ATG Search](#)

Creating XHTML Documents from Repository Items

To make repository items searchable, the ATG platform creates textual representations of these items. These textual representations are XHTML documents whose tags contain the values of the repository item's properties. The XHTML is written out in a form that allows it to be efficiently streamed for indexing.

The XHTML documents also encode the parent and child relationships of the items. For example, an XHTML document that represents an ATG Commerce product can include information about its parent category's properties, as well as information about the properties of the child SKUs. This makes it possible to search category and SKU properties as well as product properties.

To specify how to transform repository items to their XHTML representations, you configure a globally scoped Nucleus component of class `atg.repository.search.indexing.IndexingOutputConfig`. (An `IndexingOutputConfig` component applies to only one repository, so for each repository you want to index, you must create one or more separate components.) The `IndexingOutputConfig` class has a number of properties that configure various aspects of the XHTML document creation and submission process. This section describes some of the key aspects of configuring an `IndexingOutputConfig` component.

Note that the DCS Search module includes an `IndexingOutputConfig` component named `/atg/commerce/search/ProductCatalogOutputConfig` that is preconfigured to work with a typical catalog repository. See [Indexing ATG Commerce Catalog Data](#) for more information.



Specifying the Definition File and the Repository

When you create an `IndexingOutputConfig` component, you set the following properties:

definitionFile

The full Nucleus pathname of the XML indexing definition file that configures the repository item types and properties to include in the XHTML documents. See [Definition File Format](#) for information about how to create the definition file.

repository

The full Nucleus pathname of the repository that the definition file applies to.

For example, the `ProductCatalogOutputConfig` component has these settings:

```
definitionFile=/atg/commerce/search/product-catalog-output-config.xml
repository=/atg/commerce/catalog/ProductCatalog
```

It is also possible to specify the repository in the indexing definition file using the `repository-path` attribute of the top-level `item` element. If the repository is specified in the definition file and also set by the component's `repository` property, the value set by the `repository` property overrides the value set in the definition file.

Note that in an ATG Content Administration environment, the repository should *not* be set to a versioned repository. Instead, it should be set to the corresponding unversioned target repository. For example, an `IndexingOutputConfig` component for a product catalog in a CA environment could be set to:

```
repository=/atg/commerce/catalog/ProductCatalog_production
```

For more information about specifying repositories in a CA environment, see the [ATG Search Installation and Configuration Guide](#).

Definition File Format

An indexing definition file begins with a top-level `item` element that specifies the item descriptor to use, and then lists the properties of that item type to include. The properties appear as `property` elements within `text-properties` or `meta-properties` elements. The values of properties included in `text-properties` elements are indexed as searchable text, while the values of properties included in `meta-properties` elements can be used in constraints or facets.

For example, in a product catalog, you'd typically want a property containing a product description to be searchable, but you'd typically want a `price` property to be usable as a constraint (e.g., to constrain the results to include only those products whose price is less than \$100). If you want a property to be both searchable and usable as a constraint, it must appear within both the `text-properties` and `meta-properties` elements. For example, you might use a `manufacturer` property this way, enabling searching for a specific manufacturer, or constraining the results to exclude items from a certain manufacturer.

The top-level `item` element in the definition file can contain child `item` elements for properties that refer to other repository items (or arrays, Collections, or Maps of repository items). Those child `item` elements in turn can contain `property` and `item` elements themselves.



For example, the following definition file enables indexing of users and their home addresses:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE item PUBLIC "-//Art Technology Group, Inc.//DTD Repository Output
  Specifier 1.0//EN" "http://www.atg.com/dtds/search/indexing-dependency-
  schema.dtd">

<i tem i tem-descri ptor-name="user" i s-document="true">
  <ti tle property-name="l astName"/>
  <meta-properti es>
    <property name="dateOfBi rth" type="date"/>
  </meta-properti es>
  <text-properti es>
    <property name="fi rstName"/>
    <property name="l astName"/>
  </text-properti es>
  <i tem property-name="homeAddress">
    <text-properti es>
      <property name="address1"/>
      <property name="address2"/>
      <property name="ci ty"/>
      <property name="state"/>
      <property name="postal Code"/>
      <property name="phoneNumber"/>
    </text-properti es>
  </i tem>
</i tem>
```

Note that in the above example, the top-level `item` element has the `is-document` attribute set to `true`. This attribute specifies that an XHTML document should be generated for each item of that type (in this case, each user item). If, instead, you want to generate a separate XHTML document per `homeAddress` item, you set `is-document` to `true` for the `homeAddress` item type and not for the user item type. In that case, the parent properties (`firstName` and `lastName` in the above example) are repeated in each XHTML document.

XHTML Document Format

In an XHTML document that the ATG platform generates from a repository item, metadata properties are represented by `meta` tags in the head of the document, while text properties are represented by `div` tags in the body of the document. The ATG platform generates `meta` and `div` tags using values from the repository item's properties, and uses a URL of the following form to uniquely identify each document:

```
atgrep: / repository-name/ i tem-descri ptor-name/ repository-id
```

`meta` and `div` properties are output with the following formats:



```
<meta name="atg: type: property-name" content="property-value">

<div class="atg: role: property-name" id="ID">
```

The XHTML documents name the properties using a simplified bean property notation. For example, a document generated from a user profile might have `firstName` and `lastName` properties corresponding to the equivalent properties of the user repository item type. The document properties corresponding to the `city` and `postalCode` properties of the `homeAddress` child repository item would be `homeAddress.city` and `homeAddress.postalCode`.

Multi-value properties are given names without array subscripts, as are the property names of multi-value repository item properties. For example, suppose a user repository item can have multiple `homeAddress` child items. In the ATG platform, the `state` properties of these items would be named `homeAddress[0].state` and `homeAddress[1].state`. In the XHTML documents, however, both properties would be named `homeAddress.state`.

In addition to the properties you specify in the definition file, the output document also automatically includes a few special metadata properties. These properties provide information that identifies the repository items represented in the document, and they are indicated by a dollar-sign (\$) prefix: `$repositoryId`, `$repository.repositoryName`, and `$itemDescriptor.itemDescriptorName`. It is often a good idea to exclude some of these properties to reduce the size of your index. See [Suppressing Properties](#).

The output for the document-level item also includes a `$url` property and a `$baseUrl` property, which each contain the URL representing this repository item. The difference between these properties is that if a `VariantProducer` is used to generate multiple documents from the same repository item, the `$url` property for each document will include unique query arguments to distinguish the document from the others. The `$baseUrl` property, which omits the query arguments, will be the same for each document. See [Using Variant Producers](#) for more information.

The following example shows sample XHTML output from the definition file that appears in the [Definition File Format](#) section:

```
<html >
  <head>
    <meta content="text/html ; charset=UTF-8" http-equiv="Content-Type" />
    <title>Madeira</title>
    <meta name="atg: date: dateOfBirth" content="-1583175600000" />
    <meta name="atg: string: $repository.repositoryName"
      content="UserProfiles" />
    <meta name="atg: string: $repositoryId" content="743" />
    <meta name="atg: string: $itemDescriptor.itemDescriptorName"
      content="user" />
    <meta name="atg: string: $url " content="atgrep: /UserProfiles/user/743" />
    <meta name="atg: string: $baseUrl "
      content="atgrep: /UserProfiles/user/743" />
    <meta name="atg: string: homeAddress.$repository.repositoryName"
      content="UserProfiles" />
    <meta name="atg: string: homeAddress.$repositoryId" content="743" />
```



```
<meta
  name="atg:string:homeAddress.$itemDescriptor.itemDescriptorName"
  content="contactInfo"/>
</head>
<body>
  <div class="atg:role:firstName" id="0">
    Frankl i n
  </div>
  <div class="atg:role:lastName" id="1">
    Madei ra
  </div>
  <div class="atg:role:homeAddress.address1" id="2">
    802 Repl i cant Bl vd.
  </div>
  <div class="atg:role:homeAddress.city" id="3">
    Los Angel es
  </div>
  <div class="atg:role:homeAddress.state" id="4">
    CA
  </div>
  <div class="atg:role:homeAddress.postalCode" id="5">
    98506
  </div>
  <div class="atg:role:homeAddress.phoneNumber" id="6">
    (212)555-1234
  </div>
</body>
</html >
```

Specifying Properties for Indexing

The [Definition File Format](#) section describes how to specify most common properties for inclusion in the index. This section discusses some less common options in greater detail.

Specifying Multi-Value Properties

In most cases, you specify a multi-value property, such as an array or Collection, using the property element, just as you specify a single-value property. In the following example, the phoneNumbers property stores an array of phone numbers:

```
<text-properties>
  <property name="city"/>
  <property name="state"/>
  <property name="postalCode"/>
```



```
<property name="phoneNumbers" />
</text-properties>
```

Notice that `phoneNumbers` is specified in the same way as `city`, `state`, and `postalCode`, which are all single-value properties. The XHTML output document will include a separate element for each value in the `phoneNumbers` array.

If a property is an array or Collection of repository items, you specify it using the `item` element, and set the `is-multi` attribute to `true`. For example, in a product catalog, a product item will typically have a multi-valued `childSKUs` property whose values are the various SKUs for the product. You might specify the property like this:

```
<item property-name="childSKUs" is-multi="true">
  <text-properties>
    <property name="displayName" />
    <property name="description" />
  </text-properties>
</item>
```

If you index by product (and product is document-level item), the output document will include the `displayName` and `description` value for each of the product's SKUs.

Specifying Map Properties

To specify a Map property, you use the `item` element, set the `is-multi` attribute to `true`, and use the `map-iteration-type` attribute to specify how to output the Map entries. If the Map values are primitives or Strings, set `map-iteration-type` to `wildcard`, as in this example:

```
<item property-name="personalData" is-multi="true" map-iteration-type="wildcard">
  <text-properties>
    <property name="*" />
  </text-properties>
</item>
```

In the XHTML output, the Map keys are treated as subproperties of the Map property, and the Map values are treated as the values of these subproperties. All of the Map entries are included in the output. So, for example, the output from the definition file entry shown above might look like this:

```
<div class="atg:role:personalData.firstName" id="1">Fred</div>
<div class="atg:role:personalData.age" id="2">37</div>
<div class="atg:role:personalData.height" id="3">68</div>
```

If you want to output only a subset of the Map entries, explicitly specify the keys to include, rather than using the wildcard character (*). For example:



```
<i tem property-name="personal Data" is-multi="true" map-iteration-type="wildcard">
  <text-properties>
    <property name="firstName"/>
    <property name="height"/>
  </text-properties>
</i tem>
```

Maps of Repository Items

If the Map values are repository items, set `map-iteration-type` to `values`, and specify the properties of the repository item that you want to output. For example, suppose you want to index a `productInfos` Map property whose keys are product IDs and whose values are product info items:

```
<i tem property-name="productInfos" is-multi="true" map-iteration-type="values">
  <text-properties>
    <property name="displayName"/>
    <property name="size"/>
  </text-properties>
</i tem>
```

The output will include `displayName` and `size` tags for each product info item in the Map. In this case, the Map keys are ignored, the properties of the repository items are treated as subproperties of the Map property, and the values of the items are treated as the values of the subproperties. The output looks like this:

```
<div class="atg:role:productInfos.displayName" id="1">Funny hat</div>
<div class="atg:role:productInfos.size" id="2">8</div>
<div class="atg:role:productInfos.displayName" id="3">Clown shoes</div>
<div class="atg:role:productInfos.size" id="4">14</div>
```

Specifying Properties of Item Subtypes

A repository item type can have subtypes that include additional properties that are not part of the base item type. This feature is commonly used in the ATG Commerce catalog for the SKU item type. A SKU subtype might add properties that are specific to certain SKUs but which are not relevant for other SKUs.

When you list properties to index, you can use the `subtype` attribute of the property element to specify properties that are unique to a specific item subtype. For example, suppose you have a `furniture-sku` subtype that adds properties specific to furniture SKUs. You might specify your SKU properties like this:

```
<i tem property-name="childSKUs">
  <text-properties>
    <property name="description"/>
    <property name="color"/>
    <property name="woodType" subtype="furniture-sku"/>
  </text-properties>
</i tem>
```

```

    </text-properties>
  </item>

```

This specifies that the description and color properties should be included in the XHTML output for all SKUs, but for SKUs whose subtype is furniture-sku, the woodType property should also be included.

The item element also has a subtype attribute for specifying a subtype-specific property whose value is a repository item. If woodType is a repository item, the example above would look something like this:

```

<item property-name="childSKUs">
  <text-properties>
    <property name="description" />
    <property name="color" />
  </text-properties>
  <item property-name="woodType" subtype="furniture-sku"/>
  <text-properties>
    <property name="hardness" />
    <property name="stainType" />
  </text-properties>
</item>
</item>

```

Specifying a Default Property Value

You may find it useful to specify a default value for certain indexed properties. For example, suppose you are indexing address data, and for some addresses no value appears in the repository for the city property. In these cases, you could set the property value in the index to be "city unknown." A user could then search for this phrase and return the addresses whose city property is null.

To set a default value, you use the default-value attribute of the property element. For example:

```
<property name="city" default-value="city unknown" />
```

Specifying Non-Repository Properties

When you index a repository, you can include in the index additional properties that are not part of the repository itself. For example, you might want to include a creationDate property to record the current time when an XHTML document is created. The value for this property could be generated by a custom property accessor that invokes the Java Date class.

To specify a property like this, use the is-non-repository-property attribute of the property element. This attribute indicates that the property is not actually stored in the repository, and prevents warnings from being thrown when the IndexingOutputConfig component starts up. Note that you must also specify a custom property accessor that is responsible for obtaining the property values:

```
<property name="creationDate" is-non-repository-property="true"
  property-accessor="dateAccessor" />
```



If no actual property accessor is needed, set the `property-accessor` attribute to `null`. For example, you might do this if you have a default value that you always want to use for the property:

```
<property name="creationDate" is-non-repository-property="true"
  default-value="Mon Mar 15 16:07:15 EDT 2010"
  property-accessor="null"/>
```

See [Using Property Accessors](#) for more information about custom property accessors.

Suppressing Properties

The output document automatically includes certain standard JavaBean properties of the `RepositoryItem` object as metadata properties. These properties provide information that identifies the repository items represented in the document, and they are indicated by a dollar-sign (\$) prefix: `$repositoryId`, `$repository.repositoryName`, and `$itemDescriptor.itemDescriptorName`.

You may want to return these properties in search results, to enable accessing the indexed repository and repository items in page code. Typically you would do this for the document-level item type. For other item types, you may not need these properties. If you don't, it is a good idea to suppress them from the index, as they may significantly increase the size of the index.

To suppress one of these properties, specify the property in the indexing definition file with the `suppress` attribute. For example:

```
<item property-name="homeAddress" is-document="false">
  <meta-properties>
    <property name="$repositoryId" suppress="true"/>
    <property name="$repository.repositoryName" suppress="true"/>
    <property name="$itemDescriptor.itemDescriptorName" suppress="true"/>
  </meta-properties>
</item>
```

Renaming an Output Property

By default, the name of a property in the XHTML output is the same as its name in the repository. So in the example shown in the [Definition File Format](#) section, the `firstName` property of the `user` item appears in the XHTML as `firstName`. The properties of child items are given output names relative to the top-level item, so the `postalCode` property of the `homeAddress` item appears in the XHTML as `homeAddress.postalCode`.

You can give a property a different name in the XHTML by using the `output-name` attribute of the property element. For example, if you are indexing addresses that are all in the United States, you might want to do this:

```
<property name="postalCode" output-name="zipCode"/>
```

Note that if you specify `output-name` for a property of a child item, the name of the parent item is not prefixed to the output name. So in this example, the property in the XHTML would actually be `zipCode`,

not homeAddress. zipCode. If you want the prefix to be included, you need to specify it explicitly, by setting output-name to homeAddress. zipCode.

Translating Property Values

In some cases, the property values that you want to include in the index (and therefore in the generated XHTML documents) may not be the actual values used in the repository. For example, you may want to normalize values (e.g., index the color values Rose, Vermilion, Crimson, and Ruby all as Red, so when a customer searches for Red, items of these other colors are returned). Or you may want to translate values into another language (e.g., index the color value Green as Vert, so when a customer searches for Vert, green items are returned).

To translate property values for indexing, you use the translate child element of the property element. The translate element has an input attribute for specifying a property value found in the repository, and an output attribute for specifying the value to translate this to in the XHTML documents. For example:

```
<property name="color">
  <translate input="Rose" output="Red" />
  <translate input="Vermilion" output="Red" />
  <translate input="Crimson" output="Red" />
  <translate input="Ruby" output="Red" />
</property>
```

The property element also has prefix and suffix child elements that you can use to append a text string before or after the output property values. For example, you can use the suffix element to add units to the property values:

```
<property name="length">
  <suffix value=" cm" />
</property>
```

Note that the prefix and suffix values are concatenated to the property values exactly as specified, with no additional spaces. If you want spaces before the suffix string or after the prefix string, include the spaces in the value attribute, as in the example above.

You can use the prefix, suffix, and translate elements individually or in combination. The following example translates the size values S, M, and L, to "size small," "size medium," and "size large," to make it easier for customers to search for specific sizes:

```
<property name="size">
  <prefix value="size " />
  <translate input="S" output="small" />
  <translate input="M" output="medium" />
  <translate input="L" output="large" />
</property>
```



Translating Based on Locale

The `prefix`, `suffix`, and `translate` elements all have optional `locale` attributes that allow you to specify different values for different locales. For example:

```
<property name="onSale">
  <translate locale="en_US" input="true" output="on sale"/>
  <translate locale="fr_FR" input="true" output="à la vente"/>
</property>
<property name="weight">
  <suffix locale="en_US" output=" grams"/>
  <suffix locale="fr_FR" output=" grammes"/>
</property>
```

When the XHTML documents are generated, the `IndexingOutputConfig` component determines which tags to use based on the current locale. So if the locale is `en_US`, only the tags that specify that locale are applied.

Multilingual environments typically use the `LocaleVariantProducer`, which generates multiple XHTML documents for each indexed item, one document for each locale specified in its `locales` array property. (See [Using Variant Producers](#) for more information.) If the value of the `locales` array is `en_US`, `fr_FR`, two sets of documents are generated, one using the `translate`, `prefix`, and `suffix` tags whose locale is `en_US`, and one using the tags whose locale is `fr_FR`.

If a tag does not specify a locale, that tag is used as the default when the current locale does not match any of the other tags. In the following example, `Rose` is translated to `Rouge` if the locale is `fr_FR`, but is translated to `Red` for any other locale:

```
<property name="color">
  <translate input="Rose" output="Red"/>
  <translate locale="fr_FR" input="Rose" output="Rouge"/>
</property>
```

Loading the XHTML Documents into ATG Search

Once you have created your XML definition file, the ATG platform can generate the XHTML documents from the items in the repository. To make them available for searching, these documents must be loaded into ATG Search and indexed. Depending on your ATG environment, data loading may be an operation that is performed occasionally (if the content rarely changes) or frequently (if the content changes often). To be as flexible as possible, the ATG platform provides two approaches to loading the data:

- **Bulk loading** loads all of the data generated from a repository into ATG Search. Each time a bulk load is performed, the ATG platform creates and loads the complete set of XHTML documents for the repository, and ATG Search builds a complete new index.



- **Incremental loading** loads only the data that has changed since the last load. The incremental loader records which repository items have changed since the last incremental or bulk load. It deletes the XHTML documents that represent items that have been deleted, and creates and loads XHTML documents for any items that are new or have been modified. ATG Search then incrementally indexes those documents.

Bulk loading and incremental loading are not mutually exclusive. For some environments, only bulk loading will be necessary, especially if content is updated only occasionally. For other environments, incremental loading will be needed to keep the search content up to date, but even in that case it is important to perform a bulk load occasionally to ensure the integrity of the indexed data. In particular, if you make any structural changes to an indexed repository, you should run a bulk load so ATG Search builds a complete index to replace the existing index. Note that modifying a product catalog often results in structural changes to the catalog repository.

In addition to configuring the Nucleus components described in this section, you must also create a project in Search Administration that manages the indexing process. You configure the project by specifying the content to be indexed and various additional parameters, such as text processing options or a schedule for performing the indexing. For example, you might schedule incremental indexing to take place every four hours, and schedule bulk indexing to be performed once a week. For more information, see the [ATG Search Administration Guide](#).

Configuring Document Loading

You supply the configuration for the document loading operation through the following properties of the `IndexingOutputConfig` component:

bulkLoader

A Nucleus component of a class that implements the `BulkLoader` interface. This is typically set to `/atg/search/repository/BulkLoader`. Any number of `IndexingOutputConfig` components can use the same bulk loader.

enableIncrementalLoading

If `true`, incremental loading is enabled.

incrementalLoader

A Nucleus component of a class that implements the `atg.search.repository.indexing.IncrementalLoader` interface. This is typically set to `/atg/search/repository/IncrementalLoader`. Any number of `IndexingOutputConfig` components can use the same incremental loader.

The `/atg/commerce/search/ProductCatalogOutputConfig` component is preconfigured with these settings. If you create a new `IndexingOutputConfig` component, be sure to set these properties.

The `IncrementalLoader` component uses an implementation of the `PropertiesChangeListener` interface to monitor the repository for add, update, and delete events. It then analyzes these events to determine which ones necessitate updating XHTML documents, and creates a queue of the affected repository items. When a new incremental update is triggered, the `IncrementalLoader` processes the items in the queue, generating and loading a new XHTML document for each changed repository item.



Using Monitored Properties

By default, the Incremental Loader determines which changes necessitate updates by monitoring the meta-properties and text-properties specified in the XML definition file. In some cases, however, the properties you want to monitor are not necessarily the ones that you want to output. This is especially the case if you are outputting derived properties, because these properties do not have values of their own.

For example, suppose your user item type has firstName and lastName properties, plus a fullName derived property whose value is formed by concatenating the values of firstName and lastName. You might want to output the fullName property, but to detect when the value of this property changes, you need to monitor (but not necessarily output) firstName and lastName.

You can do this by including a monitor element in your definition file to specify properties that should be monitored but not output. For example:

```
<text-properties>
  <property name="fullName" />
</text-properties>
<monitor>
  <property name="firstName" />
  <property name="lastName" />
</monitor>
```

For information about derived properties, see the [ATG Repository Guide](#).

Tuning Incremental Loading

The number of changed items that accumulate in the queue can vary greatly, depending on how frequently your data changes and how long you specify between incremental updates. Rather than processing all of the changes at once, the IndexingOutputConfig component groups changes in batches called generations.

IndexingOutputConfig has a maxIncrementalUpdatesPerGeneration property that specifies the maximum number of changes that can be assigned to a generation. By default, this value is 1000, but you can change this value if necessary. Larger generations require more ATG platform resources to process, but reduce the number of ATG Search jobs required (and hence the overhead associated with starting up and completing these jobs). Smaller generations require fewer ATG platform resources, but increase the number of ATG Search jobs.

Viewing the XHTML Output

The bulk loader and the incremental loader generate the XHTML representations of the data. If loading is initiated from Search Administration, the loaders use a special document submitter (created automatically) to submit the documents to Search Routing, which writes batches of documents to disk as command files during the index estimation process. These command files are then executed to stream the XHTML data to the search engine for indexing.



You can initiate loading from ATG Dynamo Server Admin, to help optimize and debug your XHTML output. In this case, XHTML documents are not submitted to Routing, and no indexing job is initiated. Instead, the documents are directed either to the console or to files, depending on the value of the `IndexingOutputConfig` component's `documentSubmitter` property:

- To direct the documents to files, set `documentSubmitter` to a component of class `atg.repository.search.indexing.submitter.FileDocumentSubmitter`.
- To direct the documents to the console, set `documentSubmitter` to a component of class `atg.repository.search.indexing.submitter.ConsoleDocumentSubmitter`, or just leave the property null, and a `ConsoleDocumentSubmitter` will be created automatically.

To initiate loading from ATG Dynamo Server Admin:

1. Access the page for the `IndexingOutputConfig` component in the Component Browser of the ATG Dynamo Server Admin.
2. In the Methods section, click the link for `bulkLoadForDiagnosti cs`.
3. At the confirmation prompt, click `Invoke Method`.

Note that for a multisite application, the generated documents will reflect all enabled sites.

Look over the XHTML output carefully to identify potential improvements. For example, if you see redundant values for certain properties, you can specify the `unique` filter in the XML definition file for those properties (see [Using Property Value Filters](#)). Or if you notice that certain text properties have values that users are unlikely to search for, you can remove those properties from the definition file.

Configuring a FileDocumentSubmitter Component

If `documentSubmitter` is set to a component of class `FileDocumentSubmitter`, a separate file is created for each XHTML document generated. The location and names of the files are automatically determined based on the following properties:

baseDirectory

The pathname of the directory to write the files to.

filePrefix

The string to prepend to the name of each generated file. Default is the empty string.

fileSuffix

The string to append to the name of each generated file. Default is `".html"`.

nameByRepositoryId

If `true`, each filename will be based on the repository ID of the item the file represents. If `false` (the default), files are named `0.html`, `1.html`, etc.

overwriteExistingFiles

If `true`, if the generated filename matches an existing file, the existing file will be overwritten by the new file. If `false` (the default), the new file will be given a different name to avoid overwriting the existing file.



Viewing the Output in the Component Browser

You can view the XHTML output for a single document-level repository item (and its child items) in the Component Browser of ATG Dynamo Server Admin. To do this, access the page for the `IndexingOutputConfig` component in the Component Browser. The page will include a Test Document Generation section that looks similar to this:

Test Document Generation

Generate XHTML document or documents for repository item type **product**

product ID:

Fill in the repository ID, and click Generate. The page will display the XHTML output. The output is not directed to Routing, and no indexing job is initiated.

Note that you can also view XHTML output in ATG Search Administration. See the [Content Inspection](#) section.





6 Customizing the XHTML Output

The previous chapter described the basic configuration of the repository indexing system. For some environments, this configuration may be sufficient. However, many environments have complex repositories that require custom handling of the XHTML output. For example, a repository may contain multiple versions of the same data, each in a different language. To enable the repository indexing system to handle situations like this, the `atg.repository.search.indexing` package includes interfaces and classes that you can implement or extend to handle the particular needs of your environment.

These interfaces include `PropertyAccessor`, which defines how the document loaders obtain property values, and `VariantProducer`, which specifies logic for creating multiple XHTML documents from the same repository item. Classes that implement these interfaces must be stateless, because they can be accessed by multiple threads at the same time. Rather than maintaining state themselves, these classes instead use the `Context` class to store state information and to pass data to each other.

In addition to the `PropertyAccessor` and `VariantProducer` interfaces, the `atg.repository.search.indexing` package includes two other interfaces that you can use to customize your output, `PropertyFormatter` and `PropertyValueFilter`. Classes that implement these interfaces are applied after all of the output properties have been gathered, so these classes do not have access to the `Context` object.

This chapter discusses:

[Using Property Accessors](#)

[Using Variant Producers](#)

[Using Property Formatters](#)

[Using Property Value Filters](#)

For additional information about the classes and interfaces described in this chapter, see the [ATG API Reference](#).

Using Property Accessors

The document loaders access property values through an implementation of the `atg.repository.search.indexing.PropertyAccessor` interface. This interface defines separate methods for obtaining text properties, meta properties, and properties of child items. By default, the loaders in most cases use the `atg.repository.search.indexing.PropertyAccessorImpl` class, which invokes the `RepositoryItem.getPropertyValue()` method. You can write your own



implementations of `PropertyAccessor` that use custom logic for determining the values of properties that you specify. The simplest way to do this is to subclass `PropertyAccessorImpl`.

In the definition file, you can specify a custom property accessor for a property by using the `property-accessor` attribute. For example, suppose you have a Nucleus component named `/MyStuff/MyPropertyAccessor`, of a custom class that implements the `PropertyAccessor` interface. You can specify it in the definition file like this:

```
<property name="price" property-accessor="/MyStuff/MyPropertyAccessor" />
```

The value of the `property-accessor` attribute is the absolute path of the Nucleus component. To simplify coding of the definition file, you can map `PropertyAccessor` Nucleus components to simple names, and use those names as the values of `property-accessor` attributes. For example, if you map the `/MyStuff/MyPropertyAccessor` component to the name `myAccessor`, the above tag becomes:

```
<property name="price" property-accessor="myAccessor" />
```

You can perform this mapping by setting the `propertyAccessorMap` property of the `IndexingOutputConfig` component. This property is a `Map` in which the keys are the names and the values are `PropertyAccessor` Nucleus components that the names represent. For example:

```
propertyAccessorMap+=\
    myAccessor=/MyStuff/MyPropertyAccessor
```

FirstWithLocalePropertyAccessor

The `atg.repository.search.indexing.accessor` package includes a subclass of `PropertyAccessorImpl` named `FirstWithLocalePropertyAccessor`. This property accessor works only with derived properties that are defined using the `firstWithLocale` derivation method. `FirstWithLocalePropertyAccessor` determines the value of the derived property by looking up the `currentDocumentLocale` property of the `Context` object. Typically, this property is set by the `LocaleVariantProducer`, as described in [Accessing the Context Object](#).

You can specify this property accessor in your definition file using the attribute value `firstWithLocale`. (Note that you do not need to map this name to the property accessor in the `propertyAccessorMap`.) For example:

```
<property name="displayName" property-accessor="firstWithLocale" />
```

For information about the `firstWithLocale` derivation method, and about derived properties in general, see the [ATG Repository Guide](#).

GenerativePropertyAccessor

The `atg.repository.search.indexing.accessor` package includes a subclass of `PropertyAccessorImpl` named `GenerativePropertyAccessor`. This is an abstract class that adds the ability to generate multiple property names and associated values for a single property tag in the indexing definition file. For example, the `PriceListMapPropertyAccessor` subclass of



GeneratePropertyAccessor generates, for a single price property in the definition file, a separate XHTML tag for each price list.

You can write your own subclass of GeneratePropertyAccessor. Your subclass must implement the getPropertyNamesAndValues method. This method returns a Map in which each key is a property name, and the corresponding Map value contains the value to be associated with the property name.

Using Variant Producers

By default, for the repository item type designated by the is-document attribute, the IndexingOutputConfig component generates one XHTML document per item. In some cases, though, you may want to generate more than one document for each repository item. For example, suppose you have a repository whose text properties are stored in both French and English, and the language displayed is determined by the user's locale setting. In this case you will typically want to create two documents from each repository item, one with the text content in French, and the other one in English.

To handle situations like this, the ATG platform provides an interface named atg.repository.search.indexing.VariantProducer. You can write your own implementations of the VariantProducer interface, or you can use implementations included with the ATG platform. This interface defines a single method, prepareNextVariant(), for determining the number and type of variants to produce. Depending on how your repository is organized, implementations of this method can use a variety of approaches for determining how to generate variant documents.

LocaleVariantProducer

The ATG platform includes an implementation of the VariantProducer interface, atg.repository.search.indexing.producer.LocaleVariantProducer, for generating variant documents for different locales. This class has a locales property where you specify the list of locales to generate variants for. For example:

```
locales=en_US, fr_FR
```

You specify the VariantProducer components to use by setting the variantProducers property of the IndexingOutputConfig component. Note that this property is an array; you can specify any number of VariantProducer components, and the IndexingOutputConfig will generate a separate variant for each possible combination of values of the variant criteria. For example, suppose you use LocaleVariantProducer and the locales property is set as shown above. If you add a second VariantProducer that creates three variants (1, 2, and 3), the total number of variants generated for each repository item would be six (French 1, English 1, French 2, English 2, French 3, and English 3).

Accessing the Context Object

When a document loader is invoked, it creates an object of class atg.repository.search.indexing.Context. The PropertyAccessor and VariantProducer classes use this object to store and retrieve state information about the loading process. The Context object contains the current list of parent repository items that were navigated to reach the current item



(both product specifiers and repository items), the current document URL (if any), the current collected output values (if any), and status information.

One of the main uses of the `Context` object is to store information used to determine what variant to generate next. For example, each time a new document is generated, the `Local eVariantProducer` uses the next value in its `Local e` array to set the `currentDocumentLocal e` property of the `Context` object. A `PropertyAccessor` instance might read the `currentDocumentLocal e` property and use its current value to determine the locale to use for the property.

For more information about the `Context` object, see the [ATG API Reference](#).

Using Property Formatters

If a property takes an object as its value, the document loader must convert that object to a string to include it in an output document. The `PropertyFormatter` interface defines two methods for performing this conversion: `formatText()` (for text properties) and `formatMeta()` (for meta properties).

By default, the document loaders use the implementation class `atg.repository.search.indexing.formatter.PropertyFormatterImpl`. For text properties, this class simply invokes the object's `toString()` method. For metadata properties, it invokes `getLong()` for numbers, `getTime()` for dates, and `toString()` for other objects.

You can write your own implementations of `PropertyFormatter` that use custom logic for performing the conversion. The simplest way to do this is to subclass `PropertyFormatterImpl`.

In the definition file, you can specify a custom property formatter by using the `formatter` attribute. For example, suppose you have a Nucleus component named `/MyStuff/MyPropertyFormatter`, of a custom class that implements the `PropertyFormatter` interface. You can specify it in the definition file like this:

```
<property name="price" formatter="/MyStuff/MyPropertyFormatter" />
```

The value of the `formatter` attribute is the absolute path of the Nucleus component. To simplify coding of the definition file, you can map `PropertyFormatter` Nucleus components to simple names, and use those names as the values of `formatter` attributes. For example, if you map the `/MyStuff/MyPropertyFormatter` component to the name `myFormatter`, the above tag becomes:

```
<property name="price" formatter="myFormatter" />
```

You can perform this mapping by setting the `formatterMap` property of the `IndexingOutputConfig` component. This property is a `Map` in which the keys are the names and the values are `PropertyFormatter` Nucleus components that the names represent.



Using Property Value Filters

In some cases, it is useful to filter a set of property values before outputting an XHTML document. For example, suppose your document-level item has many child items, and each child item has a color property that can have only a few possible values. Rather than outputting the color of every child item, you could include each color in the document just once, by using a filter that removes duplicate property values.

The `PropertyValuesFilter` interface defines a method for filtering property values. The `atg.repository.search.indexing.filter` package includes several implementations of this interface:

- `UniqueFilter` removes duplicate property values, returning only the unique values.
- `ConcatFilter` concatenates all of the property values into a single string.
- `UniqueWordFilter` concatenates all of the property values into a single string, and then removes all duplicate words.
- `HtmlFilter` removes any HTML markup from the property values.

This section provides information about what these filters do and when they're appropriate.

In the definition file, you can specify property filters by using the `filter` attribute. Note that you can use multiple filters on the same property. The value of the `filter` attribute is a comma-separated list of Nucleus components. The component names must be absolute pathnames.

To simplify coding of the definition file, you can map `PropertyValuesFilter` Nucleus components to simple names, and use those names as the values of `filter` attributes. You can perform this mapping by setting the `filterMap` property of the `IndexingOutputConfig` component. This property is a `Map` in which the keys are the names and the values are `PropertyFilter` Nucleus components that the names represent.

Note, however, that you do not need to perform this mapping to use the `UniqueFilter`, `ConcatFilter`, `UniqueWordFilter`, or `HtmlFilter` class. These classes are mapped by default to the following names:

Filter Class	Name
<code>UniqueFilter</code>	<code>unique</code>
<code>ConcatFilter</code>	<code>concat</code>
<code>UniqueWordFilter</code>	<code>uniqueword</code>
<code>HtmlFilter</code>	<code>html</code>

So, for example, you can specify `UniqueFilter` like this:

```
<property name="color" filter="unique"/>
```



UniqueFilter

You may be able to reduce the size of your index by filtering the property values to remove redundant entries. For example, suppose each XHTML document represents a product with several child SKUs. You might include the SKUs' `salePrice` property in the index as a metadata property, so it can be used for faceting. Depending on the product, many of the SKUs may have the same value for `salePrice`. So the resulting entries in an XHTML document might look something like this:

```
<meta name="atg:float:childSKUs.salePrice" content="190.0" />
<meta name="atg:float:childSKUs.salePrice" content="205.0" />
<meta name="atg:float:childSKUs.salePrice" content="190.0" />
<meta name="atg:float:childSKUs.salePrice" content="205.0" />
<meta name="atg:float:childSKUs.salePrice" content="205.0" />
```

By filtering out redundant entries, you can reduce this to:

```
<meta name="atg:float:childSKUs.salePrice" content="190.0" />
<meta name="atg:float:childSKUs.salePrice" content="205.0" />
```

To automatically perform this filtering, specify the `UniqueFilter` class in the XML definition file:

```
<property name="salePrice" filter="unique" />
```

As a general rule, it is a good idea to specify the unique filter for a property if multiple items in an XHTML document may have identical values for that property. If you specify this filter for a property and every value of that property in an XHTML document is unique (or if only one item with that property appears in the document), the unique filter will have no effect on the resulting XHTML (either negative or positive). However, executing this filter increases processing time to create the document, so it is a good idea to specify it only for properties that will benefit from it.

ConcatFilter

You may also be able to reduce the size of your index by concatenating the values of text properties. For example, suppose a product's SKUs have a `color` property, whose values are red, green, blue, and yellow. The resulting entries in the XHTML document will be:

```
<div class="atg:role:childSKUs.color" id="1">red</div>
<div class="atg:role:childSKUs.color" id="2">green</div>
<div class="atg:role:childSKUs.color" id="3">blue</div>
<div class="atg:role:childSKUs.color" id="4">yellow</div>
```

By concatenating the values, you can reduce this to:

```
<div class="atg:role:childSKUs.color" id="1">red green blue yellow</div>
```

To combine these values into a single tag, specify the `ConcatFilter` class in the XML definition file:

```
<property name="color" filter="concat" />
```




This setting invokes an instance of the `atg.repository.search.indexing.filter.ConcatFilter` class. Note that you do not need to create a Nucleus component to use this filter.

Some guidelines for using `concat`:

- Do not use the `concat` filter for metadata properties. This will almost always produce undesirable results.
- Concatenating property values may have some slight effects on the search results. It is probably best to use the `concat` filter only for properties with short text values, such as the `color` property shown above. Concatenating properties with long text values, such as `longDescription`, can have negative effects.
- You can use both the `unique` and the `concat` filters on the same text property, by setting the value of the `filter` attribute to a comma-separated list. The filters are invoked in the order that they are listed, so it is important to put the `unique` filter first for it to have an effect. For example:

```
<property name="color" filter="unique, concat"/>
```

UniqueWordFilter

The `atg.repository.search.indexing.filter.UniqueWordFilter` class concatenates the values of a property into a single string, and then removes all duplicate words. For example, suppose a product's SKUs have a `size` property, and the resulting entries in the XHTML document are:

```
<div class="atg:role:childSKUs.size" id="1">medium</div>
<div class="atg:role:childSKUs.size" id="2">large</div>
<div class="atg:role:childSKUs.size" id="3">x large</div>
<div class="atg:role:childSKUs.size" id="4">xx large</div>
```

By applying `UniqueWordFilter`, you can reduce this to:

```
<div class="atg:role:childSKUs.size" id="1">medium large x xx</div>
```

Note that `UniqueWordFilter` converts all Strings to lowercase, so that redundant words are eliminated even if they don't have identical case.

You can specify `UniqueWordFilter` in the XML definition file like this:

```
<property name="size" filter="uniqueword"/>
```

You do not need to create a Nucleus component to use this filter.

Although `UniqueWordFilter` concatenates values and removes redundancies, it is not equivalent to using `UniqueFilter` and `ConcatFilter`. `UniqueFilter` considers the entire string when it eliminates redundant values, not individual words. In this example, each complete string is unique, so `UniqueFilter` would not actually eliminate any values, and the result would be:



```
<div class="atg:role:childSKUs.size" id="1">
  medium Large x Large xx Large
</div>
```

Note: You should use `UniqueWordFilter` carefully, as under certain circumstances it can have undesirable effects. If you use a custom dictionary that includes multi-word terms, searches for those terms may not return the expected results, because the filter may rearrange the order of the words in the index.

HtmlFilter

The `atg.repository.search.indexing.filter.HtmlFilter` class removes any HTML markup from a property value. This is useful, for example, if text properties include tags for bolding or italicizing certain words, as in this `LongDescription` property of a product:

```
You'll <b>love</b> this Italian <i>leather</i> sofa!
```

Because the HTML markup is included in the index, searches may return unexpected results. In this example, searching for “leather sofa” might not return the product, because that string does not actually appear in the `LongDescription` property.

Using `HtmlFilter`, this value appears in the index as:

```
<div class="atg:role:LongDescription" id="1">
  You'll love this Italian leather sofa!
</div>
```

Now a search for “leather sofa” will find the value in this property, and return this product.



7 Indexing ATG Commerce Catalog Data

This chapter discusses issues specific to indexing repository items in an ATG Commerce catalog. It includes the following topics:

[Indexing the Product Catalog Repository](#)

[Indexing Price Data in Price Lists](#)

Indexing the Product Catalog Repository

The DCS. Search module includes an `IndexingOutputConfig` component and definition file that are preconfigured to work with a typical catalog repository. The component, `/atg/commerce/search/ProductCatalogOutputConfig`, is configured to use the XML definition file `/atg/commerce/search/product-catalog-output-config.xml`. This definition file is intended to be used as a starting point; you will typically need to modify it to reflect your catalog and to optimize the XHTML output.

By default, the `ProductCatalogOutputConfig` component's top-level `item` element is:

```
<item is-document="true" item-descriptor-name="product">
```

This element configures the `ProductCatalogOutputConfig` to create an XHTML document for each product in the product catalog. This means that each document indexed by ATG Search corresponds to a product, so that when a user searches the catalog, each individual result returned represents a product. The definition file specifies that each output document should include information about the product's parent category and child SKUs (as well as the product itself), so that users can search category or SKU properties in addition to product properties.

SKU-Based versus Product-Based Indexing

You may prefer to create a separate XHTML document for each SKU, so that each SKU can be returned as a separate search result. For example, consider an environment where different SKUs for a given product have different prices. With product-based indexing, if a search query specifies a constraint such as "return items whose price is less than \$10.00," the query will return any product that has a SKU whose price is less than \$10.00, even if all the other SKUs of that product have higher prices. With SKU-based indexing, the query will return only the individual SKUs whose price is under \$10.00.



This behavior is especially desirable for faceted search. Note, however, that indexing by SKU results in a larger index than product-based indexing, because of the larger number of XHTML documents, and therefore can lead to slower performance. If most of your products have only a single SKU, or if your faceting properties have the same value for all of a product's SKUs (e.g., if you use manufacturer as a faceting property), product-based indexing is more efficient.

Enabling SKU-Based Indexing

To support SKU-based indexing, ATG Commerce includes the DCS. Search. Index. SKU indexing module. This module makes various configuration changes to ATG Commerce components to enable SKU-based indexing rather than product-based indexing. For example, it modifies the ProductCatalogOutputConfig component's definition file (product-catalog-output-config.xml), setting the is-document attribute to false for the product item type, and to true for the childSKUs item type.

The following is a sample XHTML document generated for a SKU, based on the product-catalog-output-config.xml definition file. Note that product remains the top-level item, so although a separate document is created for each SKU, the property names are still relative to the product:

```
<html >
  <head>
    <meta content="text/html; charset=UTF-8" http-equiv="Content-Type" />
    <title>Wheel Cylinder</title>
    <meta name="atg:string:$repositoryId" content="prod10003"/>
    <meta name="atg:date:creationDate" content="988310992"/>
    <meta name="atg:string:$repository.repositoryName"
      content="CustomProductCatalog"/>
    <meta name="atg:string:$itemDescriptor.itemDescriptorName" content="product"/>
    <meta name="atg:string:parentCategory.$repositoryId" content="cat20039"/>
    <meta name="atg:date:parentCategory.creationDate" content="983821019"/>
    <meta name="atg:string:ancestorCategories.$repositoryId" content="cat20002"/>
    <meta name="atg:string:ancestorCategories.$repositoryId" content="cat20039"/>
    <meta name="atg:string,docset:ancestorCategories.catalogSpecified"
      content="cat20002_300004"/>
    <meta name="atg:string,docset:ancestorCategories.catalogSpecified"
      content="cat20039_300004"/>
    <meta name="atg:string:catalogs.$repositoryId" content="300004"/>
    <meta name="atg:string,index:childSKUs.$repositoryId" content="sku10004"/>
    <meta name="atg:string:childSKUs.$itemDescriptor.itemDescriptorName"
      content="configurableSku"/>
    <meta name="atg:date:childSKUs.creationDate" content="988311239"/>
    <meta name="atg:float:childSKUs.listPrice" content="0.0"/>
    <meta name="atg:boolean:childSKUs.onSale" content="false"/>
    <meta name="atg:string:childSKUs.url"
      content="atgrep:/CustomProductCatalog/configurableSku/sku10004?catalog=300004&locale=en_US"/>
    <meta name="atg:string,index:childSKUs.baseUrl"
      content="atgrep:/CustomProductCatalog/configurableSku/sku10004"/>
  </head>
```



```

<body>
  <div class="atg:role:displayName" id="0">
    Wheel Cylinder
  </div>
  <div class="atg:role:description" id="1">
    Fits GM vehicles
  </div>
  <div class="atg:role:longDescription" id="2">
    Kawaf wheel cylinders are designed to replace OEM parts. Available for
    a variety of applications.
  </div>
  <div class="atg:role:keywords" id="3">
    wheel cylinder
  </div>
  <div class="atg:role:parentCategory.displayName" id="4">
    Drum Components
  </div>
  <div class="atg:role:childSKUs.displayName" id="5">
    wheel cylinder, GM
  </div>
  <div class="atg:role:childSKUs.listPrice" id="6">
    0.0
  </div>
</body>
</html >

```

Handling Catalog-Specific Properties

In addition to the category, product, and SKU items, the catalog repository includes catalog items that represent different hierarchies of categories and products. Each user is assigned one catalog, and sees the navigational structure, product and SKU items, and property values associated with that catalog. A given product may appear in multiple catalogs. The product repository item type includes a catalogs property which is an array of the catalogs the product is included in.

As a result of this flexibility, the property values of an individual category, product, or SKU can vary depending on the catalog. Therefore when you index the catalog, you may need to generate multiple XHTML documents for each product or SKU (one for each catalog the item is included in).

To support creation of multiple XHTML documents per product or SKU, ATG Commerce includes these key components:

- `/atg/commerce/search/CustomCatalogPropertyAccessor` – This component is of class `atg.commerce.search.producer.CustomCatalogPropertyAccessor`, which implements the `atg.repository.search.indexing.PropertyAccessor` interface. (See [Using Property Accessors](#).) This accessor is specified in the `product-catalog-output-config.xml` definition file to obtain the value of the catalogs property:



```
<item is-multi="true" property-accessor="customCatalog"
  property-name="catalogs">
```

The CustomCatalogPropertyAccessor component is mapped to the name customCatalog by the ProductCatalogOutputConfig component's propertyAccessorMap property:

```
propertyAccessorMap+=\
  customCatalog=CustomCatalogPropertyAccessor
```

- /atg/commerce/search/CustomCatalogVariantProducer – This component is of class atg.commerce.search.producer.CustomCatalogVariantProducer, which implements the atg.repository.search.indexing.VariantProducer interface. (See [Using Variant Producers.](#)) This component is added to the ProductCatalogOutputConfig component's variantProducers property:

```
variantProducers+=\
  CustomCatalogVariantProducer
```

Note that if a user has not been assigned a catalog, search queries and facets may not work properly for that user. This is generally not an issue for actual customers, who should all have catalogs assigned to them, but it may affect internal users testing the application. Make sure any internal users accessing the application have catalogs assigned to them.

Generating the XHTML Documents

The CustomCatalogPropertyAccessor and the CustomCatalogVariantProducer work together to generate a separate XHTML document for each catalog a product is in.

ProductCatalogOutputConfig uses the CustomCatalogVariantProducer to generate, for an individual product, a separate XHTML document for each catalog listed in the catalogs property of the product. The value of this property in the catalog repository is an array of all the catalogs the product is included in. The variant producer iterates through these catalogs individually, so that each XHTML document will contain, for other properties of the product, only the values associated with a single catalog.

The values of the product's properties are obtained by the CustomCatalogPropertyAccessor, based on the catalog associated with the current variant. Note that these properties must be derived properties that use the CatalogMapDerivation class to derive their catalog-specific values.

Note: If your catalogs do not use catalog-specific values, you should disable the CustomCatalogVariantProducer by removing it from the variantProducers property of ProductCatalogOutputConfig. This will reduce the number of XHTML documents generated, decreasing the time it takes to index the data.

Determining Which Properties to Index

The key to producing optimal XHTML documents is specifying the right set of properties – omitting properties that you don't need, and including the ones that you do. The smaller your XHTML documents are, the smaller your index will be; a smaller index uses less memory and can be searched more quickly than a larger one. Of course, an index that's missing important data is not very useful, so you need to



make tradeoffs based on the needs of your environment. This section provides guidelines to help you determine which properties to include as text properties and which ones to include as metadata.

In addition to omitting unnecessary properties, you can reduce the size of your XHTML documents by applying various property value filters to the repository data. For information about these filters and how to use them, see the [Using Property Value Filters](#) section of the [Customizing the XHTML Output](#) chapter. Also, see the [Indexing Output Config Analysis](#) section for tools to help pinpoint unnecessary properties in your index.

Guidelines for Text Properties

Some guidelines for determining which properties to include as text properties, and which ones to omit:

- **Include** properties whose values contain text that users are likely to search for. For product catalogs, typical properties might be `description`, `longDescription`, `color`, and `brand`.
- **Don't include** multiple properties that contain the same data. For example, if a product's `description` and `longDescription` properties always contain the same text, include only one of these properties in the index. (Similarly, if the `description` property always contains a subset of the text in the `longDescription` property – such as the first sentence – then include `longDescription` and omit `description`.)
- **Don't include** properties that have values that users are unlikely to search for. These include date and Boolean values, and some types of numeric values. (Note, however, that these properties are often appropriate for metadata.)
- **Don't include** properties that may lead to irrelevant or undesired results. For example, suppose you have a `Shoes` category with two subcategories, `Men's Shoes` and `Women's Shoes`. If the `description` property of the `Shoes` category is “Men's and women's shoes,” and you include `ancestorCategories.description` in the index, searches for “women's shoes” will return men's shoes as well as women's, because the `ancestorCategories.description` property for each item in `Men's Shoes` contains the phrase “women's shoes.”
- **Be careful** not to confuse the name of a property with its values. For example, you might be inclined to include a Boolean property named `onSale`, on the assumption that users may include “on sale” in search queries. But the resulting index will *not* include `onSale` (the name of the property), it will include `true` and `false` (the values of the property), so searching for “on sale” will not have the desired effect.

Keep in mind that these are just guidelines, and you may need to deviate from them depending on the requirements of your environment. For example, you may want to include a Boolean property as a text property if you translate `true` and `false` into searchable Strings. (See [Translating Property Values](#).) Or there may be certain numeric properties (e.g., product codes) that you may want to make available for searching.

Guidelines for Metadata Properties

Some guidelines for determining which properties to include as metadata properties, and which ones to omit:



- **Include** any properties that you want to use for creating facets (for example, faceting by size).
- **Include** any properties that you want to use in search configuration rankings and rules (for example, ranking by brand).
- **Include** any properties that you want to use as sort criteria (for example, sorting by sale price).
- **Include** any properties that you want to use in query constraints (for example, using catalog id to restrict results to items in the catalog assigned to the user).
- **Include** any properties that you want to be able to access in search results. For example, you might include \$repositoryId for the document-level item so you can access the repository item that a search result represents. But don't include such properties if you don't need them; see [Suppressing Properties](#).
- **Don't include** properties that do not match any of these criteria. In particular, you should not include long text fields (such as LongDescription) as metadata properties. Although it is possible to include these properties in constraints (e.g., "do not return results whose LongDescription contains Acme"), such constraints are very inefficient, and these properties can increase the index size significantly.

Excluding Uncategorized Products from Indexing

You can limit the set of items to index by setting the `IndexingOutputConfig` component's `repositoryItemGroup` property to a component whose class implements the `atg.repository.RepositoryItemGroup` interface. This interface defines a logical grouping of repository items. Only the items that are included in this logical grouping will be included in the index.

For example, many commerce applications display search results only for items that can also be viewed by browsing the catalog hierarchy. To be viewed this way, a product must have an associated category. Therefore, the `ProductCatalogOutputConfig` component's `repositoryItemGroup` property is set by default to `/atg/commerce/search/IndexedItemsGroup`. The `IndexedItemsGroup` component uses a targeting ruleset to select only products that are properly categorized. Note that `IndexedItemsGroup` has a `repository` property that specifies the repository that the items are selected from; this must match the repository that the `ProductCatalogOutputConfig` is associated with.

It is also possible to specify a repository item group in the indexing definition file using the `repository-item-group` attribute of the top-level `item` element. If a repository item group is specified in the definition file and also by the component's `repositoryItemGroup` property, the value set by the `repositoryItemGroup` property overrides the value set in the definition file.

`IndexedItemsGroup` uses the following targeting ruleset, which returns all products that have ancestor catalogs:

```
<ruleset>
  <accepts>
    <rule op=isNotNull >
      <valueof target="computedCatalogs" >
    </rule>
```




```
</accepts>  
</ruleset>
```

Indexing a Multisite Catalog Repository

If you are using ATG's multisite support, many of the item types in the catalog repository have a `siteIds` property whose value is a comma-separated list of the sites an item appears on. For example, if you have three sites, A, B, and C, and a certain product is available on sites A and C (but not B), the value of the product's `siteIds` property would be `siteA, siteC` (assuming those are the site IDs).

The `siteIds` properties in the catalog repository are defined as context membership properties. For the document-level item type, the XHTML output includes a special `$siteId` metadata property representing the repository item's context membership property. (The metadata property in the XHTML is always named `$siteId`, regardless of the actual name of the context membership property.) The output includes a separate tag for each site listed in the context membership property. For example:

```
<meta name="atg:string,index:$siteId" content="siteA"/>  
<meta name="atg:string,index:$siteId" content="siteC"/>
```

Note that the XHTML output documents include tags only for sites that correspond to the content set selected in Search Administration. For example, if the value of a product's `siteIds` property is `siteA, siteC, siteD`, but the content set being indexed is associated only with sites C and D, the XHTML document will not include a tag for site A. If an item's `siteIds` property is null, or if it lists only sites that are not associated with the selected content set, no XHTML document is generated for the item.

Creating a Separate Document for Each Site

If you want to create a separate document for each site, you can do so by using the `/atg/search/repository/UniqueSiteVariantProducer` component. This component is of class `atg.commerce.search.producer.UniqueSiteVariantProducer`, which implements the `atg.repository.search.indexing.VariantProducer` interface.

`UniqueSiteVariantProducer` creates a separate document for each site that meets both of these criteria:

- The ID of the site is included in the `siteIds` property of the item being indexed.
- The site corresponds to the content set selected in Search Administration.

For example, if you are indexing by product and the value of a product's `siteIds` property is `siteE, siteF, siteG`, and the content set being indexed is associated with sites B, E, and F, `UniqueSiteVariantProducer` creates two XHTML documents, one for site E and one for site F. The documents are virtually identical, except that each one has a different value for the `$siteId` property. A search that returns the product may return multiple instances, depending on the query constraints and grouping options used.

To use the `UniqueSiteVariantProducer`, add it to the `ProductCatalogOutputConfig` component's `variantProducers` property:

```
variantProducers+=\
/atg/search/repository/UniqueSiteVariantProducer
```

Note that this variant producer does not require a corresponding property accessor.

Indexing Price Data in Price Lists

If your ATG Commerce catalog uses price lists, a single item may have multiple prices, with the actual price applied depending on who is purchasing the item. A customer is assigned a price list, and when that customer accesses a product or SKU, ATG Commerce looks up the item's price in that price list. Another customer looking at that same item might see a different price, if that customer is assigned a different price list.

When a customer issues a search query, the results may depend on the correct prices for that customer being present in the index. For example, if the query has a constraint such as "return items whose price is less than \$5.00," the set of items matching this constraint will depend on the price list the customer is assigned. Therefore, ATG Search must be able to determine which price list is assigned to the user, look up the price in that price list, and apply the constraint based on that price.

There are three key aspects of working with price lists in ATG Search:

- Because prices in price lists aren't stored in properties of the product or SKU, you must create a special property to represent the price in queries. This property can then be used for faceting and in constraints. See [Creating the Price Property](#).
- The XHTML document created from a product or SKU must include a price for each price list, so the resulting index will include all of the prices. See [Indexing Price List Data](#).
- The search client must determine which price list is assigned to the customer issuing the query, and return the price that is associated with that price list. See *Specifying a Price List in the Search Request* in the [ATG Search Query Guide](#).

Creating the Price Property

Because prices in price lists aren't stored in properties of the product or SKU, you must create a special property to represent the price in queries. You can use a component of class `atg.commerce.search.PriceListPropertyProvider` to create a metadata property whose value is set at query time to the item price from the appropriate price list. ATG Commerce includes a component of this class, `/atg/commerce/search/PriceListPropertyProvider`, which creates a metadata property named `price`. You can use this property for constraints and faceting as if it were actually a property of the indexed item.

Note that if you create an additional `PriceListPropertyProvider` component, you must register it with the `/atg/commerce/search/MetaPropertyResolver` component by adding it to that component's `propertyProviders` array property. For example:

```
propertyProviders+=/mystuff/MyPriceListPropertyProvider
```



Indexing Price List Data

When you index your catalog, the item prices are read from the price lists and used to construct meta tags in the XHTML documents. A separate meta tag is created for each price list, and the property name in the tag identifies the price list the tag is associated with. To read the prices from the price lists, you use a property accessor of class

`atg.repository.search.indexing.accessor.PriceListMapPropertyAccessor`. (This class is a subclass of `atg.repository.search.indexing.accessor.GenerativePropertyAccessor`, which is described in the [GenerativePropertyAccessor](#) section.)

ATG Commerce provides a component of this class,

`/atg/commerce/search/PriceListMapPropertyAccessor`. You can specify this property accessor in the definition file like this:

```
<property name="price" type="float"
  property-accessor="PricePropertyAccessor"
  is-non-repository-property="true"/>
```

The property name you use is the one created in the `PriceListPropertyProvider` component (by default, `price`). The `property-accessor` attribute is set to `PricePropertyAccessor`, which is mapped to `/atg/commerce/search/PriceListMapPropertyAccessor` in the `ProductCatalogOutputConfig` component. The `is-non-repository-property` attribute indicates that the property is not actually stored in the catalog repository; this attribute prevents warnings from being thrown when the `IndexingOutputConfig` component starts up.

When the `PriceListMapPropertyAccessor` is invoked for an item, it iterates through all available price lists and outputs a separate meta tag for each one. Each tag contains the item price from one price list. The format of the names of the output properties in the XHTML is set through the `PricePropertyPrefix` property of the `PriceListMapPropertyAccessor` component. By default, the value of this property is:

```
childSKUs.price@
```

The price list ID is appended to this prefix in the tag associated with a given price list. For example, if there are four possible price lists, the XHTML output might include:

```
<meta name="atg:float:childSKUs.price@pl90001" content="9.99" />
<meta name="atg:float:childSKUs.price@pl90002" content="7.99" />
<meta name="atg:float:childSKUs.price@pl90003" content="5.99" />
<meta name="atg:float:childSKUs.price@pl90004" content="4.99" />
```

So, for example, the price for this item in price list `pl90003` is 5.99.

If a price list does not have a price for the item, the property accessor does the following:

1. Determines if the price list inherits a price for the item from another price list. If so, the accessor outputs a meta tag using the inherited price.
2. If the price list does not inherit a price, no tag is output for that price list.





8 Managing Search Environments

A search environment designates a set of one or more machines as being responsible for one of the following tasks:

- Indexing your content. Each search project has a single environment for indexing, which can be distributed across multiple hosts.
- Answering end-user queries using the deployed index. A project can have any number of environments for this purpose. These environments should be hosted on a machine optimized for high availability and performance.

Each search project automatically includes one indexing environment and one answering environment (called “default”) on the local host. You can create additional environments to add more resources to these tasks.

Each environment has at least one associated host, each of which has a number of search engines specified for it. When the environment is in use, it will run up to the specified number of search engines to complete its task. ATG Search automatically distributes indexes to the hosts, creating as many instances as needed. See the [ATG Search Installation and Configuration Guide](#) for more information about search engines.

This chapter includes the following topics:

[Creating Search Environments](#)

[Configuring Content Adapters for New Search Environments](#)

[Attaching Environments to Search Projects](#)

[Starting and Stopping Search Environments](#)

[Viewing Search Environments](#)

[Moving Search Environments Between Projects](#)

[Deleting Search Environments](#)

Creating Search Environments

A new search project automatically includes an indexing environment on the local machine, which is required to index the project. You can add host resources to this environment to improve indexing performance, but you never need to create a new indexing environment, since the project requires only one and creates it for you.



ATG Search also includes a default answering environment on the local machine. Indexes are copied to the answer environments in order to serve search results to end-users. The default local environment is for testing purposes; you should add resources to this environment in order to make sure it will support production use. You may also want to create additional answering environments; you may want to deploy your index to a test location before deploying it to production.

Note: The default RMI port for new environments is 8860. If this port cannot be used, then when you create an environment in Search Administration, Search Administration cannot connect to it until you change that setting. To change the default port:

1. On the Projects tab, go to Projects > *Your Project* > Environments > *Host Name* > Advanced Settings.
2. Change the Launcher Service Port.
3. Click Save.

You can create environments directly from a search project, or create environments and then attach them to a project.

To create a search environment:

1. To create an environment within a project, On the Projects tab, go to Projects > *Your Project* > Environments and click Add Environment.

To create an environment that is not attached to a project, on the Projects tab, go to Environments and click New Search Environment. You can attach this environment to a project later on, if a project needs more resources.

Note: Each indexing environment requires two ports per content set to be indexed.

Environments represent resources for indexing, testing, and deploying indexes.

Select an option:

Add an existing, unassociated Environment:

Create a new Environment

★ Environment Name:

2. Enter the environment name. Environment names cannot include spaces.
3. Click Add.
4. Specify a host machine or machines. When you enter a host name, an additional row appears below it, so you can add more than one host to the environment.



Host Machines

Host Machines are the computers that run Search Engines for Environments. Each Environment has its own configuration for any given host machine. Once a host has been added, the table below will show how the server is being utilized by this environment and by all other environments.

*Host Machine Name	Restrict to Specific Content Set	* Search Engines for this Environment	CPUs on Machine	Total Running Engines	Total Search Engines Across All Envs	Current Utilization	Potential Utilization	
rstevenson-T60	<Select Content Set(s)>	1	2	1	2	Under	Full	
	<Select Content Set(s)>	1			[New]			

Enter the host name as it appears on your network.

- Specify the content sets for which this environment should serve answers. This setting is optional; if not set, the environment is used for all content sets.

If you explicitly link a particular content set with an environment, other environments will not serve answers for that content set.

Select Content Sets

Content Sets

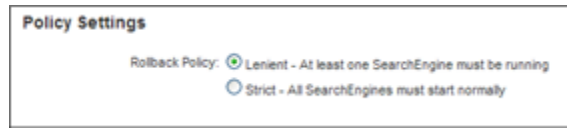
Content Set 1

OK Cancel

- Specify the maximum number of search engines the host can use. When you save the new environment, Search Administration uses this information to determine whether the new host has sufficient resources to support those engines. You can then update the number of CPUs on that host if the default (1) is not correct.

- Under Policy Settings, do **not** select a rollback policy.

Note: This feature has been deprecated and will be removed in a future release.



8. Click Save.
9. To specify Advanced Settings, click the new environment’s Host Machine Name, then click the Show link.

Settings include:

- Base Search Engine Working Directory—Specifies the base directory for index files for the machine’s search engines. Engines will create their own subdirectories automatically. This must be a directory on a local (non-network) drive; using a network drive can result in poor performance. For example:
`Configuration.default.rootIndexPath=d:\indexfiles`
- Port number(s) to use for communication with search engine instances.
- Launcher Service Port—This is the RMI port for the launcher service to use when starting search engines.

The Machine Information section includes CPU count and operating system information for troubleshooting purposes. If this information is blank, either the machine is not correctly configured, the launching service is not running, or the launching service is running on a different port.

10. Click Save.

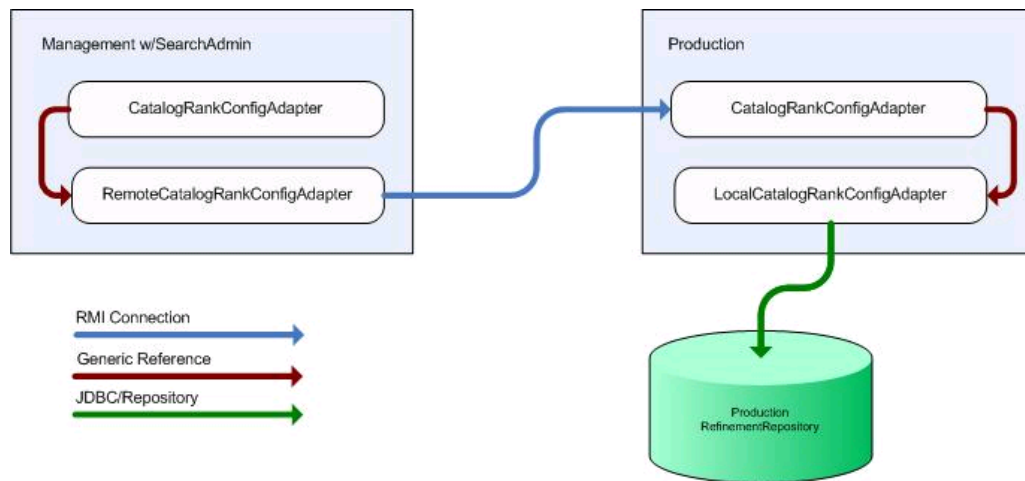
Configuring Content Adapters for New Search Environments

If your content includes Facet Sets or Auxiliary data, and requires the use of content adapters, then you must perform the configuration steps described in this section when you create a new search environment. The default adapters supplied by ATG support a maximum of two environments, which are assumed to be your production and staging environments. This section describes the procedures for adding a third environment, called test in the examples.

The adapter component names and their roles can be somewhat confusing. Each adapter component exists as a pair; these component pairs exist on both asset management and the production ATG servers.

Each pair on the asset management server consists of a remote and a standard version, for instance `CatalogRefineConfigAdapter` (standard) and `RemoteCatalogRefineConfigAdapter` (remote). The standard version of each component includes a single property, which is set to the path of the remote version of that component.

The production ATG server includes a standard and a local version of each of these same components. The following diagram shows the `CatalogRankConfigAdapter` components and their relationships as an example:



Creating Content Adapters

When you add a new environment type to your installation, you must create content adapter components for it. To do this, duplicate the following three component pairs on the asset management server (where Search Administration runs):

- /atg/commerce/search/config/CatalogRankConfigAdapter
/atg/commerce/search/config/RemoteCatalogRankConfigAdapter
- /atg/commerce/search/config/SearchUpdateAdapter
/atg/commerce/search/config/RemoteSearchUpdateAdapter
- /atg/commerce/search/refinement/CatalogRefinementConfigAdapter
/atg/commerce/search/refinement/RemoteCatalogRefinementConfigAdapter

You can find components by searching in the Component Browser of the ATG Dynamo Server Admin (see the [ATG Installation and Configuration Guide](#) for information).

Place the copies in your asset management server's local config directory, and give them names that reflect the new environment type. In this example, the components have `_test` appended to the name. When you create the copy, make sure that your new component is complete and includes the class and scope information.

Then configure the `componentPath` property of each standard component to point to the remote version of each component. For example, configure the new `CatalogRankConfigAdapter_test` to point to the new `RemoteCatalogRankConfigAdapter_test`:

```
componentPath=RemoteCatalogRankConfigAdapter_test
```

The remote components each include a number of properties; for this purpose, the important ones are `remotePath`, `remoteHost`, and `*path`. Configure these properties to point to the component running on the production server (which in this case is a test environment you are adding). In `RemoteCatalogRankConfigAdapter_test`, configure these properties:

```
# /atg/commerce/search/config/RemoteCatalogRankConfigAdapter_test.properties
...
remoteHost=testhost.company.com
remotePort=8860
# path to the customization data adapter on the remote test host
remoteCustomizationDataAdapterNucleusPath=\
/atg/commerce/search/config/CatalogRankConfigAdapter_test
```

The path in this case is the path on the remote machine (in the component name, Remote indicates that the component points to a remote component; in the property name, remote indicates the pointer to that remote component).

Adding New Adapters to Adapter Registry

In order to become visible to Search Administration, each new adapter component must be added to the nameToNucleusPath Map property of the /atg/search/adapter/customization/CustomizationAdapterRegistry component.

```
nameToNucleusPath+=\
QA Rank Configuration
Adapter=/atg/commerce/search/config/RemoteCatalogRankConfigAdapter_test
```

The Map key is the name of the adapter as you would like it to appear in Search Administration.

Warning: Once you start your ATG instance, the name is imported into the database and cannot be changed through the configuration file. To fix naming errors, use the ATG Control Center to delete the information from your repository, then redo it (see the [ATG Installation and Configuration Guide](#) for information on the ACC).

Attaching Environments to Search Projects

If you have an environment that was created outside of a project, you can add it to a search project. An environment can be associated with only one project.

To attach the environment to a project:

1. On the Projects tab, go to Projects > *Your Project* > Environments.
2. Click Add Environment.
3. Select Add an Existing, Unassociated Environment.
If this option is grayed out, you have no unassociated environments.
4. Select the environment name.
5. Click Add.



Starting and Stopping Search Environments and Engines

To start or stop a search environment is to start or stop the search engines on all hosts that are part of that environment. If some engines in an environment are running and others are not, stopping the environment stops all running engines, and starting the environment starts all currently stopped engines.

To start or stop a search environment:

1. On the Projects tab, go to Projects > *Your Project* > Environments.
2. Click the Start or Stop link for the environment.

Environment ^	Hosts	Status		
indexing environment 0 (MyProject)	1	Stopped		
ProductionSearchEngine	1	Running	Stop	

In the Component Browser of the ATG Dynamo Server Admin, you can use the `/atg/search/routing/RoutingSystemService` page to stop and start individual engines. You can use the `/atg/search/routing/SearchEngineService` page to stop and start engines by environment (all engines on a specific machine belonging to a specific environment) or by machine (all engines on a machine, from any environment).

Name	Env Type	Parent Project	Index	Deploying Index	Host Count	Engine Count	Running Engine Count	Operations	Cancel Indexing
indexing environment	indexing	CRS Search Project	100007	null	1	1	0	start stop cycle delete	n/a
commerce	production	CRS Search Project	100007	null	1	1	1	start stop cycle delete	n/a

Viewing Search Environments

To view environments associated with a particular search project, in the Projects tab, go to Projects > *Your Project* > Environments.

To view all environments, in the Projects tab, go to Environments. The page lists unattached search environments, then attached environments by search project.



Environments				
New Search Environment				
Unattached Environments				
Environment		Hosts	Status	
(None)				
Search Project: CRS Search Project [Edit Search Project] [Manage Environments]				
Environment	Primary	Hosts	Status	
commerce	⊕	1	Running	
indexing environment		1	Stopped	
			Stop	

For each environment, it shows the number of hosts used for that environment, the environment priority, and the environment status. If you have multiple environments for a project (which is not the case for most installations), the primary environment is the one used for search requests.

Click an individual environment name to view detailed information for that environment, its hosts, and the content on that environment. Click the View by Host Machine and View by Content Set tabs to see information about environment use.


The Search Engines for this Environment column on the Environment’s Host Machines page determines the maximum number of engines that environment will use on a machine. Use a value greater than 1 only if your index has more than one partition.

Note: If you are using parallel indexing (see [Configuring Indexing for Large Content Sets](#) in the [Search Performance Tuning](#) section), keep in mind that the number of indexing engines for each machine will not exceed the Search Engines for this Environment value.

Moving Search Environments Between Projects

Once an environment is associated with a search project, you can remove it from that project and associate it with a different project.

To move an environment:


1. On the Projects tab, go to Projects > *Your Project* > Environments.
2. Click the remove icon () for the environment you want to remove.
3. In the dialog box, select Detach Environment.
4. Click OK.
5. On the Projects tab, go to Projects > *Project* > Environments.
6. Click Add Environment.
7. Select Add an Existing, Unassociated Environment.
8. Select the unattached environment’s name.



9. Click Add.


Deleting Search Environments

To delete a search environment that is attached to a project:

1. On the Projects tab, go to Projects > *Your Project* > environments.
2. Click the remove () icon for the environment you want to delete.
3. In the dialog box, select Delete Environment.
4. Click OK.

Note that you cannot delete an environment that is currently designated as primary; you must first make a different environment primary, then delete the first environment.

To delete an unattached search environment:

1. On the Projects tab, go to Environments.
2. Click the remove () icon for the environment you want to delete.
3. Click OK.

Note: Information relating to deleted environments may still appear in search reports; however, the only identifying information for a deleted environment is the repository ID.





9 Topic Sets

Topics can be associated with content in your index to help users locate information. Topics are created in hierarchies called topic sets. For example, you might provide the following topics sets to help customers locate information:

```
Store Information
  Northeast Stores
    Store A
      Store Hours
    Directions to Store
      Store B
```

Rule patterns and macros allow you to automatically associate your content with topics.

Note: If you make changes to topic sets, you must reindex your search project before the topic changes are applied to your content.

This chapter includes the following sections:

- [Adding Topic Sets](#)
- [Adding Topic Sets to Projects](#)
- [Modifying Topic Sets](#)
- [Exporting Topic Sets](#)
- [Adding Topics](#)
- [Moving Topics](#)
- [Renaming Topic Sets and Topics](#)
- [Deleting Topic Sets and Topics](#)
- [Automatically Associating Content with Topics](#)
- [Using Macros in Patterns](#)

Adding Topic Sets

There are several ways in which you can add topic sets to ATG Search. This section describes the steps to perform for each method.



Creating New Topic Sets in the Workbench

To create a topic set:

1. On the Search Workbench tab, select Topic Sets from the Show menu.
2. Click New Topic Set.
3. Enter a name (required) and description (optional) for the new topic set.
4. Select a default language.

The default language is used in several ways. Rule patterns in topic sets can individually specify a language to which they apply, in which case the default language is used for patterns that do not individually specify a language.

For integrations with ATG Knowledge, the default language is used when browsing topics by language, and when authoring solutions, to determine which topics to display, based on the solution language.

5. Click one of these buttons:
 - Create to create the new topic set and return to the Topic Sets page.
 - Create and Add Topics to create the new topic set and move to the Add Topics page.

See [Adding Topics](#) in this chapter for further instructions.

Importing Topic Sets as XML Files

To import a pre-existing XML file (see [Exporting Topic Sets](#)) that defines a topic set:

1. On the Search Workbench tab, select Topic Sets from the Show menu.
2. Click Import Topic Set.
3. Browse to the XML file containing the topic set definition.
4. Specify whether or not macro information contained in the file should be imported. If imported, you can select whether the imported macros overwrite any existing global macros or are ignored.
5. Specify whether you want to preserve existing topic IDs, or create new IDs for the imported topics.

If you are importing topics that were exported from ATG Knowledge, use the Preserve imported topic IDs setting. ATG Knowledge solutions refer to these IDs, and if they are not preserved, that reference is broken.

If you are migrating topics from Enterprise Search 4.0, you should generate new topic IDs.

6. Provide a name and description to use for the imported topic set in ATG Search.
7. Click Import.



Adding Topic Sets to Projects

To add a topic set to a search project, you can use either Project Administration as described in [Adding Customizations to Content](#), or the Workbench.

1. On the Search Workbench tab, select Topic Sets from the Show menu.
2. In the Topic Sets menu, select Topic Sets > *Your Topic Set*.
3. Click the Associate Search Projects tab.
4. Click the checkbox next to the project name.
5. Click Save.


Modifying Topic Sets

To modify a topic set and its topics:

1. On the Search Workbench tab, select Topic Sets from the Show menu.
2. In the Topic Sets menu, select Topic Sets > *Your Topic Set*.
3. Make your changes.
4. Click Save.

Exporting Topic Sets

You can export topic sets to an XML file to use that same topic set in a different ATG Search installation, or for troubleshooting purposes. To export a topic set:

1. On the Search Workbench tab, select Topic Sets from the Show menu.
2. On the Topic Sets page, click the Export button () for the topic set you want to export.
3. Check the Global Pattern Macros box to export macro information with the topic set.
4. Click Export. The export uses your browser's file download defaults.

Adding Topics

You can add topics to topic sets or to topics. You can nest topics to any depth.

To add topics:

1. On the Search Workbench tab, select Topic Sets from the Show menu.



2. In the Topic Sets menu, select Topic Sets > *Your Topic Set*.
3. Click Add New Topics to add a topic at the current level of the topic hierarchy, or select from the Go to Topic list to add child topics to an existing topic.
4. Enter the following information for the new topic(s):
 - Name

Note: Topic names must be unique across all topic sets. You cannot use the same topic name in different topic sets.
 - Display Name
 - Description

Additional rows are added automatically as you add more topics.
5. Click Create Topics.

Moving Topics

To automatically order topics alphabetically:

Right-click the parent topic and select Sort Child Topics Alphabetically.

To manually move topics within the topic hierarchy:

1. On the Search Workbench tab, select Topic Sets from the Show menu.
2. In the Topic Sets menu, select Topic Sets > *Your Topic Set*.
3. Right-click the topic and select from the menu to copy or cut it.



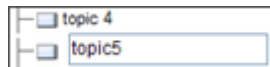
4. Right-click the topic under which you want to paste the moved topic. You cannot paste a subtopic back into the topic from which it was cut or copied.



Renaming Topic Sets and Topics

To rename a topic set or topic:

1. On the Search Workbench tab, select Topic Sets from the Show menu.
2. In the Topic Sets menu, navigate to the topic set or topic you want to rename.
3. Right-click the topic set or topic, and select Rename from the menu.
4. Enter the new name in the text field.



Click anywhere outside the text field when finished.

Deleting Topic Sets and Topics

To delete a topic set:

1. On the Search Workbench tab, select Topic Sets from the Show menu.
2. Click the delete icon (✖) for that topic set.
3. Click OK to confirm the deletion. The dialog alerts you if there are search projects that use the topic set being deleted.

To delete a single topic:

1. On the Search Workbench tab, select Topic Sets from the Show menu.
2. In the Topic Sets menu, select Topic Sets > *Your Topic Set*.
3. Right-click the topic, and select Delete from the menu.



4. Click Delete to confirm.

Automatically Associating Content with Topics

Content can be automatically associated with topics while the content is being indexed. To do this, you create rules, which contain patterns. When the content is indexed, ATG Search looks for those patterns in the content and computes a score, using the weights associated with the matching patterns. ATG Search determines whether a content item should be associated with a topic based on that score.

Adding Rule Patterns

The most common type of rule is a text pattern rule, which consists of a text string written in the syntax of ATG Search's text pattern language (see the [Understanding Rule Pattern Syntax](#) section of this chapter).

To add a rule pattern to a topic:

1. On the Search Workbench tab, select Topic Sets from the Show menu.
2. In the Topic Sets menu, select Topic Sets > *Your Topic Set* > *Your Topic*.
3. Click the Rules tab.
4. Enter the following information for each rule.
 - Pattern—Text string representing the text pattern used for matching.
 - Weight—Integer value indicating the weight value associated with a match against this pattern. This value may range between -100 and 100.
 - Group—You can optionally assign the rule to one or more comma separated groups, known as "Rule Groups". The groups are defined by virtue of being entered into this field. Grouped rules can be enabled or disabled by group name in Text Processing Option sets, allowing rules to be targeted to specific content.
 - Language—Select an option from the languages available for your installation. The default value is either English, or the value selected for the Topic Set as a whole on the Topic Set page. This setting indicates the language in which terms in the pattern are to be interpreted. It also specifies the language of the content against which the pattern should be matched. For example, a pattern that is set



to French can only match against content that has been indexed as French. This feature allows a single topic set to categorize documents in several different languages.

Rules are applied in order from top to bottom. Use the arrow buttons to move rules up or down the list.

5. Click Save to save the rules and remain on this topic, or Save and Edit Next to add a rule to the next topic at this level.

The new rule has the ID “New” until it is saved. ATG Search automatically assigns an ID to the new rule.

Understanding Topic Weights

Each rule is assigned a weight value from -100 to 100. Use a negative weight value to lower the rating for topic assignment. Use a positive weight value to raise the rating for topic assignment. Use a weight value of zero (0) to preempt other rule matches.

For example, for the topic *union strike*, you might define the following rules (weights are in parentheses):

-
1. strike three (0)
 2. strike out (0)
 3. strike two (0)
 4. strike one (0)
 5. union-strike (100)
 6. strike (100)
 7. baseball (-100)
-

Any document containing the word *strike* matches rule 6; however these documents may or may not be about union strikes. Any document that contains any of the first four phrases also matches rule 6, but because rules 1—4 appear first, they preempt the rule 6 match. (For example, *strike out* matches rules 2 and 6. Since rule 2 comes first, it is selected and receives a weight value of 0. Rule 7 lowers the Topic rating by -100 for every mention of *baseball* in the document.

The actual assignment of topics to documents occurs during indexing. The options that control thresholds for this are in TPO sets (see the [Text Processing Option Sets](#) chapter of this guide for more information):

- Topic limit per doc—Controls the maximum number of topics to assign to an item; the default is 10.
- Topic relevance threshold
- Topic confidence threshold

The topic limit determines the maximum number of topics to assign to any content item; the default is 10. The relevance and confidence thresholds set thresholds for which the pattern weights must score the document above, in order for the topic to be considered a relevant topic for an item. The default values are 1 and 0, respectively.



Understanding Rule Pattern Syntax

The syntax patterns in the list that follows can be combined to form rules.

- **Stem**—(text)—An alphanumeric sequence of characters that matches any form of the stem. For example, the pattern bank matches the terms banks, banki ng, bank, banker, etc.
- **Quoted string**—" text"—A string delimited by quotation marks (") that matches its literal form. For example, the pattern "banks" only matches the term banks.
- **Percent quoted string**—(%text%)—A string delimited by percent signs (%) that matches any expansion of the stem, using the lexicon's data. Since ATG Search allows users to define their own terminology in a Dictionary, this expansion data can be controlled by the user. For example, the pattern %computer% might match the terms laptop, desktop, etc.
- **Wildcard**—(* text)—A string containing one or more asterisks (*). This pattern matches any stem that matches the wildcard pattern. For example, the pattern *computer would match the terms supercomputer, mi ni computer, computer, etc.
- **Macro**—(\$macro_name)—The name of a macro beginning with a dollar sign (\$). This pattern matches any of the stems listed in the macro definition. For example, the pattern \$bank would match any term in the definition such as bank, i nsti tuti on, credi t uni on, savi ngs bank, etc.
- **Meta tag**—(<meta name="property" content="value">)—A <meta> tag embedded in an indexed document. For example, the pattern <meta name="product" content="eServer"> would match any indexed document containing this <meta> tag. Meta tag rules have no text region so they can not compete against any other rules.

The properties created are controlled by the Text Processing Option set. Only document metatags that have been configured with these options are used to match these special topic patterns.

- **URL**—("/URL")—A full or partial Internet address that points to a document or document directory. The URL can be a full URL or any relative path using the following four forms:
 - http://www.mycorp.com/company/support/help.htm is a full URL and only matches itself.
 - http://www.mycorp.com/company/support is a URL directory that matches any document contained within that directory or its subdirectories.
 - http://www.mycorp.com/company/support/* is a URL directory that matches any document contained within that directory but NOT its subdirectories.
 - /support/help.htm is a relative URL, which matches any document whose URL ends with string support/help.htm.
- **Docset**—A document set name. Sometimes the physical layout of indexed content suggests an appropriate way to categorize the content; for example, content from a specific subdirectory of a file system might map to a topic. ATG Search provides syntax



for specifying document set patterns that should map to a topic. The syntax is similar to the URL rules. Note that this rule pattern cannot specify a single index item, only a set of them. The syntax requires a quoted string beginning with a forward slash, and has three forms:

- `"/di r"` – An absolute document set path, such as `"/Documents/www. mycorp. com/product/hel p"`. Matches any index item contained within the document set, either directly or within child sets.
- `"/rel "` – A relative document set path, such as `"/product/hel p"`. Matches all index items whose physical document set ends with the relative path.
- `"/di r/*"` – An absolute path to a document set followed by an asterisk, such as `"/Documents/www. mycorp. com/product/*"`. Matches any index item that is immediately under the document set path, excluding those within child sets.

Example Patterns for Topic Rules

This section illustrates how to use patterns when creating topic rules.

Example 1

The simplest pattern consists of a sequence of terms, such as `bank loan fee`. The default interpretation of this pattern is that the terms may occur in any order, but they must be next to each other. This essentially forms a Boolean AND expression. Since the adjacency constraint is pairwise between terms, this pattern is interpreted as:

```
((bank AND loan WITHIN 1) AND (loan AND fee WITHIN 1))
```

Example 2

The adjacency constraint can be increased or eliminated by placing a dot operator (`.`) and a number in between the terms. Following the dot operator with an asterisk (`*`) eliminates the adjacency constraint. For example, the pattern `bank . 3 loan . * fee` is interpreted as the Boolean expression:

```
((bank AND loan WITHIN 3) AND (loan AND fee))
```

Example 3

To enforce order between two terms, insert a hyphen (`-`) between them. For example, the pattern `bank-loan . * fee` is interpreted as the Boolean expression:

```
((bank PRECEDES loan WITHIN 1) AND (loan AND fee))
```

Example 4

The pattern `%install% . * %problem%` matches input with `install` (or expansions) and `problem` (or expansions).

**Example 5**

The pattern "error"-57 matches input with the literal string "error" preceding the number 57.

Example 6

The pattern %myproduct% matches input with any expansion of a client's product line called myproduct.

Enabling and Disabling Rule Patterns


Disabled rules are not used during topic assignment.

To enable or disable a rule pattern:

1. On the Search Workbench tab, select Topic Sets from the Show menu.
2. In the Topic Sets menu, select Topic Sets > *Your Topic Set* > *Your Topic*.
3. Click the Rules tab.
4. Check or uncheck the rule to enable or disable it.

Deleting Rules

To delete a rule:

1. On the Search Workbench tab, select Topic Sets from the Show menu.
2. In the Topic Sets menu, select Topic Sets > *Your Topic Set* > *Your Topic*.
3. Click the Rules tab.
4. Click the delete icon () for that rule.
5. Click Save.

Using Macros in Patterns

Macros allow complex and/or frequently used text pattern strings to be defined and referenced by a simple name in a pattern rule. A macro named myMacro can be referenced in a text pattern by the string \$myMacro. The macro is expanded within the text pattern to its defining string.

Macros can be defined at the topic level or the global level.

Adding Global Macros

To add global macros:

1. On the Search Workbench tab, select Topic Sets from the Show menu.
2. Click the Edit Macros link.



3. Enter the macro name and definition. New lines appear below when you enter this information, so you can add additional macros.
4. Click Save.

Adding Local Macros

To add a pattern macro to a rule:

1. On the Search Workbench tab, select Topic Sets from the Show menu.
2. In the Topic Sets menu, select Topic Sets > *Your Topic Set* > *Your Topic*.
3. Click the Rules tab.
4. Click the Local link.



5. Enter the macro name and definition. New lines appear below when you enter this information, so you can add additional macros.
6. Click Save.

Deleting Macros

To delete macros:

1. On the Search Workbench tab, select Topic Sets from the Show menu.
2. In the Topic Sets menu, select Topic Sets > *Your Topic Set* > *Your Topic*.
3. Click the Rules tab.
4. Click the Local link.



5. Click the delete icon (✖) for the macro you want to delete.



6. Click Save.



10 Dictionaries

ATG Search includes a general-purpose dictionary. However, this dictionary may not include some terms that you want to index, such as product names, company names, or industry-specific terminology. Dictionary inspection tools allow you to examine the default dictionary, and term dictionaries allow you to add new terms to ATG Search. By doing so, you can significantly enhance and fine-tune your users' search results. You can organize these terms into your own dictionaries for easier management.

As part of adding terms to a dictionary, you can also set *term weights*. Term weights affect how ATG Search decides which documents are relevant to a user's search. See the [Understanding Term Weights](#) section for information and important cautions.

Note: If a dictionary is changed, all indexes that use that dictionary must be reindexed.

This chapter includes the following sections:

[Why Use Dictionaries](#)

[Using Dictionary Inspection Tools](#)

[Managing Dictionaries](#)

[Managing Terms](#)

[Managing Synonyms](#)

[Understanding Term Weights](#)

[Managing Term Weight Sets](#)

Why Use Dictionaries

This section addresses some common questions about search dictionaries and offers suggestions for situations when you should use them—and when you should *not* use them.

Before adding a custom term to a dictionary, search your existing index for the term and its potential synonyms or alternate spellings. If the search results are satisfactory, you do not need to add the term. If you do add custom terms, keep a list of terms you have added. Whenever you make further changes to the dictionary, search on the previously added terms to make sure that the results are still correct. Always test your changes thoroughly before deploying your index.



Why Create a Dictionary

You should create a custom dictionary if you need to add terms to the core search dictionary. You should create multiple dictionaries if your added terms need to be administered separately, or applied differently to different indexes.

For example, your company sells both construction materials and heavy equipment. You maintain two separate indexes for customer searches, and a third, combined index, for internal use. You would create two term dictionaries, one for material terms and one for equipment terms. Each customer-facing index would include the related dictionary, and the combined index would include both.

Why Add Terms

There are three main reasons to add custom terms to the search dictionary:

- The term is not in the default dictionary, but does appear in your searchable content or in user queries. Examples include product names and industry-specific terms.
- You want to add a synonym that does not exist in the default dictionary to an existing dictionary term. These might include product or brand names that are often abbreviated or referred to by acronyms. To do this, add the term to a custom dictionary and then add the synonym. (See also, [Why Add Synonyms and Child Terms](#))
- You want to override the default synonyms in the dictionary, either because you do not want to use those default synonyms, or because you want to add synonyms that apply to your content but are not in the default dictionary. To do this, add the term to a custom dictionary and then add synonyms if appropriate. (See also, [Why Add Synonyms and Child Terms](#))

When you are deciding whether you should add a custom term, consider the questions posed in the sections that follow.

Note: Before you add any terms to a dictionary, use Dictionary Inspection to make sure that you are not adding a term that is already in the core dictionary. Adding duplicate terms can cause unpredictable search results for your users, if you have not also specified the appropriate override behavior for the dictionary.

Does the Term Appear in Other Forms

If your term (such as a company name, brand name, or product name) is unlikely to be pluralized or to appear in more than one form, you do **not** need to add it to a dictionary. Although the term is recorded and reported on as “unknown,” that does not mean that the term is not searchable. A term searched on always matches that exact string in the indexed content, whether or not the term is in a dictionary.

If the term *does* appear in a plural or other variant form, then you should add it as a custom term. If the term is a single word, specify the part of speech when you add the term; ATG Search can then identify morphological variations such as the plural form or past tense automatically. If the term consists of multiple words that have potential variations, you will need to enter each variation as a synonym or spelling variation.



For example, if you add your product name *Widget* to a custom dictionary and specify that it is a noun, ATG Search will also recognize *Widgets*, without having to enter the plural form separately. If you add *Widget* as a literal, only that precise form is recognized.

Is the Term a Misspelling of an Existing Dictionary Term

If your reports suggest that users are searching with a misspelled version of a dictionary term, you may want to add the misspelled version of the term to a custom dictionary.

However, *before* you add the term, make sure that you try a search with the misspelling; ATG Search does include a spellchecking feature, which will attempt to correct the misspelled term. If ATG Search can determine the correct form of the term, the search will work, and you do not need to add the term.

Note, however, that if the misspelled version of the term actually *appears in the content*, the spellchecker assumes that it is a valid term, does not correct it, and the search will return *only* the misspelled items, because those match exactly the term the user entered.

If your tests with the misspelled search term do not achieve good results, add the correct version of the term to a dictionary as a custom term, and also add the misspelled version or versions as spelling variations of that term.

Is the Term Part of a Larger Phrase

If the term appears in your content only as part of a larger phrase, add the entire phrase as a custom term. Note that you only need to add the phrase if the term appears in other forms (such as a plural) or has synonyms.

For example, *cous cous* is often written as one word, *couscous*, so *cous cous* should be added and given a synonym of *couscous*.

Similarly, *Aloe vera* can be referred to simply as *aloe*. Phrases such as this require careful consideration. Before adding it as a custom term, compare the results of a search on *aloe vera* to a search on just *aloe*, and see if the results are consistent and satisfactory. Since searches on catalog data are AND searches by default, searches on *aloe vera* may not return good results if the content contains only *aloe*. If this is the case, you should add *aloe vera* as a custom term, with *aloe* as a synonym. There is no need to enter *vera* as a synonym, since *aloe vera* is never referred to as just *vera*.

Note that if you define a phrase (a term containing more than one word) in a custom dictionary, that phrase becomes an unbreakable unit and will no longer match any of its separate parts and will not take any prefixes or suffixes at all. For example, if you define *visa credit card* as a term, the phrase will not match *visa* or *credit card* or even *visa credit cards*. It will only match the exact string *visa credit card*. If you want any variations of the phrase to match, you have to define all of those variations as synonyms. It is best not to add phrases to a custom dictionary except as a fix for actual problems that come up in search.

Why Add Synonyms and Child Terms

When you add a term to the dictionary, you can add synonyms for that term. You can also add child terms, which can have synonyms of their own. These features increase the chance for your users to find information that is relevant to their interests.



There are two main situations in which you add synonyms. The first is when you are already adding a custom term, and that term has synonyms that should be included. For example, you might have a product or brand that is referred to by an abbreviation or acronym as well as by its full name (for example, *Art Technology Group* vs. *ATG*).

Alternatively, you may want to override synonyms that already exist in the default dictionary, either because you do not want to use those default synonyms, or because you want to add synonyms that apply to your content but are not in the default dictionary:

- If terms in the default dictionary have synonyms you do not want to use, you can remove them by adding the term in a custom dictionary and either giving it no synonyms at all, or only those synonyms you want to use.
- It may be that terms are synonyms in your searchable content but not in the default dictionary. For example, a web site that sells antivirus software might want the verbs *clean* and *remove* to be synonymous.

When you add a known term (a term that is already in the core dictionary) to the custom dictionary, you automatically overwrite any synonyms the term might have in the core dictionary; the synonyms are not erased, but your custom entry is used instead of the core entry. Therefore, if you want to add to the existing synonyms and preserve what is already there, you must enter each of the term's core synonyms into your custom dictionary.

An important difference between synonyms and child terms is that synonyms are two-directional; synonyms are generally expected to expand to each other. With child terms, however, the expansion is one-way. You can use *propagation* to specify whether the child term expands to the parent, or only to the parent term to the child.

One situation in which you might use parent and child terms instead of synonyms is a product that has multiple subtypes. For example, you might carry a brand of phone that has a variety of possible configurations, in which case the brand would be the parent term, and each of the models a child term attached to it. A search for the parent brand would return the child models, but a search for a specific model would not necessarily return the parent brand.

The example that follows shows the difference between synonym behavior and child term behavior. It assumes that you are propagating child terms to their parent. The important thing to keep in mind is that synonyms form a reciprocal relationship, whereas the parent-child relationship is one-directional.

For example: assuming you are propagating children to parents, create the following dictionary terms and synonyms:

- Parent Term: *Party Food*
 - Child Term: *Pizza* with reciprocal synonyms *Deep Dish* and *Thin-Crust*
 - Child Term: *Soda* with reciprocal synonyms *Pop* and *Cola*

With these relationships in place, a search for any particular term also returns the synonyms:

- *Pizza* > *Deep-Dish*, *Thin-Crust*
- *Deep-Dish* > *Pizza*, *Thin-Crust*
- *Thin-Crust*-> *Pizza*, *Deep-Dish*



- *Soda > Pop, Cola*
- *Pop > Soda, Cola*
- *Cola > Soda, Pop*

Additionally, a search for the parent term returns the child terms and all of their synonyms:

- *Party Food > Pizza, Deep-Dish, Thin-Crust, Soda, Pop, Cola*

Note, however, that the parent term only expands *down* to its children, and that *Cola* does not expand to *Party Food* or *Pizza*.

An alternate arrangement could involve only child terms, and no synonyms:

- Parent Term: *Party Food*
 - Child Term: *Pizza*
 - Child Term: *Deep-Dish*
 - Child Term: *Thin-Crust*
 - Child Term: *Soda*
 - Child Term: *Pop*
 - Child Term: *Cola*

With this relationship, a search for *Pizza* would return only *pizza*, but a search for *Party Food* would return all of the child terms:

- *Party Food > Pizza, Deep-Dish, Thin-Crust, Soda, Pop, Cola*

Whenever you add terms to your dictionary, think carefully about the relationships between the terms you are adding, how terms are used in your content, and how your users will phrase their queries.

Using Dictionary Inspection Tools

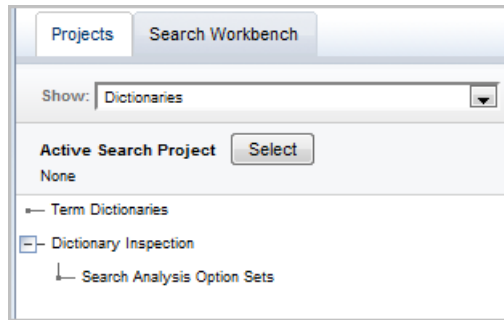
The Search Administration dictionary inspection page gives you a window into your search dictionary. It shows how the dictionary, custom terms, and text processing option sets affect content processing during indexing, and an analysis tool that shows how ATG Search interprets queries. These features allow you to see the effect of terms and term relationships in both core and custom dictionary terms. This can help prevent you from adding extraneous terms to your dictionary, or overriding existing terms unnecessarily, and is useful in term-related troubleshooting.

Looking Up Dictionary Terms

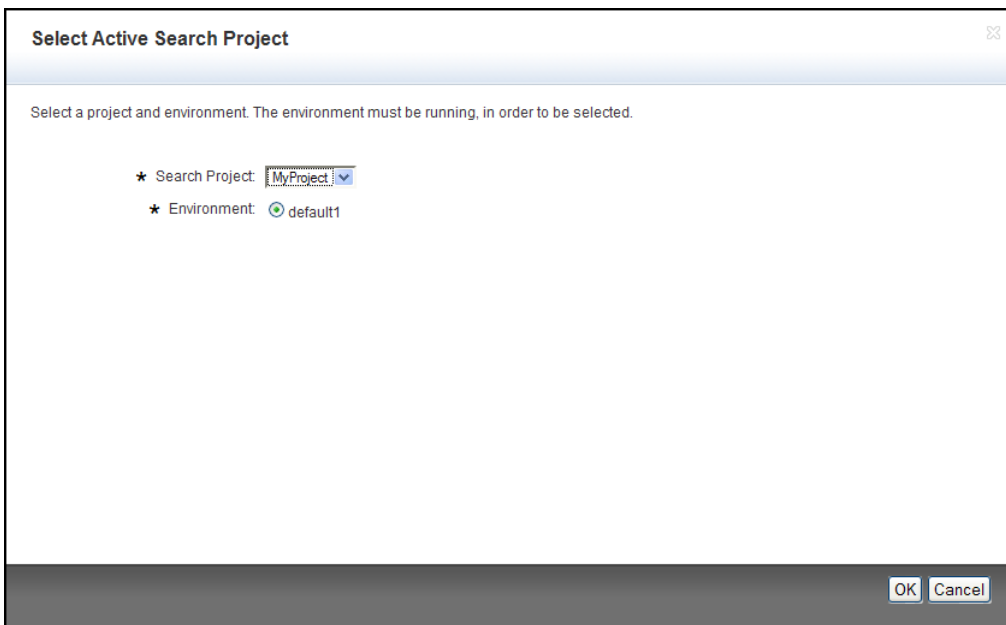
The Term Lookup page allows you to view information in the ATG Search core dictionary and any custom dictionaries you have added. You can select different dictionary adaptors and text processing option (TPO) sets to use for the lookup to see how these changes affect your dictionary data.

1. On the Search Workbench tab, select Dictionaries from the Show menu.

- Go to Dictionary Inspection.



- Click Select. In the Select Active Search Project dialog, select a project and environment.



- On the Dictionary Inspection page, click the Term Lookup tab.
- Enter a single term and click Lookup.

The results show comprehensive dictionary information for that term, including stem, language, part of speech, dictionaries it appears in, number of times a form of the word appears, and links to synonyms and other forms of the word.

For example, looking up *computer* in the base dictionary returns the following information:




Term: ⓘ

1 result for: computer
(3 near or partial matches)

Stem:
computer (0 content items; 0 total occurrences)

Language:
English

Part of speech:
noun


Term Source:
core 

Frequency of Forms:
--

Expansions and Synonyms:

Terms	Part of speech	Language	Adaptor
desktop	noun	English	
laptop	noun	English	
main frame	noun	English	
notebook	noun	English	
pc	noun	English	

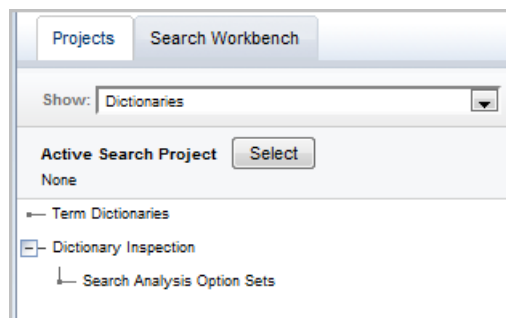
Near or partial matches in the dictionary:
compute
com
put

If the term displayed is not part of the core dictionary, you can copy it into a custom dictionary using the  icon. This provides you with a starting point from which you can modify the term for your custom use, rather than creating terms from scratch.

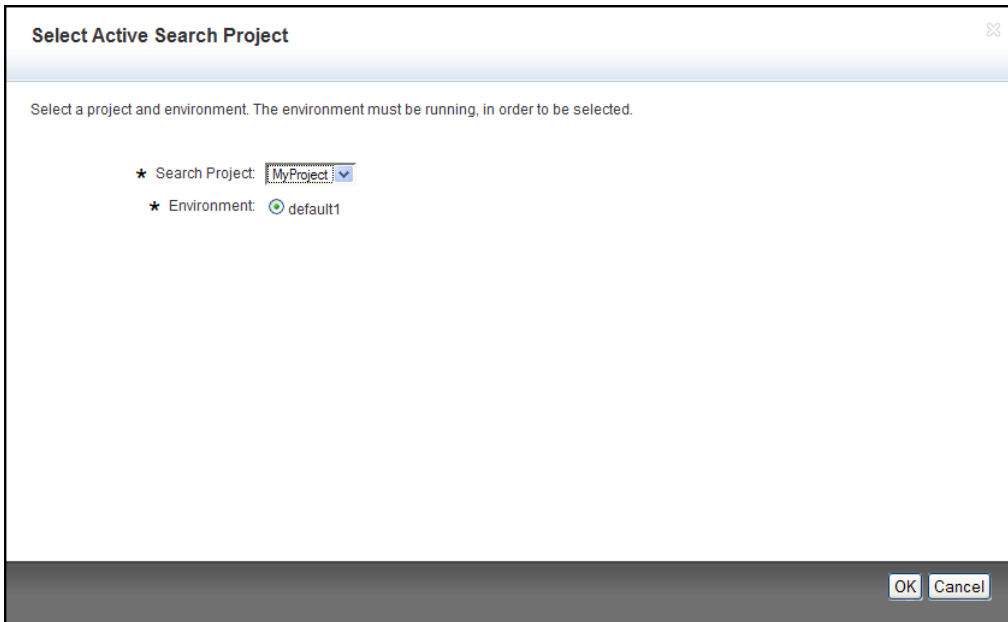
Evaluating Content Processing

The Text Processing page allows you enter a piece of text and see how ATG Search would analyze it for indexing. This shows how indexing options and TPO sets affect particular content items. To use text processing:

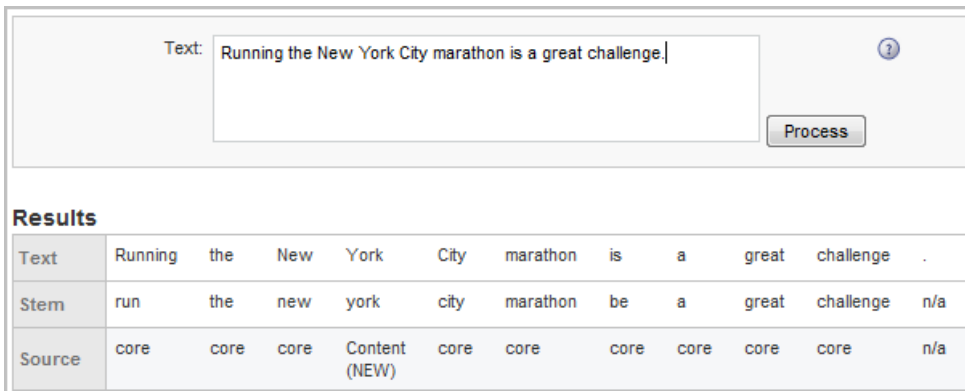
1. On the Search Workbench tab, select Dictionaries from the Show menu.
2. Go to Dictionary Inspection.



3. Click Select. In the Select Active Search Project dialog, select a project and environment.



4. On the Dictionary Inspection page, click the Text Processing tab.
5. On the Text Processing tab, select the content-level TPO set to apply, if any.
6. Enter a piece of text that resembles the sort of content in your search index, such as a product description or a piece of "how-to" information that is part of your site.
7. Click Process.



For each word in the text, the page displays the following:

- The word entered.
- The stem or root that ATG Search determined for the word.

An important feature of ATG Search is the ability to determine root forms of words through complex analysis, rather than merely performing a literal text match. For example, if content items contain the terms *selling*, *sells*, and *sell*, and



the query contains *selling*, a literal match would only retrieve documents that contain *selling*, potentially missing relevant results. ATG Search, however, can retrieve documents that include any form of *sell* based on a query that contains any form of *sell*. This can provide your users with better search results.

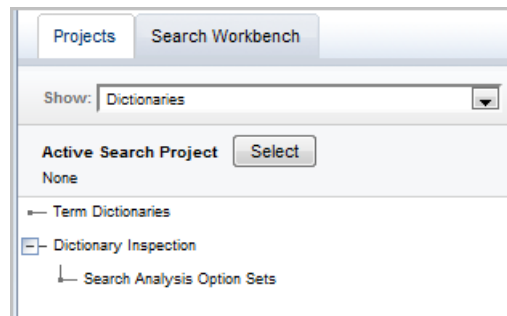
- The source, which is either a dictionary or content you have indexed. If the term exists in neither the content nor any loaded dictionaries, that word is marked “Content (NEW)” in the processing results.

Analyzing Search Text

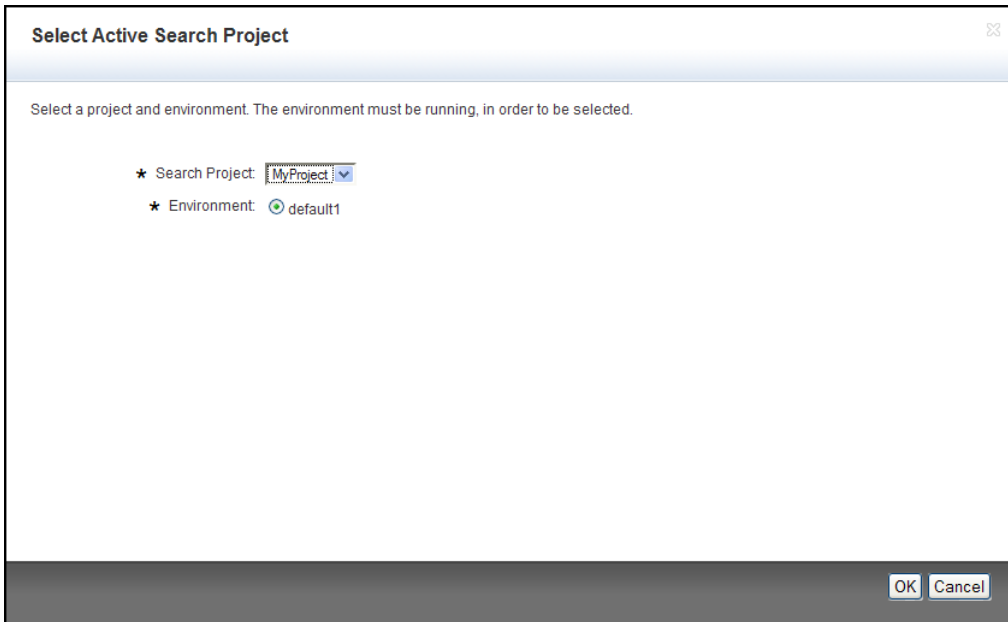
While the Text Processing tab shows how indexed content is analyzed, the Search Text Analysis tab shows how query text is processed by the search engine. To do this, it uses Search Analysis Option Sets, which are analogous to the settings that either appear in your Web site’s search user interface or are configured in your Web site’s form handlers. This allows you to experiment with how these settings affect your users’ searches without having to change your site. It is important to understand that these settings should be configured the same as they are on your Web site if you want to troubleshoot a search issue that you have observed on your Web site. Inspection allows you to vary settings in a controlled, non-production environment in order to determine how best to solve problems or make improvements to your site.

To analyze search text:

1. On the Search Workbench tab, select Dictionaries from the Show menu.
2. Go to Dictionary Inspection.



3. Click Select. In the Select Active Search Project dialog, select a project and environment.



4. On the Dictionary Inspection page, click the Search Text Analysis tab.
5. On the Search Text Analysis tab, select the SAO set to apply, if any. If you have not created any, ATG Search uses the default settings.
6. Enter text for analysis, just as you would if using the search user interface on your Web site, then click Analyze.

You can use wildcards in your search text, and the following operators:

Operator Example	Parsed Text	Operator Meaning
++headlight	headlight	required (in document)
!!compact	compact	not (in document)
+ ford + chrysler <i>brakes</i>	ford Chrysler	OR required
+headlight	headlight	required (in statement)
!compact	compact	not (in statement)
"NYC"	NYC	Exact match



Operator Example	Parsed Text	Operator Meaning
'GM'	GM	Case match upper
'kleenex'	kleenex	Case match lower
'Santa Claus'	Santa Claus	Case match mixed

An example result follows:

Text:

Analyze

Results

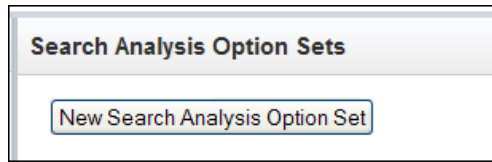
Text	MyCorp	widget	store	locations
Boolean Restriction: Item*	required	required	required	required
Index Term (Stem)		widget	store	locate
Term Source		core	core	core
Term Weight	100	100	95	95
Number of index items containing term	0	0	208	0
Spelling Suggestions				
Term Expansions		appliance contraption control gadget	mart shop storehouse	find site situate
Term Expansion source		core	core	core

* values are the result of the Interpretation Mode selected in the SAO Set

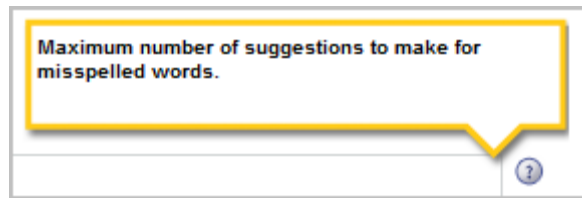
Creating Search Analysis Option Sets

To create a Search Analysis Option Set:

1. On the Search Workbench tab, select Dictionaries from the Show menu.
2. Go to Dictionary Inspection.
3. Under Dictionary Inspection, click Search Analysis Option Sets.
4. In the Search Analysis Option Sets page, click New Search Analysis Option Set.



5. Enter a name for the SAO set.
6. Set the SAO options. Details are provided in the online help; to view the description, place your cursor over the icon as shown:

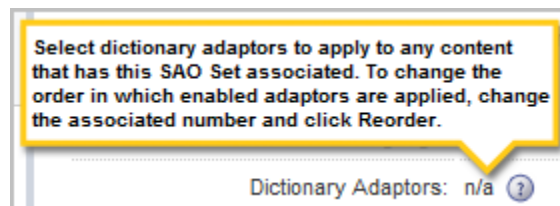


7. Click Save Options when finished.

Copying Search Analysis Option Sets

If you have only minor changes to make, it may be easier to copy and modify an existing SAO set than to create one. To copy a Search Analysis Option Set:

1. On the Search Workbench tab, select Dictionaries from the Show menu.
2. Go to Dictionary Inspection.
3. Under Dictionary Inspection, click Search Analysis Option Sets.
4. In the Search Analysis Option Sets page, click the copy icon (📄) for an existing SAO set.
5. Click the copy to edit the name and other details.
6. Set the SAO options. Details are provided in the online help; to view the description, place your cursor over the icon:




7. Click Save Options when finished.

Deleting Search Analysis Option Sets

To delete a Search Analysis Option Set:



1. On the Search Workbench tab, select Dictionaries from the Show menu.
2. Go to Dictionary Inspection.
3. Under Dictionary Inspection, click Search Analysis Option Sets.
4. In the Search Analysis Option Sets page, click the delete icon () for the set you want to delete.

Managing Dictionaries

This section describes how to work with entire dictionaries. The [Managing Terms](#) section which follows describes how to work with terms within dictionaries.

Adding Dictionaries

To add a dictionary:

1. On the Search Workbench tab, select Dictionaries from the Show menu.
2. Select Term Dictionaries.
3. Click New Dictionary.
4. Enter a name (required) and description (optional) for the new dictionary.
5. Enter a Dictionary Adaptor Name (optional). If used, when the dictionary is loaded, it is loaded under this adaptor name. Note that the dictionary name is called context by the Search API.
6. Select the dictionary's language.
7. For the Load Mode, select one of the following:
 - **Modify**—ATG Search assesses the terms and term relationships of your new dictionary and automatically attempts to correct any data that conflicts with the ATG Search core dictionary. For example, if you add the custom term *modems* using modify mode, ATG Search recognizes it as the plural of an existing term and modifies *modems* to *modem*.
 - **Override**—The terms and term relationships of your new dictionary override those in the ATG Search core dictionary. Using this mode, if you add the term *modems* as a custom term, ATG Search ignores the fact that *modems* is the plural of an existing term, and adds it to the custom dictionary. *Modems* becomes a separate term from *modem* and is no longer recognized as a form of the core term. Override mode is useful if you need to create a distinction between terms that mean different things in their singular and plural forms, such as window and Windows.
 - **Reject**—ATG Search rejects any data in your new dictionary that conflicts with the ATG Search core dictionary. For example, if you add *modems* as a custom term using reject mode, ATG Search recognizes it as the plural of an existing core term and ignores the entry.



Note: For data on rejected terms, enable logging as described in [Enabling Search Logging](#), and search the response XML data for the TermLoad command.

8. Enter the maximum term expansion.

When dictionary terms are organized hierarchically, you can optionally indicate that this hierarchy implies *term expansion*. In other words, if a child term is given a positive value for the propagate attribute, it means that the child term and its synonyms propagate up to become part of the term expansion of the parent term.

Set the maximum term expansion to 0 to disable term expansion. If you set this value to a number larger than the number of levels in your dictionary, all terms are fully expanded. Note that very large expansion sets can slow down search as well as worsen results.

9. Click Create.

To add a term, term association, or synonym to a dictionary, see [Managing Terms](#) in this chapter.

Editing Dictionaries

To edit a dictionary:

1. On the Search Workbench tab, select Dictionaries from the Show menu.
2. In the Dictionaries menu, select Term Dictionaries > *Your Dictionary*.
3. Make your changes.
4. Click Save.

Duplicating Dictionaries


To duplicate a dictionary:

1. On the Search Workbench tab, select Dictionaries from the Show menu.
2. In the Dictionaries menu, select Term Dictionaries > *Your Dictionary*.
3. Right-click the dictionary name and select Copy to New from the menu.

The dictionary and its terms are copied. The new dictionary is named Copy of *Name*.

Exporting Dictionary Data

You can export dictionaries to an XML file to use that same dictionary in a different ATG Search installation, or for troubleshooting purposes. To export a dictionary:

1. On the Search Workbench tab, select Dictionaries from the Show menu.
2. On the Dictionaries page, click the Export button () for the dictionary you want to export.
3. Click Export. The export uses your browser's file download defaults.



Importing Dictionary Data


You can create a dictionary by importing a properly formatted XML file into ATG Search. Use this feature to import dictionaries created in previous versions of ATG Search, or when moving your ATG Search installation.

To import a dictionary:

1. On the Search Workbench tab, select Dictionaries from the Show menu.
2. Click Import Dictionary.
3. Browse to the XML file that defines the dictionary you want to import.
4. Provide a name (required) and description (optional) for the new dictionary.
5. Click Import.

Deleting Dictionaries

To delete a dictionary:

1. On the Search Workbench tab, select Dictionaries from the Show menu.
2. Click the delete icon () for the dictionary.
3. Click Delete to confirm.

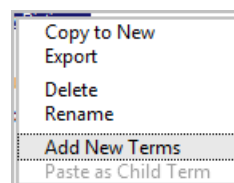
Managing Terms

In ATG Search, you can add terms and synonyms to a dictionary that you have added. Construct your own terminology hierarchy by adding root terms to your dictionary.

Adding Terms

To add a term to a dictionary:

1. On the Search Workbench tab, select Dictionaries from the Show menu.
2. Right-click the dictionary or term to which you want to add a term, and select Add New Term from the menu.



3. Enter the term.



Although ATG Search automatically identifies plural and other forms of custom terms that are single words, it does not do so for custom terms that consist of multiple words. You can add these other forms as synonyms, however (see step 6 in this procedure), to ensure that they are found when your users search.

For example, if you add the custom term “trade show,” ATG Search does not automatically expand searches to include “trade shows,” but will do so if you add “trade shows” as a synonym.

4. Select the part of speech. Use Literal Term for any terms you do not want to be subject to stemming and morphology, such as the name of people or places.
5. If the term is a phrase, select the type.
 - Literal phrase—The entered term is matched exactly.
 - Stemmed phrase—Terms within the phrase are stemmed if possible, providing the most liberal matching of the phrase.
 - Mixed phrase—The phrase includes a mixture of literal forms and stems already defined in the dictionary. Stem parts match any form of the stem, and literal forms only match that form exactly.
6. Click the Enable link if you want to assign a weight to the term, and enter the weight.

Warning: Only use this feature if you are very familiar with ATG Search. See [Managing Term Weight Sets](#) for information on weights.

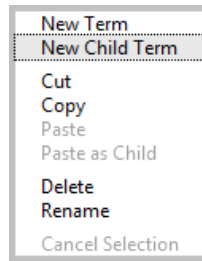
7. Enter synonym information. When you begin to type a synonym, a new line appears below it, so you can continue to add synonyms. See the [Adding Synonyms to Terms](#) section for details.
8. Click Create if you are finished adding terms. Click Create & Add Another to add additional terms at this level of the dictionary.

Note: If you are adding an acronym that could be confused with an existing word, type it with a period after each abbreviated letter. For example, the acronym FAST should be typed in as “F.A.S.T.” Otherwise, during indexing ATG Search will associate it with inflected forms such as *faster*, *fastest*. Acronyms without the potential for confusion, such as HTML, or FAQ, can be typed without periods.

Adding Child Terms

To add a child term to an existing term in the dictionary:

1. On the Search Workbench tab, select Dictionaries from the Show menu.
2. Right-click the dictionary or term to which you want to add a term, and select New Child Term from the menu.



3. Enter the term.

ATG Search does not identify plural or other forms of custom terms that consist of multiple words. You can add these other forms as synonyms, however (see step 6 in this procedure), to ensure that they are found when your users search.

For example, if you add the custom term “trade show,” ATG Search does not automatically expand searches to include “trade shows,” but will do so if you add “trade shows” as a synonym.

4. Select the part of speech. Use Literal Term for any terms where you want ATG Search to use the exact term entered rather than determining the root form. For example, you would probably use Literal Term for the names of people or places.
5. Check the Propagate box if you want searches for this term and its synonyms to return results for child terms.
6. Click the Enable link if you want to assign a weight to the term, and enter the weight. Note that if you assign a weight in the Dictionary, and also create a Term Weight Set that includes the custom term, the Term Weight Set overrides the Dictionary weight.

Warning: Only use this feature if you are very familiar with ATG Search. See [Managing Term Weight Sets](#) for information on weights.



7. Enter synonym information. When you begin to type a synonym, a new line appears below it, so you can continue to add synonyms. See the [Adding Synonyms to Terms](#) section for details.
8. Click Create if you are finished adding terms. Click Create & Add Another to add additional terms at this level of the dictionary.

Note: If you are adding an acronym that could be confused with an existing word, type it with a period after each abbreviated letter. For example, the acronym FAST should be typed in as “F.A.S.T.” Otherwise, during indexing ATG Search will associate it with inflected forms such as *faster*, *fastest*. Acronyms without the potential for confusion, such as HTML, or FAQ, can be typed without periods.

Modifying Terms

To change a term in a dictionary:



1. On the Search Workbench tab, select Dictionaries from the Show menu.
2. In the Dictionaries menu, select Term Dictionaries > *Your Dictionary*.
3. Make your changes.
4. Click Save.

Moving Terms

To move terms from one place to another:

1. On the Search Workbench tab, select Dictionaries from the Show menu.
2. In the Dictionaries menu, select Term Dictionaries > *Your Dictionary*.
3. Right-click the term to copy or cut it.
4. Right-click the place in the term hierarchy under which you want to paste the moved term. You can paste the moved term either at the same level as the selected term, or as a child of another term.

Deleting Terms

To delete a term in a dictionary:

1. On the Search Workbench tab, select Dictionaries from the Show menu.
2. In the Dictionaries menu, select Term Dictionaries > *Your Dictionary*.
3. Right-click the term.
4. Select Delete from the menu.

The term is deleted. If the deleted term had any child terms attached to it, they are deleted, as are any synonyms attached to the deleted term.

Managing Synonyms

Adding synonyms allows ATG Search to link terms, expanding its ability to provide relevant content items in response to user searches. For example, in our construction materials example, you might want a search for “glass” to be considered to be related to a search for “windows,” even though this behavior is not present in the default dictionary.

This section describes how to work with synonyms and add them to your dictionary terms.

Adding Synonyms to Terms

To add a synonym to a term:

1. On the Search Workbench tab, select Dictionaries from the Show menu.
2. In the Dictionaries menu, select Term Dictionaries > *Your Dictionary*.



3. Click a term in the dictionary to select it.
4. In the Synonyms text box, type the synonym you want to add. When you begin to type a synonym, a new line appears below it, so you can continue to add synonyms.

Synonyms					
*Synonym Term	Relationship	Part of Speech	Language	Weight	
Synonym1	Normal	Noun	English	Disable	
	Normal	Noun	English	Disable	

5. Select the relationship of the synonym to the base term:
 - Normal. Has a reciprocal relationship with the term. For example, if the term *flower* has a reciprocal synonym *blossom*, then any searches including the term *flower* would closely match answers about both flowers and blossoms, and vice-versa, though there would still be a slight preference for results containing the actual search term *flower*.
 - Spelling Variation. For searching purposes, the words are identical; if you search using either the term or its synonym, you will receive the exact same results. Use this relationship only if the term has no other meaning in your content.

Note: When a term is added as a spelling variation of another term, ATG Search converts the spelling variation into the head term. This occurs before any stemming has been done, so for spelling variation synonyms, you must enter all inflected or variant forms individually. For example, if you want all of the various forms of the term *hoodie* to be spelling variations of the head term *hoodie*, you must make separate equivalent synonyms for all forms that might be encountered in your content or entered in a search by an end-user, such as *hoody*, *hoodies*, and *hoodys*.
 - Related. Weakly associated to a term, for example as *hard drive* is related to *computer*.

Note: This relationship is used only for maintenance of previous ATG Search versions, and should not be used for new synonyms.
 - Non-reciprocal. A one-way only normal synonym association in the search between a term and the synonym. For example, if the term *color* has a non-reciprocal synonym *green*, then any searches including the term *color* would consider *green* as a synonym, but any searches including the term *green* would not consider *color* as a synonym in the search.
6. Select the part of speech. Use the Literal Term option for any term where you do not want morphological analysis performed, such as unique names, events, or places that might be confused with a similar inflected word form. For example, use Literal Term for names such as *Jane Flowers*, which could otherwise be matched with the word *flower*, or for *Microsoft Windows*, which could be matched with the word *window*.
7. If the term is a phrase, select the type.
 - Literal phrase—The entered term is matched exactly.



- Stemmed phrase—Terms within the phrase are stemmed if possible, providing the most liberal matching of the phrase.
 - Mixed phrase—The phrase includes a mixture of literal forms and stems already defined in the dictionary. Stem parts match any form of the stem, and literal forms only match that form exactly.
8. If you are linking a term to a synonym in a different language, in the Language drop-down list, select that language.

Synonyms can be defined for words in languages other than the language of the Dictionary. These cross-language synonyms allow queries in one language to retrieve results in another language. Such synonyms provide the bridge between terms in different languages. For example, you can link the term *car* in English to its equivalent synonym in French *voiture*. Further, you can link more complex terminology as synonyms, such as *Instructional Bulletin* with its French equivalent *Fiche de Directive*.

9. Enter a weight, if desired. If the main term does not have weight enabled, or if the synonym's relationship to the main term is Equivalent, weighting is disabled.

Warning: Only use this feature if you are very familiar with ATG Search. See [Managing Term Weight Sets](#) for information on weights.

10. Click Save.

Modifying Synonyms


To change a synonym in a dictionary:

1. On the Search Workbench tab, select Dictionaries from the Show menu.
2. In the Dictionaries menu, select Term Dictionaries > *Your Dictionary*.
3. Click the term to which the synonym is associated.
4. Edit the synonym.
5. Click Save.

The term record is updated in the dictionary.

Deleting Synonyms

To delete a synonym in a Dictionary:

1. On the Search Workbench tab, select Dictionaries from the Show menu.
2. In the Dictionaries menu, select Term Dictionaries > *Your Dictionary*.
3. Click the term to which the synonym is associated,
4. Click the delete icon () for that synonym.



Understanding Term Weights

Warning: Term weights represent an extremely powerful feature, and care should be taken when using them. Changing term weights can have unanticipated effects on end-user searches. Use this feature only if you fully understand the effects of changing term weights.

Normally, terms in an index are automatically weighted based on their distribution in the indexed content. The cumulative weight of the terms determines the relevance of a given piece of content to a specific end-user search.

On rare occasions, you may want to alter the automatic weighting in order to solve an issue with search result quality. When you change the weight of a term, you change the contribution it makes to the relevancy score of results in which that word appears. Adjust the weight upward, and answers containing that word are considered more relevant; adjust it downward, and those answers are less relevant.

Term weights are the mechanism for setting these specific weight values. Term weights are also a way to identify *stop words* or *noise words*, such as “the” and “or,” which generally do not contribute to answer relevance. The Term Dictionary user interface provides several places in which you can assign weights to dictionary terms and synonyms, however, note that weights that you define in Term Weight Sets override weights set in the dictionary.

When term weights are applied to content, one of the following takes place.

- If the weight specified for a term is less than the value specified in the `<indexStopThresh>` text processing option, which defaults to 0, that term is not indexed.
- If the weight specified for a term is equal to or greater than the value of `<indexStopThresh>`, but less than the value of the `<indexTermThresh>` text processing option, which defaults to 1, the term is saved for the purpose of scoring the candidate response, but the term itself is not indexed for the purpose of retrieval.
- If the weight specified for a term is greater than or equal to the value of `<indexTermThresh>`, then the term is indexed for the purpose of retrieval, and the weight of the term is as specified. The specified weight overrides the auto-weighted weight value for the term that would have been assigned to it through the content indexing process.

Note: See the online help for [Text Processing Option Sets](#) for information on these and other settings.

Because the system default value is 0 for `indexStopThresh`, there are by default no actual stop words configured in ATG Search. A few terms are weighted 0, and contribute no term weight score to the result rankings. ATG Search uses zero-weighted and low-weighted terms in this manner, rather than true stop words, to support phrase retrieval and literal matching on low-content strings (for example, “To be or not to be”).



Managing Term Weight Sets

Term Weight Sets provide a way to change weights in the core ATG Search dictionary, and can not be used for adding new terms. For custom dictionaries, you can set the weight directly on the term or synonym.

Only one weight set can be specified per search project per language. Changing Term Weight Set values requires re-processing the content in a clean index in order for the changes to have predictable results.

Note: Term weight changes are very rarely required. Do not use this feature if you do not understand its potential impact on end-user searches.

Default Term Weight Sets

ATG Search includes default term weight sets, which are loaded during indexing along with the project's main language. These defaults are included in your project, although they do not appear in the list of customizations. Any custom weight sets are loaded afterwards, and override the defaults.

To override the defaults, you can either create a custom term weight set that includes the term you want to override, or import the default term weights and edit them directly in the Workbench; the latter approach is strongly recommended over creating term weights from scratch. You can import a default term weight set into a new custom term weight set from:

```
<SearchDictionary>/SearchEngine/Language/Language_name/TermWeights.xml.
```

Note that after you have defined your custom term weight set, you must select it on the Customization page in order to load it for a project.

Adding Weight Sets

ATG Search includes a System Default Term Weights set (one per language). You can construct your own weight set by importing this default set and adding to it, by building your own weight set from scratch, or by importing a weight set.

To add a weight set:

1. On the Search Workbench tab, select Term Weights from the Show menu.
2. Click New Term Weight Set.
3. Enter the name (required) and description (optional) for the weight set.
4. Select a language.
5. Click Create.

To add individual terms to the weight set, see [Adding Terms to Weight Sets](#).

Copying Weight Sets

To copy a weight set:



1. On the Search Workbench tab, select Term Weights from the Show menu.
2. Right-click the name of the weight set and select Copy to New from the menu.

The Content tab's navigation tree displays the new weight set, and the weight set's page opens. To add individual terms and weights to your weight set, see [Adding Terms to Weight Sets](#).

Importing Weight Sets

You can create a weight set by importing a properly formatted XML file into ATG Search. Use this feature to import dictionaries created in previous versions of ATG Search, or when moving your ATG Search installation.

To import a weight set:

1. On the Search Workbench tab, select Term Weights from the Show menu.
2. Click Import Term Weight Set.
3. Browse to the XML file that defines the weight set.
4. Enter a name and description for the new weight set in ATG Search.
5. Click Import.

To add individual terms and weights to your weight set, see [Adding Terms to Weight Sets](#).

Adding Terms to Weight Sets


You can add and delete individual terms and adjust weights within weight sets that you create.

To add a term to your weight set:

1. On the Search Workbench tab, select Term Weights from the Show menu.
2. In the Term Weights, go to Term Weights > *Your Weight Set*.
3. Enter a term. When you begin typing the term, a new line appears below it, on which you can enter more terms.
4. Assign the weight. The range is 0 to 100, with 100 having the highest weight or influence in the search. The colored icon and Weighting Effect column changes to reflect the effect of the assigned weight.
5. Click Save.

Deleting Terms from Weight Sets

To delete a term from a weight set:

1. On the Search Workbench tab, select Term Weights from the Show menu.
2. In the Term Weights, go to Term Weights > *Your Weight Set*.
3. Click the delete icon () for the term.



Deleting Weight Sets

To delete a weight set:

1. On the Search Workbench tab, select Term Weights from the Show menu.
2. Right-click the name of the weight set.
3. Select Delete from the menu.
4. Click Delete to confirm.



11 Text Processing Option Sets

Text Processing Options are parameter settings that control a wide variety of options related to the processing of text both during indexing and query processing. Examples of text processing options include the ability to specify whether specified metatags should be indexed as text or excluded from indexing, whether items are indexed at the document or sentence level, and the minimum number of words that should be considered a distinct sentence.

These parameters are gathered into sets, which apply all of their text processing option settings when building an index or processing a query.

Some individual options within a text processing option set must have the same value across all content in a project (project-level options), while others may vary with the content (content-level options). In Search Administration, a single TPO set is associated with the project as a whole, referred to as the project TPO set. It's possible to also associate option sets with content, and these are called content TPO sets.

This chapter includes the following sections:

[Adding Text Processing Option Sets](#)

[Duplicating Text Processing Option Sets](#)

[Modifying Text Processing Option Sets](#)

[Exporting Text Processing Option Sets](#)

[Deleting Text Processing Option Sets](#)

You can add TPO sets at either the project or the content level.

The option values used for text processing come from:

- The project TPO set if no content TPO set is specified.
- The merge of the project TPO set and content TPO set, if both sets are specified

If both project-level and content-level TPO sets are present, project-level options are always specified by the project TPO set. For content-level options, the content TPO set either explicitly specifies its own value, or leaves it unset, and the value is inherited from the project TPO set. Only one content TPO set can be associated for a given content name.

Note that if you are indexing structured content from a repository, such as a product catalog, you do not need to use a TPO set to specify input languages; the repository indexing process overrides any TPO language selection and uses the languages specified in your `IndexingOutputConfig` component. However, if you are indexing file-based content in multiple languages, you must separate your content by language, and use a content-level TPO set to specify the Input Language for that content. One way to do

this would be to separate the files for each language into a discrete directory; you can then add content representing each directory to the content set separately.

If you have an active search project selected in the Workbench, you can view the merged set of options. For example, if your search project MyProject uses TPO set A at the project level and TPO set B at the content level for some content, if you edit TPO set B with MyProject selected, for any option that says “use project level setting,” it displays the actual setting used in the project.

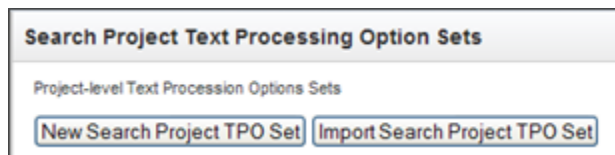
Adding Text Processing Option Sets

You can add or import text processing option sets at either the project or the content level.

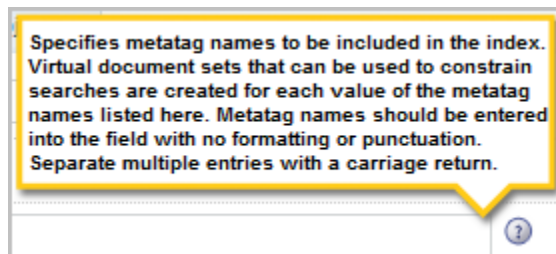
Adding Project TPO Sets

To add a project-level TPO set:

1. On the Search Workbench tab, select Text Processing Option Sets from the Show menu.
2. In the Text Processing Option Sets menu, go to Text Processing Option Sets > Search Project TPO Sets.
3. Click New Search Project TPO Set.



4. Enter a name for the TPO set.
5. Set the TPO options. Options are described in online help; to view the description, place your cursor over the icon. The example shown is for the Metatags option:



6. Click Save Options when finished. You can set options on multiple tabs before saving.



Importing Project TPO Sets

You can create a TPO set by importing a properly formatted XML file into ATG Search. Use this feature to import text processing option sets created in previous versions of ATG Search, or when moving your ATG Search installation.

To import a project-level TPO set:

1. On the Search Workbench tab, select Text Processing Option Sets from the Show menu.
2. In the Text Processing Option Sets menu, go to Text Processing Option Sets > Search Project TPO Sets.
3. Click Import Search Project TPO Set.
4. Browse to the XML file that describes the TPO set you want to import.

Search Project Text Processing Option Sets

Import from XML File

File Path:

Import to Text Processing Option Set

Name:

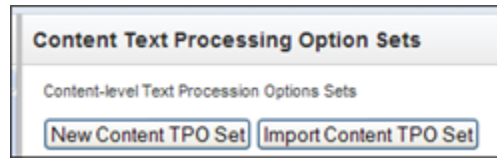
Type: Search Project
 Content - any settings specific to the Search Project level will be ignored

5. Enter a name for the new TPO set.
6. Select the TPO set's level.
7. Click Import.

Adding Content TPO Sets

To add a content-level TPO set:

1. On the Search Workbench tab, select Text Processing Option Sets from the Show menu.
2. In the Text Processing Option Sets menu, go to Text Processing Option Sets > Content TPO Sets.
3. Click New Content TPO Set.



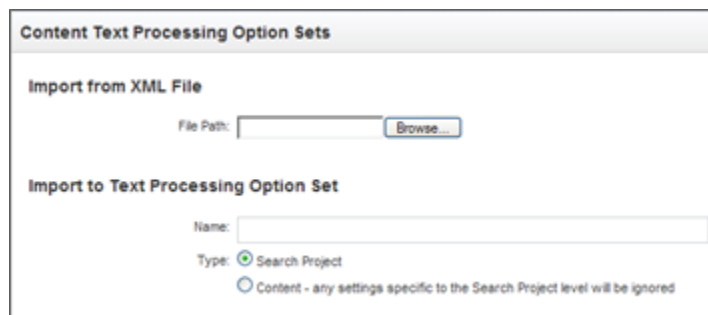
4. Enter a name for the TPO set.
5. Set the TPO options. Options are set to “Use Search Project-level setting” by default. Select “Use Content-level setting” to enable an option. Options are described in online help. To access the item description, place your cursor over the question mark icon.
Note: TPO options related to topic sets, such as Enable topic patterns, can only be used at the project level.
6. Click Save Options when finished.

Importing Content TPO Sets

You can create a TPO set by importing a properly formatted XML file into ATG Search. Use this feature to import text processing option sets created in previous versions of ATG Search, or when moving your ATG Search installation.

To import a content-level TPO set:

1. On the Search Workbench tab, select Text Processing Option Sets from the Show menu.
2. In the Text Processing Option Sets menu, go to Text Processing Option Sets > Content TPO Sets.
3. Click Import Content TPO Set.
4. Browse to the XML file that describes the TPO set you want to import.



5. Enter a name for the new TPO set.
6. Select the TPO set’s level.
7. Click Import.



Using Table Structure Parsing

You can use the Table structure parsing text processing option (located on the Advanced tab of the TPO set) to specify how tables should be parsed in HTML documents. Because this feature is intended to preserve relationships between data items in a row or column, use it for tables that contain data, as opposed to tables used for page layout. ATG Search requires that tables be identified by an `id` or `class` attribute in order to be parsed. Table parsing is specified using the form:

```
id_or_class|type|start|rowList|colList|titleType, titleArg
```

Only the `id/class` and `type` are required, the other values are optional. Explanations of each value follow.

- `id` or `class`—These attributes on the `<table>` tag identify the table to associate with this parsing information. Required.
- `type`—Describes how the table is organized, and can be any of the values listed below (`ObjectFeatureValue`, `RowFeatureList`, etc.). Required.

ObjectFeatureValue—Consists of a left-hand column of items and a header row of features or attributes. The remaining cells specify the value for the feature of the corresponding item. Although the cells in a row are related, the system treats each cell (with its object and feature) as separate text statements which can be retrieved separately.

Item	Color	Shape
ItemA	Blue	Square
ItemB	Red	Circle

The ItemA row would resolve to the following:

```
ItemA Color Blue
ItemA Shape Square
```

RowFeatureList—Consists of a header row of related features or attributes. The remaining cells specify the value for the feature. Very similar to `ObjectFeatureValue`. Since the cells in a row are related, the system treats each row (with its features) as separate text statements which can be retrieved separately.

	Color	Shape
ItemA	Blue	Square
ItemB	Red	Circle

The two rows would resolve to the following:



ItemA Color Blue Shape Square

ItemB Color Red Shape Circle

RowFeature—Consists of a header row of unrelated features or items. The remaining columns specify information about their corresponding header cell. Similar to ObjectFeatureValue, except that the left-hand column provides no information across the row, and the cells show more than just a value (possibly even full sentences). Since the cells in a column are unrelated, in this logical interpretation, the system treats each cell (with its feature) as separate text statements which can stand on their own (i.e. be retrieved separately).

ItemA	ItemB	ItemC
Is a cube	Is a pyramid	Is a sphere
Is green	Is blue	Is red

The ItemA column would resolve to the following:

ItemA Is a cube

ItemA Is green

RowOnly—Consists of a header row of unrelated features or items. The remaining columns specify information about their corresponding header cell. Similar to the Row Feature structure, except that the cells show short related information, not full sentences. Since the cells in a row are related, the system treats each row as separate text statements which can be retrieved separately.

ItemA is	Blue	Square
ItemB is	Red	Circle

The ItemA row would resolve to the following:

ItemA is Blue Square

ColumnFeatureList—Same as RowFeatureList, but rotated so that the header is the left-most column.

Item	ItemA	ItemB
Color	Blue	Red
Shape	Square	Circle



The ItemA column would resolve to the following:

ItemA Color Blue Shape Square

ColumnFeature—Same as RowFeature, but rotated so that the header is the left-most column.

ItemA	Is a cube	Is green
Item B	Is a pyramid	Is blue
Item C	Is a sphere	Is red.

The ItemA row would resolve to the following:

ItemA Is a cube Is green

ColumnOnly—Same as RowOnly, but rotated so that the header is the left-most column.

ItemA is	Item B is
Blue	Red
Square	Circle

The ItemA column would resolve to the following:

ItemA is Blue Square

- start—Either the starting row or starting column, depending on the type; the default is 0, referring to the topmost or leftmost column. Optional.
- rowList—Comma-delimited list of rows to traverse. If this value is included, any rows not specified are excluded from indexing. Optional.
- colList—Comma-delimited list of column to traverse. If this value is included, any columns not specified are excluded from indexing. Optional.
- titleType and titleArg—If your table has a title you want to include in the logical structure of the table, specify explicit as the titleType, and the text of the title as the titleArg.

Duplicating Text Processing Option Sets

To duplicate a TPO set:

1. On the Search Workbench tab, select Text Processing Option Sets from the Show menu.

2. Right-click the name of the set you want to copy.
3. Select Copy to New from the menu.

The TPO set and its configuration are copied. The new TPO set is named Copy of *Name*.

Modifying Text Processing Option Sets

To modify an existing TPO Set:

1. On the Search Workbench tab, select Text Processing Option Sets from the Show menu.
2. In the Text Processing Option Sets menu, go to Text Processing Option Sets > Search Project TPO Sets or Content TPO Sets > *Your Option Set*.
3. Make your changes.

Options in content-level TPO sets are set to “Use Search Project-level setting” by default. Select “Use Content-level setting” to enable an option.

Options are described in the Text Processing Option Reference section of this chapter.

4. When finished, click Save Options.

Exporting Text Processing Option Sets

To export a TPO set:

1. On the Search Workbench tab, select Text Processing Option Sets from the Show menu.
2. In the Text Processing Option Sets menu, go to Text Processing Option Sets > Search Project TPO Sets or Content TPO Sets > *Your Option Set*.
3. Right-click the name of the set you want to export.
4. Select Export from the menu.
5. Select “All settings” or “Only settings changed from default” to export.
6. Click Export. The export uses your browser’s file download defaults.

Deleting Text Processing Option Sets

To delete a text processing option set:

1. On the Search Workbench tab, select Text Processing Option Sets from the Show menu.



2. In the Text Processing Option Sets menu, go to Text Processing Option Sets > Search Project TPO Sets or Content TPO Sets > *Your Option Set*.
3. Right-click the name of the set you want to delete.
4. Select Delete from the menu.
5. Click Delete to confirm.





12 Facet Sets

Facets allow users to search within an existing result set. For example, an end-user conducts a search for luggage on a commerce site, then refines their search by color, price, or material. Facets are organized hierarchically, with “child facets” allowing users to refine their search at additional levels. For example, a merchandiser at a luggage commerce site could configure product properties such as color, price, and material. When an end-user conducts a search, they can refine the results using those properties to find exactly what they’re looking for.

Facets are typically created using ATG Merchandising (see the [ATG Merchandising Guide for Business Users](#)). If you don’t want to use Merchandising, you can create facets in Search Administration, following the steps described in this chapter. Keep in mind that some facet support tasks that Merchandising does automatically are manual in Search Administration.

Facets are based on properties. One common case is that properties are associated with items in a catalog repository, but you can also create facets for unstructured content that has been appropriately tagged with meta-properties. For catalog search purposes, properties must be identified as either text properties (which can be searched on) or meta-properties (which can be used as facets or as constraints on searches). When you index your catalog, both property types are output as an XHTML representation and indexed, but meta-properties and text properties are represented differently. The names of meta-properties are determined either explicitly, from the `IndexingOutputConfig` properties files, or are automatically constructed from the item definition, for example `product.childSKU.color`.

When you create facets in ATG Merchandising, the Merchandising application gets a list of the valid meta-property names from `IndexingOutputConfig` component. When you construct facets in Search Administration, you have to supply the correct meta-property name. You may need to look at the generated XHTML to confirm the names. See [Indexing ATG Commerce Catalog Data](#) for information on XHTML generation.

Additionally, the form handlers and components supplied for ATG Commerce include code to dynamically select or explicitly specify a Merchandising facet set in the query request attributes. These form handlers would have to be modified to select the appropriate facet sets created in Search Administration. Facet sets created in ATG Merchandising reside in the `/atg/search/repository/RefinementRepository`, which is versioned. Facet sets created in Search Administration reside in the `/atg/searchadmin/RefinementConfigRepository`, which is not versioned.

This chapter includes the following sections:

[Creating Facet Sets](#)

[Adding Child Facets](#)



- [Moving Facets](#)
- [Modifying Facet Sets and Facets](#)
- [Deleting Facet Sets and Facets](#)
- [Importing Facet Sets](#)
- [Exporting Facet Sets](#)
- [Duplicating Facet Sets](#)
- [Viewing Facet Set Adaptors](#)

Creating Facet Sets

To create a facet set:

1. On the Search Workbench tab, select Facet Sets from the Show menu.
2. Click New Facet Set.
3. Enter a name (required) and description (optional).

The screenshot shows a 'New Facet Set' dialog box with two tabs: 'General' and 'Associated Search Projects'. The 'General' tab is active. Under the 'Facet Set Basics' section, there are three input fields: 'Name' (required, indicated by a star), 'Description', and 'Mappings'.

4. In the Mappings field, you can optionally enter property values to associate with this facet set. When you begin typing a mapping entry, a new line appears so that you can enter additional mappings.
5. Click Create.

Adding Child Facets

Child facets allow end-users to progressively refine their search results. To add a child facet:

1. On the Search Workbench tab, select Facet Sets from the Show menu.
2. In the Facet Sets menu, go to Facet Sets > *Your Facet Set*.
3. Click Add New Facet.



Top-Level Facets: 0

Facet Name	Property	Property Type
(None)		

4. Enter the facet name, and the property on which the refinement is based.
5. Select the property type: float, integer, string, date, enum, or boolean.
6. Select a Facet Value Display Method. This selection affects the other options available for configuring the facet.
 - One selection for each facet property value—Each individual value is returned and displayed to the user for refinement. This option works best for properties such as Brand, in which the user is expected to select a single value by which to refine results.
 - Ranges—Ranges of values are returned and displayed to the user. Use this option for facets based on a property such as price; you would probably not want a user to refine their search by an exact price (such as \$9.99), but by a range, such as \$5.00-\$9.99.
7. If using One selection for each facet property value, select the facet sorting order.
 - Alphanumeric—Returned facet values are sorted alphabetically.
 - Value Count—Returned facet values are sorted according to the number of items that have that facet value.
 - Explicit Order—If using this sorting method, you must specify the values to use:
 - Specified Facet Values—Returns only the values you enter, in the order you provide.
 - Excluded Facet Values—Specifically excludes the values entered.
8. If using Ranges, specify a Selection Option.
 - Dynamic-range selections – Generates the ranges automatically based on guidelines you specify.

Dynamic-range selections - generate automatically based on guidelines:

Maximum number of selections:

Minimum facet property values per selection:

Minimum numerical span per selection:

Round selection values to the nearest:

- Fixed-range selections – You specify the size of the ranges.



Fixed-range selections - generate selections based on guidelines:

Number of selections:

Numerical span per selection:

- Specified selection ranges – Generate selections using these range points.

Enter a definition point for each range. The last value you enter becomes the maximum. For example, to create ranges A-D, D-M, M-S, and S-Z, enter range points A, D, M, S, and Z. Values are included in the range using a “less than” operation for all except the final point (for example, the A-D range includes values from A to C, etc.).

Note that the values you enter must match the Property Type selected; for example, you cannot select a Property Type of *float* and use alphabetic ranges.

To delete a range point, click the red X.

Specified selection ranges - generate selections using these range points:

From

Before Adding Range Points

Specified selection ranges - generate selections using these range points:

From	A	<input type="button" value="X"/>
To	D	<input type="button" value="X"/>
To	M	<input type="button" value="X"/>
To	S	<input type="button" value="X"/>

After Adding Range Points

Then specify the sorting order:

- Alphanumeric—Returned facet values are sorted alphabetically.
- Value Count—Returned facet values are sorted according to the number of items that have that facet value.
- Explicit Order—If using this sorting method, you must specify the values to use:
 - Specified Facet Values—Returns only the values you enter, in the order you provide.
 - Excluded Facet Values—Specifically excludes the values entered.

9. Configure the Advanced settings (optional):



- Filter values by the active query constraints
 - Decimal Precision of floating point ranges
 - Evaluation Levels
 - Value Mapped Data
 - Conditional Parent Selection
10. Click Create if you are finished adding facets or Create & Add Another to add more child facets at this level.

Moving Facets

To move a facet within a facet set:

1. On the Search Workbench tab, select Facet Sets from the Show menu.
2. In the Facet Sets menu, go to Facet Sets > *Your Facet Set*.
3. Right-click the facet and select from the menu to copy or cut it.
4. Right-click the facet set or facet under which you want to paste the moved facet. You can paste the moved facet either at the same level as the selected facet, or as a child of that facet.

Modifying Facet Sets and Facets

To modify a facet set or facet:

1. On the Search Workbench tab, select Facet Sets from the Show menu.
2. In the Facet Sets menu, go to Facet Sets > *Your Facet Set*.
3. Make your changes.
4. Click Save.

Deleting Facet Sets and Facets

To delete a facet set or facet:

1. On the Search Workbench tab, select Facet Sets from the Show menu.
2. In the Facet Sets menu, go to Facet Sets > *Your Facet Set*.
3. Right-click the name of the facet set or facet you want to delete.
4. Select Delete from the menu.
5. Click Delete to confirm.



Importing Facet Sets

You can create a facet set by importing a properly formatted XML file into ATG Search. Use this feature to import facet sets created elsewhere, or when moving your ATG Search installation

To import a facet set:

1. On the Search Workbench tab, select Facet Sets from the Show menu.
2. Click Import Facet Set.

3. Browse to the XML file that describes the facet set you want to import.
4. Enter a name for the new facet set.
5. Click Import.

Exporting Facet Sets

To export a facet set to an XML file:

1. On the Search Workbench tab, select Facet Sets from the Show menu.
2. In the Facet Sets menu, go to Facet Sets > *Your Facet Set*.
3. Right-click the facet set and select Export from the menu.

4. On the Export Facet Set page, click Export. The export uses your browser's file download defaults.



Duplicating Facet Sets

To duplicate a facet set:

1. On the Search Workbench tab, select Facet Sets from the Show menu.
2. In the Facet Sets menu, go to Facet Sets > *Your Facet Set*.
3. Right-click the facet set and select Copy to New from the menu. The facet set and all of its child facets are copied. The new facet set's default name is Copy of *Name*.

Viewing Facet Set Adaptors

In addition to the facet sets you create in the Workbench, you can create and manage facets using ATG Merchandising (see the [ATG Merchandising Guide for Business Users](#)). These two types of facet sets exist completely independent of one another; you cannot edit facet sets created in Merchandising in the Workbench, or vice versa.

You can, however, use the Workbench to view your indexed facet sets regardless of where they were created. This allows you to see which facet sets have been indexed and confirm the definitions of individual facet sets.

To view facet set adaptors:

1. On the Search Workbench tab, select Facet Sets from the Show menu.
2. In the Facet Sets menu, go to Facet Set Adaptors.

The page lists your facet sets and their points of origin.

Facet Set Adaptors		
Facet Set Adaptor	Location	Auto Apply All
File System Refinement Config	Directory: file://C:/ATG/ATG9.0/home/localconfig/atg/searchadmin/customizations/refinements/	No
(Local)	Search Workbench	No

When "Auto Apply All" is turned on, all facet sets from that adapter are included in the index.





13 Query Rules

Query Rules allow you to create patterns, which are matched against queries that users submit and then perform actions in response to those matches.

One situation in which you might want to use query rules is when users commonly search for terms that are not included in the indexed content. For example, they often search for “boy’s clothing,” but the product descriptions in your catalog do not contain the word “boy.” A query rule could indicate that queries containing “boy” should also check for “men” and “his.”

A rule set consists of any number of rules, which are organized into rule groups.

This chapter includes the following sections:

[Creating Query Rule Sets](#)

[Enabling and Disabling Query Rules, Patterns, and Actions](#)

[Importing Query Rules](#)

[Exporting Query Rules](#)

[Moving Query Rule Sets and Rule Groups](#)

[Deleting Query Rule Sets](#)

[Deleting Query Rule Groups](#)

[Deleting Query Rules](#)

[Deleting Patterns and Actions](#)

Creating Query Rule Sets

The overall process for adding query rules is:

1. Create a rule set; see [Creating a Rule Set](#).
2. Add a group to the rule set; see [Adding Rule Groups to Rule Sets and Rule Groups](#).
3. Add rules to the groups; see [Adding Rules to Rule Groups](#).

Note: If you change the name of a query rule set or delete a query rule set, you must perform a full reindex of your content.



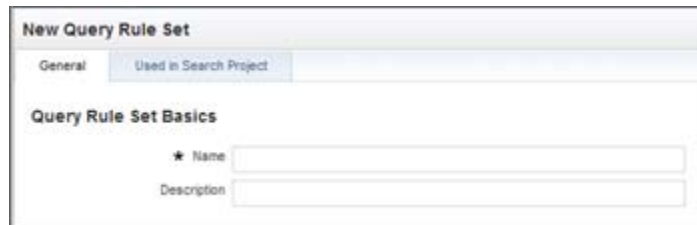
Creating a Rule Set

To create a new rule set:

1. On the Search Workbench tab, select Query Rules from the Show menu.
2. Click New Query Rule Set.



3. Enter a name (required) and description (optional) for the rule set.



4. Click Create, or Create and Add New Group to proceed immediately to adding rule groups to the new rule set.

Adding Rule Groups to Rule Sets and Rule Groups

Rule Groups are used to organize rules within your rule set. To add a rule group to a rule set:

1. On the Search Workbench tab, select Query Rules from the Show menu.
2. In the Query Rules menu, go to Query Rule Sets > *Your Rule Set*.
3. Click Add New Groups.



4. Enter a name for each new group. New rows appear automatically as you enter group names, allowing you to add as many as you want.



5. Click Save.

To add a subgroup to a rule group:

1. On the Search Workbench tab, select Query Rules from the Show menu.
2. In the Query Rules menu, go to Query Rule Sets > *Your Rule Set* > *Your Rule Group*.
3. Under Child Query Rule Groups, click Add New Groups.

4. Enter a name for each new group. New rows appear automatically as you enter group names, allowing you to add as many as you want.

5. Click Save.

Adding Rules to Rule Groups

Query rule patterns consist of a sequence of elements and operators that are matched against the end-user's search input. A text pattern consists of one or more elements; all elements must match in order for the whole pattern to match. A single element might match multiple index terms. When a match is found, the query rule actions modify the query behavior as specified in the rule.



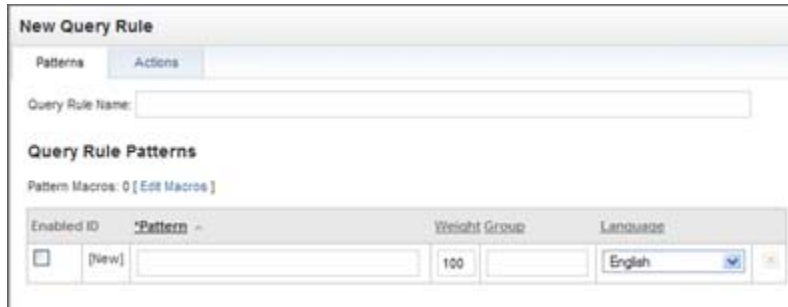
Rule syntax is described in the table following this procedure.

To add a new query rule:

1. On the Search Workbench tab, select Query Rules from the Show menu.
2. In the Query Rules menu, go to Query Rule Sets > *Your Rule Set* > *Your Rule Group*.
3. Under Child Query Rules, click Add New Rule.



4. On the Pattern tab, enter the following:



- The pattern itself.
- Weight given to the pattern (default value is 100).
- Group name, if desired. This is not the same as the rule group, but a feature that allows you to activate or deactivate multiple rules as a unit.
- The language of the pattern.

5. On the Actions tab, enter the following:



- The action to be performed on the pattern.
- Note:** At least one action must be defined for each rule.



- The language of the action.

The Language value controls what language is used for references to literal terms. Since the rule's terms are only in one language, then they can only match if that language is also the query language. The exception to this is the %term% syntax, which will expand this term using any available cross-language expansions. Specifying the language value on a rule expresses the concept of "specific" language rules.

6. Click Create to add this rule and return to the rule group, or Create & Add Another to add more patterns to the rule.

The table that follows describes the pattern elements and their syntax.

Pattern Element	Description
Simple index term element	<p>The most basic pattern element is a simple index term, expressed as just the alphabetic string of the term, such as sel I . ATG Search performs morphological analysis when loading the text patterns, so the index term of the string is used, rather than the literal string.</p> <p>Example: Simple element sel I i ng is interpreted as the index term sel I , which matches any form of sel I , such as sel I i ng, sol d, and sel I s.</p>
Literal term element	<p>To match an exact form of a term, enclose the alphabetic string of the form in double-quotes. This restricts the index term to matching the morphological information in the element.</p> <p>Example: The element "sel I i ng" only matches sel I i ng.</p>
Number term element	<p>An element consisting of numbers is treated in the same way as a simple index term element. Typically, numbers don't have morphological forms, except some plurals (such as 1980s) and ordinals (10th). To restrict the number element to match only the literal number, enclose the element in double-quotes, just like the literal term element.</p>



Pattern Element	Description
Expanded index term element	<p>The simple index term element only matches morphological forms of itself. To expand the pattern to match any of the term's thesaurus entries as well, enclose it in percent characters, such as %sel I %.</p> <p>When this syntax is used, the element matches any form of the term, plus any of the terms listed in its thesaurus entries. The element can be limited to thesaurus entries for a particular part of speech by appending it after the term.</p> <p>Examples: %sel I : verb% or %charge: noun%.</p>
Wild card elements	<p>Text patterns can contain wild cards. The results are similar to the expanded term element, except the expansion does not use the thesaurus.</p> <p>For example, the element i nstal I * expands to any index term that begins with i nstal I .</p>
Negative elements	<p>Elements marked with the negative operator (!), such as ! buy or ! "buyi ng" require that the statement does not include these terms in order for the pattern to match.</p> <p>Note: You cannot create a rule with only negative elements; for instance, you cannot specify "!shoes" as the pattern, since that would require ATG Search to match on every other imaginable search term.</p>
Element sequences	<p>A text pattern can consist of more than one element. Multiple elements that appear in the same pattern with intervening white space create a Boolean AND operator, such as <i>sell stock</i>. This pattern matches the elements in any order (either <i>sell stock</i> or <i>stock sell</i>) but requires them to be adjacent.</p>
Element order operator	<p>To stipulate that one element must precede another in order for the pattern to match a statement, use the order operator (-) between the elements, such as sel I - stock. This operator only affects the two surrounding elements. Preceding white space is required to distinguish the operator from a normal hyphen. For example, F-14 would not be considered the order operator.</p>



Pattern Element	Description
Element distance operator	<p>By default, two adjacent elements in a pattern are required to be adjacent in the matching statement. To adjust the distance that is allowed to appear between the elements, a period is used. This operator can take the following forms:</p> <p>. – a single period means one intervening term is allowed between the elements.</p> <p>.. – two periods mean two intervening term are allowed between the elements.</p> <p>... – three periods mean three intervening terms are allowed between the elements; same as .3; and so on, with more periods.</p> <p>.N – a single period followed by an integer means N intervening terms are allowed.</p> <p>.* – a single period followed by an asterisk means infinite intervening terms are allowed.</p> <p>Note that white-space is required before and after any of these operators.</p>
Element order and distance operators	<p>Text patterns can combine the order and distance operators to require that the elements must be in order and within the denoted distance. The operators can be added in either order, but both require preceding white space.</p> <p>Example: sel l . . stock or sel l . . – stock..</p>
Element macros	<p>Users can define macros that expand to an OR of terms when referenced. The syntax of a macro reference is \$name. For example, the user might define a macro called StockActi on as the terms sel l , buy, purchase, and trade. The text patterns can now reference \$StockActi on, which expands to those four terms and matches any form of any of those terms.</p>

Query rules have actions that execute and modify the query behavior in some way. The actions are expressed in a command argument notation *action,arg1,arg2,...,argN*. Most actions refer to patterns, but a few act independently. The action types and their argument structures are described in the table that follows.



Action	Syntax	Argument Description	Example
Add additional terms to the query	<i>add, term, term, ...</i>	<i>term</i> —Strings representing any index term. If the term is a morphological form, it is analyzed and converted to its index term. The action makes no reference to the pattern.	The following action adds the term <i>help</i> to the query terms: add, hel p
Remove terms from the query	<i>del ete, refterm, refterm, ...</i>	<i>refterm</i> —References to index term elements in the query patterns, using the same string as in the pattern. A <i>refterm</i> argument that can not be found as an element is ignored.	The following action deletes the term <i>system</i> from a query that matches the given pattern: customer . * support . * system del ete, system
Change the weight of terms in the query	<i>wei ght, val ue, refterm, refterm, ...</i>	<i>value</i> —A non-negative integer ranging from 0 to 100. <i>refterm</i> —Reference to index term elements of the query patterns, using the same string as in the pattern. A <i>refterm</i> argument that can not be found as an element is ignored.	The following action sets the weight of the term <i>floppy</i> to 100 in a query that matches the given pattern: fl oppy di sk . * erro r wei ght, 100, fl oppy
Add additional negative (!) terms to the query	<i>not, term, term, ...</i>	<i>term</i> —Strings representing any index term. If the term is a morphological form, it is analyzed and converted to its index term.	The following action adds a negated term <i>!fee</i> to a query term vector: not, fee
Add additional required (+) terms to the query	<i>addreq, term, term, ...</i>	<i>term</i> —Strings representing any index term. If the term is a morphological form, it is analyzed and converted to its index term.	The following action adds a required term <i>+procedure</i> to the query term vector: addreq, procedure
Add additional negative (!!) terms to the query	<i>notdoc, term, term, ...</i>	<i>term</i> —Strings representing any index term. If the term is a morphological form, it is analyzed and converted to its index term.	The following action adds a negated term <i>!sony</i> to the query term vector: notdoc, sony



Action	Syntax	Argument Description	Example
Add additional required (++) terms to the query	<code>addreqdoc, term, term, ...</code>	<i>term</i> —Strings representing any index term. If the term is a morphological form, it is analyzed and converted to its index term.	The following action adds a required term <code>++dell</code> to the query term vector: <code>addreqdoc, del l</code>
Make a query term required (+) in the statement results	<code>requi re, refterm, refterm, ...</code>	<i>refterm</i> —A reference to index term elements of the query patterns, using the same string as in the pattern. A <i>refterm</i> argument that can not be found as an element is ignored.	The following action makes the term <i>floppy</i> required in queries that match the given pattern: <code>fl oppy – di sk</code> <code>requi re, fl oppy</code>
Make a query term required (++) in the index item (document) results	<code>requi redoc, refterm, refterm, ...</code>	<i>refterm</i> —Index term elements of the query patterns, using the same string as in the pattern. A <i>refterm</i> argument that can not be found as an element is ignored.	The following action makes the term <i>insurance</i> required in document in queries that match the given pattern: <code>i nsurance . * l i abi l i ty</code> <code>requi redoc, i nsuranc e</code>
Make a query term have no term expansions	<code>noexpand, refterm, refterm, ...</code>	<i>refterm</i> —A reference to index term elements of the query patterns, using the same string as in the pattern. A <i>refterm</i> argument that can not be found as an element is ignored.	The following action removes the expansions for the term <i>charge</i> in queries that match the given pattern: <code>bat tery . * charge</code> <code>noexpand, charge</code>
Change the expansion of a query	<code>expand, refterm, term, term, ...</code>	<i>refterm</i> —A reference to a simple index term element of the query patterns, using the same string as in the pattern. A <i>refterm</i> argument that can not be found as an element is ignored. <i>term</i> —Forms of any index term, reduced by morphological analysis.	The following action changes the expansion of the term <i>book</i> in queries that match the given pattern: <code>savi ngs – book</code> <code>expand, book, account , passbook</code>



Action	Syntax	Argument Description	Example
Add to the expansion of a query term	<i>addexpand, refterm, term, term, term, ...</i>	<i>refterm</i> —A reference to a simple index term element of the query patterns, using the same string as in the pattern. A <i>refterm</i> argument that can not be found as an element is ignored. <i>term</i> —Forms of any index term, reduced by morphological analysis.	The following action adds the expansion <i>account</i> to the term <i>book</i> in queries that match the following pattern: savings – book addexpand, book, account
Remove terms from the expansion of a query term	<i>del expand, refterm, term, term, term, ...</i>	<i>refterm</i> —A reference to a simple index term element of the query patterns, using the same string as in the pattern. A <i>refterm</i> argument that can not be found as an element is ignored. <i>term</i> —Forms of any index term, reduced by morphological analysis.	The following action deletes the expansion <i>human</i> from the term <i>person</i> for queries that match the given pattern: sales – person del expand, person, human
Make a query term match literally	<i>literal, refterm, refterm, ...</i>	<i>refterm</i> —A reference to index term elements of the query patterns, using the same string as in the pattern. A <i>refterm</i> argument that can not be found as an element is ignored.	The following action makes the term <i>plan</i> double-quoted in queries that match the given pattern: savings – plan literal, plan
Make a sequence of query terms match literally (in double-quotes)	<i>phrase, refterm, refterm, ...</i>	<i>refterm</i> —A reference to sequential index term elements of the query patterns, using the same string as in the pattern. A <i>refterm</i> argument that can not be found as an element is ignored.	The following action makes the phrase <i>savings plan</i> double-quoted in queries that match the given pattern: savings – plan phrase, savings, plan



Action	Syntax	Argument Description	Example
Make a sequence of query terms match in order and within an adjacency distance constraint	<i>sequence, adjacency, refterm, refterm, ...</i>	<p><i>adjacency</i>—The distance permitted between the terms of the sequence, where 0 means adjacent, 1 means one intervening term, and so on.</p> <p><i>refterm</i>—A reference to sequential index term elements of the query patterns, using the same string as in the pattern. A <i>refterm</i> argument that can not be found as an element is ignored.</p>	<p>The following action makes the sequence <i>savings plan</i> with adjacency 1 in queries that match the given pattern:</p> <p>savi ngs – pl an</p> <p>sequence, 1, savi ngs, pl an</p>
Make a sequence of query terms match within an adjacency distance constraint	<i>adjacency, adjacency, refterm, refterm, ...</i>	<p><i>adjacency</i>—The distance permitted between the terms of the sequence, where 0 means adjacent, 1 means one intervening term, and so on.</p> <p><i>refterm</i>—A reference to sequential index term elements of the query patterns, using the same string as in the pattern. A <i>refterm</i> argument that can not be found as an element is ignored.</p>	<p>The following action makes the terms <i>savings plan</i> have adjacency 2 in queries that match the given pattern:</p> <p>savi ngs – pl an</p> <p>adj acency, 2, savi ngs , pl an</p>
Add a document set constraint to the current query constraints	<i>docset, docset, docset, ...</i>	<p><i>docset</i>—A valid document set. A <i>docset</i> argument that is invalid is ignored. If more than one <i>docset</i> argument is provided, a Boolean OR of the document set constraints is added. If the current query already has constraints, these added constraints are joined with a Boolean AND.</p>	<p>The following action adds an OR of document sets as a constraint for the current query:</p> <p>docset, /Topi cs/Prod ucts/Sony, /Topi cs/P roducts/Panasoni c</p>
Add a document URL constraint to the current query constraints	<i>doc, url, url, ...</i>	<p><i>url</i>—A reference to a valid index item. A <i>url</i> argument that is invalid is ignored. If more than one <i>url</i> argument is provided, a Boolean OR of the document constraints is added. If the current query already has constraints, these added constraints are joined with a Boolean AND.</p>	<p>The following action adds an OR of two URLs as a constraint for the current query:</p> <p>doc,http://www.mycorp.com/home.html,http://www.mycorp.com/help.html</p>



Action	Syntax	Argument Description	Example
Add a meta-data property constraint to the current query constraints	<code>prop, type, name, val, val, ...</code>	<p><i>type</i>—A valid property type (enum, string, integer, float, boolean).</p> <p><i>name</i>—A valid property of the given type within the indexed content.</p> <p><i>val</i>—Values of the given property. If more than one <i>val</i> argument is provided, a Boolean OR of the property constraints is added. If the current query already has constraints, these added constraints are joined with a Boolean AND.</p>	<p>The following action adds an enum property constraint with 2 possible values:</p> <p><code>prop, enum, status, published, internal</code></p>
Add a document set preference to the current query	<code>docsetpref, weight, docset, docset, ...</code>	<p><i>weight</i>—A positive integer between 1 and 100; represents a preference for results that are within the document set arguments. The preference values effect the relevancy computation.</p> <p><i>docset</i>—A reference to a valid document set. A <i>docset</i> argument that is invalid is ignored. If more than one <i>docset</i> argument is provided, a Boolean OR of the document set preferences is added.</p>	<p>The following action adds a document set preference of 50 for one document set, essentially giving it 50 bonus points in the relevancy computation:</p> <p><code>docsetpref, 50, /Topics/Products/Sony</code></p>
Add a document URL preference to the current query	<code>docpref, weight, url, url, ...</code>	<p><i>weight</i>—A positive integer between 1 and 100; represents a preference for results that are within the document set arguments. The preference values effect the relevancy computation.</p> <p><i>url</i>—A reference to a valid index item. A <i>url</i> argument that is invalid is ignored. If more than one <i>url</i> argument is provided, a Boolean OR of the document preferences is added.</p>	<p>The following action adds a document preference of 60 for 1 document:</p> <p><code>docpref, 60, http://www.mycorp.com/home.html</code></p>



Action	Syntax	Argument Description	Example
Add a meta-data property preference to the current query	<i>proppref, weight, type, name, val, val, ...</i>	<p><i>weight</i>—A positive integer between 1 and 100; represents a preference for results that are within the document set arguments. The preference values effect the relevancy computation.</p> <p><i>type</i>—A valid property type (enum, string, integer, float, boolean).</p> <p><i>name</i>—Must be a valid property of the given type within the indexed content.</p> <p><i>val</i>—Values of the given property. If more than one <i>val</i> argument is provided, a Boolean OR of the property preferences is added</p>	<p>The following action adds an enum property preference of 40 with 2 possible values:</p> <p><code>proppref, 40, enum, status, published, internal</code></p>
Add a statement feature constraint (i.e. for fielded search) to the current query	<i>feature, field, field, ...</i>	<i>field</i> —Must be valid text fields. A text field that is invalid is ignored. If more than one <i>field</i> argument is provided, a Boolean OR of the feature constraint is added.	<p>The following action adds a statement feature constraint of 2 fields:</p> <p><code>feature, role: fix, role: change</code></p>
Add an arbitrary action result that will be included in the response to the current query	<i>action, arg, arg, ...</i>	<i>arg</i> —Arbitrary strings which are simply returned back with the response in a comma-delimited list.	<code>action, advisor, auto, http://www.xyz.com/foo.html</code>

Rule Example: Adding a Property Constraint

This example shows a simple query rule. Assume that your company sells clothing. The content tags Men’s and Women’s clothing with an item property, but these words themselves do not appear in the searchable text. Therefore, if a user searches for “Men’s shoes” the only matching term is “shoes,” and the results include both men’s and women’s shoes. The query rules check for the terms man, boy, woman, and girl (which also match their plural forms) and adds a property constraint as appropriate.



Pattern	Action
Man	prop, string, Category. Name, Mens
Boy	prop, string, Category. Name, Mens
Woman	prop, string, Category. Name, Womens
Gi rl	prop, string, Category. Name, Womens

With these query rules in use, a search for “men’s shoes” triggers an action that adds a property constraint, so that only items where the property Category. Name has the value Mens are searched.

Rule Example: Redirecting Users

One way you can use query rules is to ensure that a particular search result is displayed when customers use particular words or phrases in their search terms. For example, if someone enters a search “where is MyBusiness located,” you want to make sure that your company’s Contact Information page is displayed first in the search results, even though the search terms do not appear in the indexed page text. To accomplish this, you can use an arbitrary query rule action with the following syntax:

```
acti on, arg, arg, . . .
```

The arg arguments are arbitrary strings which are returned back with the search results in a comma-delimited list. Your search user interface must include code that knows how to interpret those arguments via the Search API (see the [ATG Search Query Guide](#) for information). For example, to redirect users you could use something resembling the following argument list:

```
acti on, auto, http: //www. xyz. com/fo o. html
```

Your client code could then redirect users to the returned URL. An example of this use can be seen the ATG Merchandising search configurations, which allow you to create redirection rules that are translated by ATG Search into query rules.

Rule Example: Promoting Preferred Results

If you don’t want to redirect users, but only make a particular search result the preferred response, you can use the *document URL preference* query rule with the following syntax:

```
docpref, wei ght, ur l , ur l , . . .
```

The wei ght argument is a positive integer between 1 and 100, and adjusts the relevancy computation to favor results that are from the document or documents listed in the URL arguments. The ur l arguments must be a reference to a valid index item (invalid URLs are ignored). If more than one ur l argument is provided, a Boolean OR of the document preferences is added. For example, the following action adds a document preference of 60 for results from one document:

```
docpref, 60, ht t p: //www. mycorp. com/home. html
```



Using Macros in Rules

Global Macros are additional data that is stored outside of any particular Rule Set. Macros work the same way in rules as they do in Topic Sets; see the [Using Macros in Patterns](#) section in this guide.

Enabling and Disabling Query Rules, Patterns, and Actions

You can disable and enable individual rules and patterns. You can also disable rules by group, using the Enable query rules and Disable query rules options in a TPO set (see the Query Options section of the Text Processing Option Sets chapter).

At the highest level, you can enable and disable query rules entirely using the `ruleMode` attribute on the request object:

```
<query ruleMode="mode">
```

The mode value is one of the following:

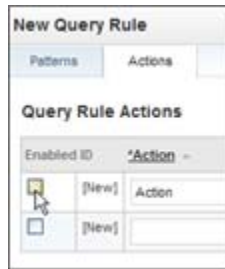
- `ignore` – No query analysis will be performed.
- `display` – Perform the query analysis, but simply return feedback about the results in the response.
- `exec` – Perform the query analysis and execute the actions, returning feedback about what was executed.

The default value is `display`, although only an index that contains query rules will return results.

See the [ATG Search Reference Guide](#) for more information.

To enable or disable a rule or pattern:

1. On the Search Workbench tab, select Query Rules from the Show menu.
2. In the Query Rules menu, go to Query Rule Sets > *Your Rule Set* > *Your Rule Group*.
3. Under Child Query Rules, click the name of the rule.
4. Click the Pattern or Action tab.
5. Click the Enabled checkbox to enable or disable the pattern or action.



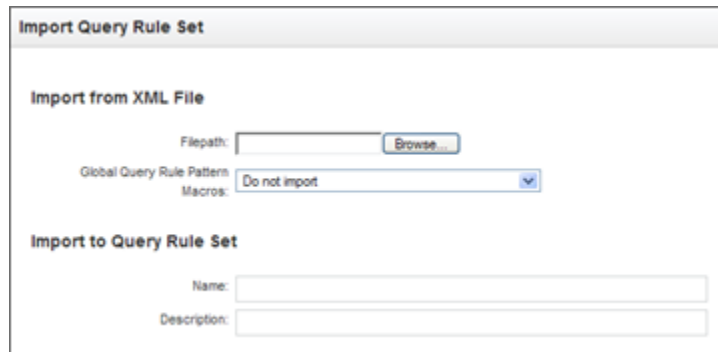
6. Click Save.

Importing Query Rules

Query rules can be created in an external XML file and then imported. The rules are imported as a new rule set. Use this feature to import query rules created in previous versions of ATG Search, or when moving your ATG Search installation.

To import query rule sets:

1. On the Search Workbench tab, select Query Rules from the Show menu.
2. Click Import Query Rule Sets.



3. Specify the location of the rule set to import.
4. Select whether or not to import macros as part of the rule set import. Imported macros overwrite any existing macros.
5. Provide a name and description for the new rule set.
6. Click Import.



Exporting Query Rules

To export your query rule sets to an XML file:

1. On the Search Workbench tab, select Query Rules from the Show menu.
2. In the Query Rules menu, right-click the query rule set, and select Export from the menu.



3. Check the Global Query Rule Pattern Macros box to export macro information as well as the rule set.
4. Click Export. The export uses your browser's file download defaults.

Moving Query Rule Sets and Rule Groups

To move query rule sets or rule groups:

1. On the Search Workbench tab, select Query Rules from the Show menu.
2. In the Query Rules menu, go to Query Rule Sets > *Your Rule Group*.
3. Right-click the rule set or rule group and select copy or cut from the menu.
4. Right-click the rule set or rule group under which you want to paste the moved item. You can paste the moved item either at the same level as the selected item, or as a child of that item.

Deleting Query Rule Sets

To delete a query rule set:

1. On the Search Workbench tab, select Query Rules from the Show menu.
2. Right-click the query rule set, and select Delete from the menu.
3. Click Delete to confirm.
4. Perform a full reindex of your content.



Deleting Query Rule Groups

To delete a query rule group:

1. On the Search Workbench tab, select Query Rules from the Show menu.
2. In the Query Rules menu, go to Query Rule Sets > *Your Rule Set*.
3. Right-click the query rule group, and select Delete from the menu.
4. Click Delete to confirm.


Deleting Query Rules

To delete a query rule:


1. On the Search Workbench tab, select Query Rules from the Show menu.
2. In the Query Rules menu, go to Query Rule Sets > *Your Rule Set* > *Your Rule Group*.
3. Right-click the query rule group, and select Delete from the menu.
4. Click Delete to confirm.

Deleting Patterns and Actions

To delete a pattern from a rule:

1. On the Search Workbench tab, select Query Rules from the Show menu.
2. In the Query Rules menu, go to Query Rule Sets > *Your Rule Set* > *Your Rule Group*.
3. Click the name of the rule to edit it.
4. On the Pattern tab, click the delete icon () for the pattern.
5. Click Save to permanently delete the pattern.

To delete an action from a rule:

1. On the Search Workbench tab, select Query Rules from the Show menu.
2. In the Query Rules menu, go to Query Rule Sets > *Your Rule Set* > *Your Rule Group*.
3. Click the name of the rule to edit it.
4. On the Action tab, click the delete icon () for the action.
5. Click Save to permanently delete the action.

Note: Rules must have at least one action.



14 Reporting

ATG Search includes several reports you can use to track how your end-users are using the product. This section describes the provided reports and how to use them. It includes the following sections:

[Accessing Reports](#)

[Running Reports](#)

[Data Logging for Reports](#)

[Search Reports](#)

For information on the database tables and ACI metadata used in ATG Search reports, see the [ATG Customer Intelligence Data Warehouse Guide](#).

Note: In order to use Commerce Search reports, your commerce site must use the ATG 10 version of the Search API, including form handlers and droplets. See the [ATG Search Query Guide](#) for information on these components. Commerce Search reports are described in the [Guide to ATG Commerce Reports](#).

Note: Information relating to deleted environments may still appear in search reports if you create an environment, use it, and then delete it before updating the data warehouse. In this case, the only identifying information for a deleted environment is the repository ID.

Accessing Reports

Users with the correct role assigned can access ATG Search reports via the Utilities menu in the Search Workspace or the ATG Reporting Center.

To access Search reports:

1. Go to the ATG Reporting Center.
2. Click ATG.
3. Search reports are organized into folders as shown. See the [Search Reports](#) section of this guide for information on specific reports. Many reports can be accessed through more than one folder. Some folders may also contain reports that are documented in the [Guide to ATG Commerce Reports](#).

▪ Customer Care > Search

[Most Frequent Search Terms](#)



- [Search Terms Grouped by Topic](#)
 - [Search Terms Not in the Dictionary](#)
 - [Search Terms with No Results](#)
- Marketing > Search Merchandising
 - [Most Frequent Search Terms](#)
 - [Search Terms Not in the Dictionary](#)
 - [Search Terms with No Results](#)
 - [Searches Trending Up and Down](#)
 - [Site Search Usage Overview](#)
 - Top Firing Search Configuration Rules—See the [Guide to ATG Commerce Reports](#)
 - Search > Search Activity
 - [All Selected Topics](#)
 - [Content Views by Topic](#)
 - [Most Frequent Search Terms](#)
 - [Most, Least, and Last Selected Topics](#)
 - [Most, Least, and Last Viewed Content](#)
 - [Search Terms Grouped by Topic](#)
 - [Search Terms Grouped by Type](#)
 - [Search Terms Grouped by Word Count](#)
 - [Search Terms Not in the Dictionary](#)
 - [Search Terms with No Results](#)
 - [Searches Trending Up and Down](#)
 - [Site Search Usage Overview](#)
 - Top Firing Search Configuration Rules—See the [Guide to ATG Commerce Reports](#)
 - Search > Search Input Detail
 - [Search Terms Consisting of Custom Terminology](#)
 - [Search Terms Existing Only as Thesaurus Entries](#)
 - [Search Terms Not in the Indexed Content](#)
 - [Search Terms with Numbers](#)
 - [Search Terms with Special Compound Terms](#)
 - Search > Search Utilization
 - [Search Activity Over Time](#)
 - [Search Activity Summary](#)
 - [Search Response Time Trends](#)



Running Reports

To run and view a report:

1. In the reports list, click the report that you want to run and view.
2. Make your selections for any prompts included for the report.

Note: If you have not created any registered sites in Site Administration, the Site prompt does not appear for any report. However, the report will still include a column (empty) for site information.

3. Click the Finish button.

A progress message is displayed while the report is being generated, and then the report is displayed. If a report takes a while to run, you can choose a different delivery method and run the report in the background by clicking Select a Delivery Method.

Data Logging for Reports

Search reports rely on data stored in the ATG Data Warehouse, described in detail in the [ATG Customer Intelligence Data Warehouse Guide](#). Before it can appear in a report, data must be collected in the form of log files, which are periodically loaded into the warehouse. This section describes the components, processes, and configuration involved.

Search Logging Components and Messages

Event messages are fired by the DAF. Search. Base component, retrieved by a message sink component, and eventually passed to the logging components in /atg/reporting/datacollection/search so that information is logged and can be reported on.

Logger Component	Message Listened For	Message Triggered By	Details Logged to File
UseSearch Environment Logger	BaseSearchMessage	User contacts a search environment for the first time (a Search server may include more than one environment).	Timestamp Profile ID Parent Session ID Repository Name (for profile) Language Code Environment ID Site Name Segment List



Logger Component	Message Listened For	Message Triggered By	Details Logged to File
QueryFileLogger	QueryMessage	The search engine responds to a user query.	Timestamp Profile ID Parent Session ID Repository Name (for profile) Language Code Environment ID Site Name Segment List Query ID Query Type Question Normalized Question Word Count Question Types Number Terms Unknown Word Terms Dictionary Adaptors Custom Terms Finder Terms Null Terms Nulx Terms Dictionary Adaptors Custom Terms Finder Terms Null Terms Nulx Terms Topics Selected Topics Search Configuration ID Search Rules Response Time Top Score Results Count Search Configuration Search Configuration Rules



Logger Component	Message Listened For	Message Triggered By	Details Logged to File
	Category Navigation Message	The user navigates through categories on the site.	Timestamp Profile ID Parent Session ID Repository Name (for profile) Language Code Environment ID Site Name Segment List Query ID Query Type Question Normalized Question Word Count Question Types Number Terms Unknown Word Terms Dictionary Adaptors Custom Terms Finder Terms Null Terms Nulx Terms Dictionary Adaptors Custom Terms Finder Terms Null Terms Nulx Terms Topics Selected Topics Search Configuration ID Search Rules Response Time Top Score Results Count Search Configuration Search Configuration Rules



Logger Component	Message Listened For	Message Triggered By	Details Logged to File
ViewContentFileLogger	ViewContentMessage	User views any document from search engine.	Timestamp Profile ID Parent Session ID Repository Name Language Code Environment ID Site Name Segment List Query ID Title URL Content Type
TopicFileLogger	PropertiesChangedEvent	INSERT, UPDATE, or DELETE performed on a topic.	Timestamp Repository Path Item Descriptor Name Item ID Event Type
EnvironmentFileLogger	PropertiesChangedEvent	INSERT, UPDATE, or DELETE performed on an environment.	Timestamp Repository Path Item Descriptor Name Item ID Event Type
ProjectFileLogger	PropertiesChangedEvent	INSERT, UPDATE, or DELETE performed on a search project.	Timestamp Repository Path Item Descriptor Name Item ID Event Type
SearchQueryCacheLogger	SearchClickThroughMessage	User clicks on a search result.	Search request and response stored in SearchQueryCache; see Commerce Search data logging in the ATG Commerce Programming Guide for additional information.

Many events log a Query Type. For all query languages, OneWordQuery and TwoWordQuery are the two basic and often the most common types. Each language has a language-specific set of query types. For queries in English, the following types are distinguished:

- HowMuchQuestion—Begins with the words “how much”
- HowManyQuestion—Begins with the words “how many”



- **HowToQuestion**—Begins with the word “how,” followed by a question word like “can” or “do,” followed by a pronoun like “I,” or just any query starting with “how to”
- **HowAdjectiveQuestion**—Begins with the word “how,” followed by an adjective, as in “How heavy is the X15?”
- **HowAdverbQuestion**—Begins with the word “how,” followed by an adverb, as in “How fast is the connection speed?”
- **HowQuestion**—Begins with the word “how”
- **WhatIsQuestion**—Begins with the words “what is”
- **WhatQuestion**—Begins with the word “what”
- **WhosQuestion**—Begins with the words “who is”
- **WhoQuestion**—Begins with the word “who”
- **WhereCanIFindQuestion**—Begins with the words “where can I find”
- **WhereQuestion**—Begins with the word “where”
- **WhyQuestion**—Begins with the word “why”
- **TellMeQuestion**—Begins with the words “tell me” or with the word “describe”
- **CanIQuestion**—Begins with the words “can I”
- **YesNoQuestion**—Begins with an auxiliary verb, like “Am I...” or “Can you...”
- **Statement**—Any query beginning with determiners or possessive pronouns, including in English: “a” “an” “the” “this” “that” “my” “our” “your” etc.
- **TwoWordQuery**—Any other query that is two words long
- **OneWordQuery**—Any other query that is one word long
- **Other**—Any other query

Other languages have a similar set of specific query types, usually identified by their initial words.

Note: To avoid skewing reporting results, the Search Testing feature of ATG Merchandising suppresses the JMS messages normally sent when a response is received to a Search query. See the *Detecting Web Spiders* section in the [ATG Search Query Guide](#) for information.

The `formatFields` property of the logging components controls the format of logged data.

Enabling and Disabling Search Logging

If you are using ATG Search with ATG Commerce, configure the `/atg/commerce/search/catalog/QueryRequest` component as shown:

```
    firingSearchEvents=true  
    saveRequest=true
```

If you are using ATG Search with ATG Knowledge, configure the following properties files as shown:

```
/atg/search/routing/command/search/BrowseRequest.properties
firingSearchEvents=true
```

```
/atg/search/routing/command/search/CategorizeRequest.properties
firingSearchEvents=true
```

```
/atg/search/routing/command/search/QueryRequest.properties
firingSearchEvents=true
```

```
/atg/search/routing/command/search/SimilarDocsRequest.properties
firingSearchEvents=true
```

```
/atg/search/routing/command/search/StructuredQueryRequest.properties
firingSearchEvents=true
```

```
/atg/search/routing/command/search/ViewDocumentRequest.properties
firingSearchEvents=true
```

Configuring Log File Location

By default the ATG Search data loggers write data to, and the warehouse loaders read event logs from, the `logs` directory of the server on which each component is running. This default location works if the loggers and loaders are running in the same server, but if they are on different servers, the loader components cannot find the log files.

For example, if you are using ATG Search with ATG Knowledge and ATG Self Service, you could have three servers named `agent` (running ATG Knowledge), `customer-facing`, (running ATG Self Service), and `management` (running Search Administration and the `SearchAdmin.datawarehouse` module). By default, log files for ATG Knowledge and ATG Self Service search activities are written to the `agent/logs` and `customer-facing/logs` folders. The data warehouse loaders in ATG Search look for the log files in `management/logs`. In this case, you could configure `defaultRoot` on the `agent` and `customer-facing` servers to point to `management/logs`, or configure that property on all three servers to point to some other location.

Search Loader Components

The data loggers write information about events to tab-delimited text files. These files are then loaded into the data warehouse by the following loader components in `/atg/reporting/datawarehouse/loaders/`:

```
EnvironmentLoader.properties
ProjectLoader.properties
SearchQueryLoader.properties
TopicLoader.properties
UseSearchEnvironmentLoader.properties
ViewContentLoader.properties
```



Scheduling Data Loading

Log files are rotated by their logger components, either at a scheduled time or when a configured limit on the number of log entries is reached. During rotation, the currently open log file is closed and a new one opened. The logging component fires a JMS message, which is received by a message sink, which adds the closed log file and its type to a queue.

Note: Rotation occurs only if the current log file contains data.

You can schedule file rotation using the `schedule` property of each logging component (see [Search Logging Components and Messages](#)). For example, to rotate the `UseSearchEnvironmentLogger` properties set the following:

```
schedule=every 1 hour without catch up
```

For information on how to specify a schedule, see the *Scheduler Services* section of the *Core Dynamo Services* chapter in the [ATG Programming Guide](#).

You can also use the `dataItemThreshold` of each component to trigger log rotation when the specified number of log records is exceeded. If you set both properties, the log will be rotated whenever it exceeds the `dataItemThreshold` and at the scheduled time, as long as the log file contains at least one log entry.

Search Reports

This section describes the reports provided with ATG Search. Reports are listed alphabetically; for information on where to find the reports in the ACI folder structure, see [Accessing Reports](#).

All Selected Topics

Use this report to see the complete list of topics users select in their search results, and how many of those selections lead to viewing content. The prompts are:

- Time Period
- Language Used in Search
- Environment
- Site

The report results are a table, which includes the following information:

- Topic—The topics that were selected
- Site—Site searched
- Selection Count—How many times the topics were selected
- Percentage of Total—The percentage of topic selections in this report represented by this topic
- View Content Count—Number of selections that resulted in the user viewing content



Topic	Selection Count	Percentage of Total	View Content Count
Hardware	5	83.3333%	2
Software Sub Topic1	1	16.6667%	0

All Selected Topics Report

Content Views by Topic

Use this report to see which content your users are accessing. The prompts are:

- Time Period
- Language
- Site
- Environment

The report results are a series of tables. For the Most, Least, and Last Selected, and for Content views by topic, the tables include the following:

- Content (or topic, in the case of Content views by topic)
- View Content Count

Topic	View Content Count
Hardware	2

Content Views by Topic Report

Most Frequent Search Terms

This report allows you to see what information on your site end-users are searching for.

This report uses the following prompts:

- Time Period
- Language
- Top N/Search Groups
- Site
- Environment

The report output is a table, which includes the following information:



- Search Group—The normalized search text. Click a search group to view the individual searches that were normalized to form a group (there may be only one search in a group)
- Site—The site searched
- Search Count—Number of times the search was used
- Percentage of total—The percentage of searches in your report represented by this search group

Search Group	Site	Search Count	Percentage of Total
lamp	ATG Store US	7	28.0000%
		7	28.0000%
shirt	ATG Store US	3	12.0000%
		3	12.0000%
password	ATG Store US	2	8.0000%
		2	8.0000%
shoe	ATG Store US	2	8.0000%
		2	8.0000%
skirt	ATG Store US	2	8.0000%
		2	8.0000%
sock	ATG Store US	2	8.0000%
		2	8.0000%

Most Frequent Search Terms Report

Most, Least, and Last Selected Topics

Use this report to show which topics are most and least interesting to end-users. The prompts are:

- Time Period
- Number of Topics to Display
- Language
- Site
- Environment

The report results are a series of tables. For the Most, Least, and Last Selected, and for Topic Selections, the tables include the following:

- Topic—The topics that were selected
- Site—The site searched
- Selection Count—How many times the topics were selected
- Percentage of Total—The percentage of topic selections in this report represented by this topic



- View Content Count—Number of selections that resulted in the user viewing content

Topic	Site	Selection Count	Percentage of Total	View Content Count
Other	Agent Site	2	28.5714%	1
	ATG Store US	1	14.2857%	1
		3	42.8571%	2
Event Tickets	Agent Site	1	14.2857%	1
	Event Site Help	1	14.2857%	1
		2	28.5714%	2
Music Purchases	Music Site Help	2	28.5714%	0
		2	28.5714%	0

Most Selected Topics section of Most, Least, and Last Selected Topics Report

Most, Least, and Last Viewed Content

Use this report to see which content your users are accessing most frequently, or which content they should be made more aware of, and other usage information. The prompts are:

- Time Period
- Number of Content Items to Display
- Language
- Site
- Environment

The report results are a series of tables. For the Most, Least, and Last Selected, and for Content views by topic, the tables include the following:

- Content (or topic, in the case of Content views by topic)
- Site searched
- View Content Count



Content	Site	View Content Count
Cancel Tickets at any Time	Music Site Help	2
	Agent Site	1
		3
Recordings are Available in a Variety of Electronic Formats	Event Site Help	1
		1
Return Recordings for any Reason	Agent Site	1
		1

Most Viewed Content Section of Most, Least, and Last Viewed Content Report

Search Activity Over Time

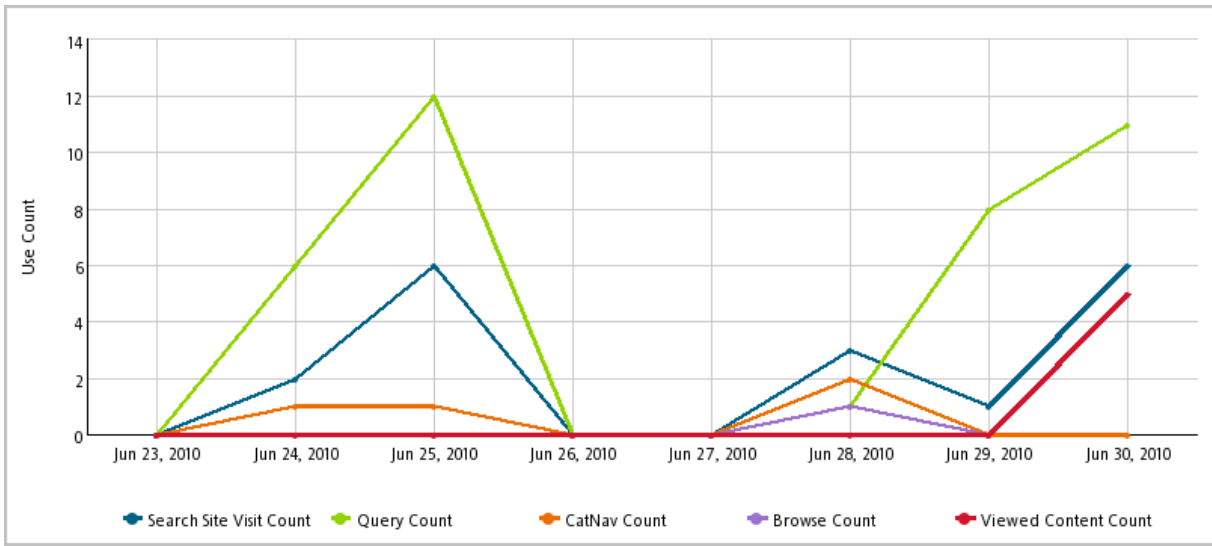
This report provides information about ATG Search usage trends, including sessions, searches, and results viewed.

The prompts for this report are:

- Time Period
- Results Interval
- Language
- Site
- Environment

The report results are a line graph and table showing the following information:

- Site—Site searched
- Sessions—Number of user sessions
- Search by Text—Number of text searches
- Browse by Topic—Topics are primarily used with ATG Service applications
- Navigate by Category—Categories are primarily used on ATG Commerce sites
- Search Results Viewed—Number of search results users viewed



Search Activity Over Time Report

Day (shows accumulated data for each day)						
		Sessions	Search by Text	Navigate by Category	Browse by Topic	Search Results Viewed
Jun 23, 2010		0	0	0	0	0
		0	0	0	0	0
Jun 24, 2010	Unregistered	1	0	1	0	0
	ATG Store US	1	6	0	0	0
		2	6	1	0	0
Jun 25, 2010	Unregistered	1	0	1	0	0
	ATG Store US	4	11	0	0	0
	ATG Home	1	1	0	0	0
		6	12	1	0	0

Search Activity Over Time Report Table

Search Activity Summary

This report provides summary data on sessions, search request frequency, and response times.

The prompts for this report are:

- Time Period
- Language
- Site
- Environment



The report results are a table for each site, showing the following information for each type of search:

- Total
- Counts per Session
- Average Counts per Day
- Average Response Time (sec)

ATG Home				
	Search by Text	Browse by Topic	Navigate by Category	Session
Total	1	0	0	1
Counts per Session	1.00	0.00	0.00	-
Average Counts per Day	0.07	0.00	0.00	-
Average Response Time (sec)	0.292	0.000	0.000	-

ATG Store US				
	Search by Text	Browse by Topic	Navigate by Category	Session
Total	32	1	0	8
Counts per Session	4.00	0.12	0.00	-
Average Counts per Day	2.29	0.07	0.00	-
Average Response Time (sec)	1.403	0.051	0.000	-

Search Activity Summary Report

Search Response Time Trends

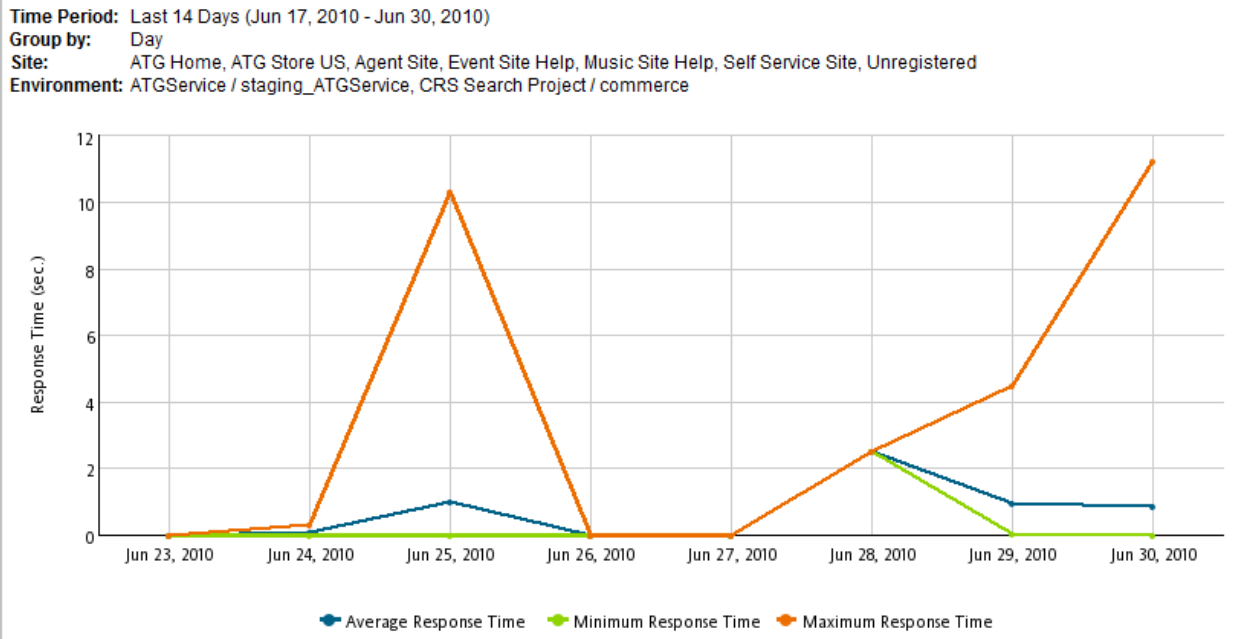
This report provides information about search response times.

The prompts for this report are:

- Time Period
- Results Interval
- Language
- Site
- Environment

The report results are a line graph and table showing the following information for all sites:

- Minimum Response Time
- Maximum Response Time
- Average Response Time



Search Response Time Trends Report

Search Terms Not in the Dictionary

This report provides information about end-user searches that include words that are not in the ATG Search dictionary. You can use this report to refine your dictionary.

The prompts for this report are:

- Time Period
- Language
- Site
- Environment

The report results are a table, which includes the following information:

- Unknown Word—Specific word searched for, that does not appear in the dictionary; click the word to see the full text of user searches that contained that word
- Site—Site searched
- Search Count—Number of times the word appeared in end-user searches
- Percentage of Total—Percentage of searches in your report that included this particular word



Unknown Word	Site	Search Count	Percentage of Total
ugli	ATG Home	1	33.3333%
	ATG Store US	1	33.3333%
		2	66.6667%
moleskin	ATG Store US	1	33.3333%
		1	33.3333%

Search Terms Not in the Dictionary

Search Terms Grouped by Topic

This report shows end-user searches that were associated with topics. The prompts for this report are:

- Time Period
- Topic
- Language
- Site
- Environment

The report results are a table, which shows the following information:

- Topic—The topic with which the end-user search was associated; click the topic name to see the actual text of all searches that resulted in this topic
- Site—Site searched
- Search Count—Number of times this search-topic combination occurred

Search Terms Grouped by Type

This report groups end-user searches according to the type of question. There are many possible types, including Statement, YesNoQuestion, HowTo Question, and OneWordQuestion.

The prompts for this report are:

- Time Period
- Language
- Site
- Environment

The report results are a table, which includes the following information:

- Search Type—The type of search performed; click the search type to see the text of the end-user searches
- Site—Site searched



- Search Count—Number of searches that fall into this search type
- Percentage of Total—Percentage of the searches in this report that are of this type

Search Type	Site	Search Count	Percentage of Total
OneWordQuery	ATG Store US	32	84.2105%
	Event Site Help	2	5.2632%
	Music Site Help	2	5.2632%
	ATG Home	1	2.6316%
		37	97.3684%
TwoWordQuery	Agent Site	1	2.6316%
		1	2.6316%

Search Terms Grouped by Type Report

Search Terms Grouped by Word Count

This report breaks down end-user searches according to how many words each search contains. The prompts for this report are:

- Time Period
- Language
- Site
- Environment

The report results are a table, which includes the following information:

- Word Count—The number of words in the search; click the number to view the text of the user searches that had this word count
- Site—Site searched
- Search Count—Number of times a search was made with this word count
- Percentage of Total—Percentage of searches in the report with this word count

Word Count	Site	Search Count	Percentage of Total
2	Agent Site	1	2.6316%
		1	2.6316%
1	ATG Store US	32	84.2105%
	Music Site Help	2	5.2632%
	Event Site Help	2	5.2632%
	ATG Home	1	2.6316%
		37	97.3684%

Search Terms Grouped by Word Count Report



Search Terms Consisting of Custom Terminology

This report shows query terms that were used as part of a search, are not in the core ATG Search dictionary, but were found in either a dictionary adaptor or were added as a custom term to the search dictionary. The prompts for this report are:

- Time Period
- Dictionary Adapter—Use to restrict the report results to terms found in a particular adapter
- Language
- Site
- Environment

The report results are a table, which includes the following information:

- Custom Term—The custom term that appeared in the search; click the term to view the text of the search
- Site—Site searched
- Search Count—Number of times a search was made with this custom term
- Percentage of Total—Percentage of searches in the report with this custom term

Search Terms with No Results

This report identifies the top search groups for which no search results were returned. Use this report to refine your index, for example by adding commonly searched-for synonyms to your dictionary. The prompts for this report are:

- Time Period
- Language
- Site
- Environment

The report results are a table, which includes the following information:

- Search Group—The normalized search text; click a search group to view the individual searches that were normalized to form a group
- Site—Site searched
- Search Count—Number of times the search was attempted
- Percentage of Total—Percentage of searches in your report represented by this search group



Search Group	Site	Search Count	Percentage of Total
lamp	ATG Store US	2	20.0000%
		2	20.0000%
password	ATG Store US	2	20.0000%
		2	20.0000%
sock	ATG Store US	2	20.0000%
		2	20.0000%
bulldoze	ATG Store US	1	10.0000%
		1	10.0000%

Search Terms with No Results Report

Search Terms with Numbers

This report provides information about end-user searches that include numbers. Use this report to refine your index, for example by adding commonly searched-for numbers to your dictionary.

The prompts for this report are:

- Time Period
- Language
- Site
- Environment

The report results are a table, which includes the following information:

- Search Number—Number searched for; click the number to see the full text of user searches that contained that number
- Site—Site searched
- Search Count—Number of times the number appeared in end-user searches
- Percentage of Total—Percentage of searches in this report that included this particular number

Search Number	Site	Search Count	Percentage of Total
42	ATG Store US	3	60.0000%
		3	60.0000%
100	ATG Store US	1	20.0000%
		1	20.0000%
12	ATG Store US	1	20.0000%
		1	20.0000%

Search Terms with Numbers Report



Search Terms with Special Compound Terms

This report provides information about end-user searches that include compound terms such as alphanumeric combinations.

The prompts for this report are:

- Time Period
- Language
- Site
- Environment

The report results are a table, which includes the following information:

- Special Compound Term—Term appearing in the query
- Site—Site searched
- Search Count—Number of times the term appeared in end-user searches
- Percentage of Total—Percentage of searches in this report that included this particular term

Special Compound Term	Site	Search Count	Percentage of Total
a1005	ATG Store US	1	50.0000%
		1	50.0000%
t510	ATG Store US	1	50.0000%
		1	50.0000%

Search Terms with Special Compound Terms Report

Search Terms Existing only as Thesaurus Entries

This report provides information on query terms that are not in the indexed content, but which do have thesaurus entries.

The prompts for this report are:

- Time Period
- Language
- Site
- Environment

The report results are a table, which includes the following information:

- Thesaurus Term—Term appearing in the query; click the term to view the term for which the user tried to search.



- Site—Site searched
- Search Count—Number of times the term appeared in end-user searches
- Percentage of Total—Percentage of searches in this report that included this particular term

Thesaurus Term	Site	Search Count	Percentage of Total
search	ATG Store US	1	100.0000%
		1	100.0000%

Search Terms Existing Only as Thesaurus Entries Report

Search Terms Not in the Indexed Content

This report provides a list of terms that are used in queries, but which do not appear in the indexed content and for which there are no thesaurus entries.

The prompts for this report are:

- Time Period
- Language
- Site
- Environment

The report results are a table, which includes the following information:

- Unretrievable Term—Term appearing in the query; click the term to view the full query
- Site—Site searched
- Search Count—Number of times the term appeared in end-user searches
- Percentage of Total—Percentage of searches in this report that included this particular term



Unretrievable Term	Site	Search Count	Percentage of Total
bulldozer	ATG Store US	2	20.0000%
		2	20.0000%
password	ATG Store US	2	20.0000%
		2	20.0000%
socks	ATG Store US	2	20.0000%
		2	20.0000%
cool	Agent Site	1	10.0000%
		1	10.0000%
kiwi	ATG Store US	1	10.0000%
		1	10.0000%
moleskin	ATG Store US	1	10.0000%
		1	10.0000%
winter	Agent Site	1	10.0000%
		1	10.0000%

Search Terms Not in the Indexed Content

Searches Trending Up and Down

This report shows which searches have seen the most increase in use vs. those with the greatest drop-off in use across the two time periods selected.

The prompts for this report are:

- Time Periods to Compare—Use the calendars to select two dates. Those mark the starting points for the two time periods you are comparing. In the Number of Days to Compare field, enter the number of days worth of data to consider in the report. For example, select June 1 and December 1 on the calendars, then enter 14 in the Number of Days field, to compare searches in the first two weeks of June to those in the first two weeks of December.
- Number of Items to Display
- Language Used in Search

The report results are two tables, one showing upward movement, the other downward, for all of the selected sites. For each table, the results include:

- Search Group—Term appearing in the query; click the term to view the full query
- Use Count—Number of times the search was made
- % Change—Change in position for the search group

Site Search Usage Overview

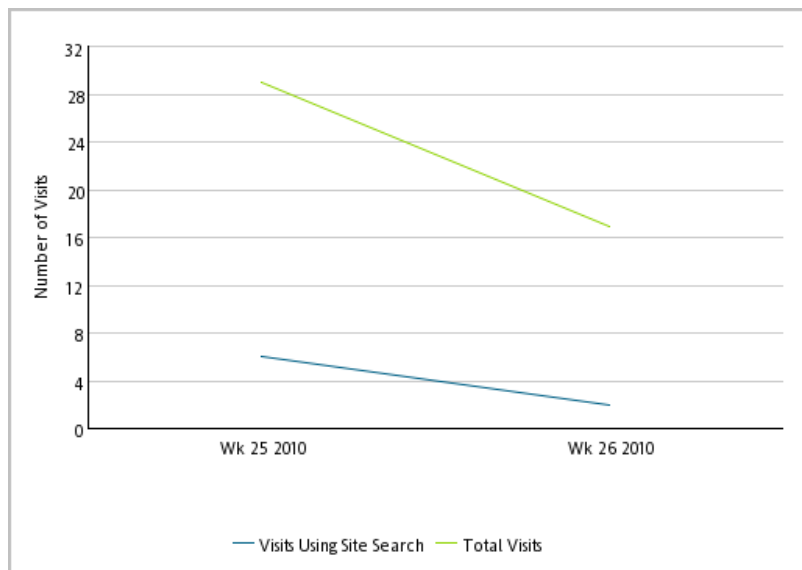
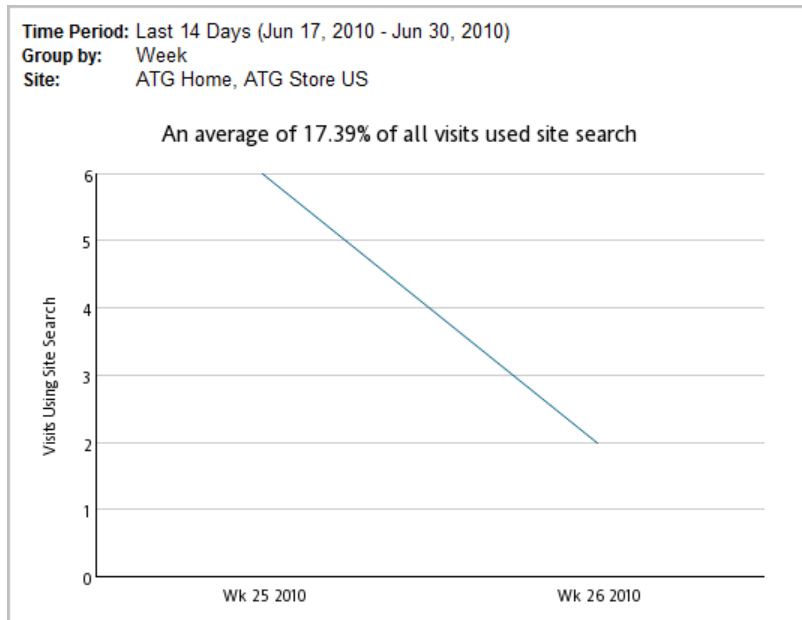
This report allows you to compare the number of visits to your site overall with the number of visits where users made searches.



This report uses the following prompts:

- Time Period
- Time Interval

The report results are two graphs; one graph shows searches only, the other, searches in relation to total site visits.





15 Troubleshooting and Performance Tuning

This chapter provides pointers you can use to tune and troubleshoot your ATG Search installation.

[Finding Debugging Information](#)

[Troubleshooting Repository Indexing](#)

[Troubleshooting Connection Problems](#)

[Troubleshooting Search Results](#)

[Search Error Messages](#)

[Search Performance Tuning](#)

[Updating Search Settings on a Live Site](#)

Finding Debugging Information

This section describes tools that can help you answer fundamental questions about the state of your installation. This information is often necessary before taking any further steps, or before contacting ATG support.

ATG Search Administration

You can use Search Administration to determine whether your index was deployed, and whether it contains data.

The Index History tab provides the status for all of your indexing tasks, the number of items in the index, and the environment to which the index was deployed.



Index History

Indexing Task	Start	End	Status	Errors	Items	Size	Deployed To	
Complete	07/18/2007 4:03 PM	07/18/2007 4:05 PM	Succeeded	0	1122	14	commerce	
Complete	07/18/2007 3:59 PM	07/18/2007 4:01 PM	Succeeded	0	1122	14	commerce	
Complete	07/18/2007 3:52 PM	07/18/2007 3:54 PM	Succeeded	0	1122	14	commerce	
Complete	07/18/2007 3:51 PM	07/18/2007 3:52 PM	Failed	1	0	0	commerce	
Complete	07/18/2007 3:07 PM	07/18/2007 3:07 PM	Cancelled	0	0	0	commerce	
Complete	07/18/2007 2:23 PM	07/18/2007 2:32 PM	Cancelled	0	0	0	commerce	

If the index did not deploy, consider the following possible causes:

- Is the DeploymentShare property correctly configured in /atg/search/routing/Configuration component?
- Is enough space available on the deployment share for the index?
- Are ports configured correctly in the RoutingSystemService component?

You can also get information about your search environments from the Environments page for your search project.

Environment	Primary	Hosts	Status	
commerce		1	Running	Stop
indexing environment		1	Stopped	

If the environment is not running, you can start it manually. If the environment does not start, you can view the host configuration and check it for errors.

If you have trouble launching one or more search engines, try the following remedies:

- If the search engine is standalone, set the /atg/search/routing/Configuration component's engineDir property to the absolute path of the search engine directory.
- Set the LaunchingService component's LoggingDebug property to true and look at the Launching: debugging line. You can then try the same arguments manually from the command line.

SearchEngineService

The SearchEngineService component gives you immediate access to ATG Search status information that is also included in the log files. The information includes the following:

- Engine status (stopped or started)
- Current engine activity, including response times
- Commands per second
- Engine failures



To view the SearchEngineService, go to
`http://host:port/dyn/admin/nucleus/atg/search/routing/SearchEngineService` (see the [ATG Installation and Configuration Guide](#) for information on ATG Dynamo Server Admin).

Enabling Search Logging

In addition to the application server logs, ATG Search includes two additional log file types:

- Search activity logs. ATG Search receives requests from and returns responses to the user interface in XML form; these activity files include detailed information about the search.
- Search engine logs. The engine logs can help determine whether an engine is functioning properly.

Note: Log files can quickly consume a great deal of space. Either clean out the logging directory often, or turn logging off when it is not required.

Enabling Search Activity Logs

The search activity files record the request input and response output for an engine. Errors in the response files can help with troubleshooting, and a missing response might indicate a timeout or other communication problem.

By default, the engine activity files log only request and response pairs corresponding to errors. You can use ATG Dynamo Server Admin to enable logging for other activities. To enable logging:

1. Go to `http://host:port/dyn/admin/nucleus/atg/search/routing/Configuration` (see the [ATG Installation and Configuration Guide](#) for information on ATG Dynamo Server Admin).
2. Change the `LoggingAllRequests` property value to `true` to log all requests made to the search engine. You can also set the following individual properties to `true` to log only certain request types:
 - `LoggingDeploy`
 - `LoggingIndex`
 - `LoggingLoad`
 - `LoggingSearch`
 - `LoggingStart`
 - `LoggingStop`

The indexing server writes out request and response XML to the following directory:

```
<ATG10dir>\home\logs\searchEngineActivity
```

The names of the log files provide information about the contents. For example, the log file name `0006320004_m_2009-04-21-101254_26200058_mymachine_6100_search_request.xml` includes the following:



- 0006320004—Request ID
- m—Indicates that the file is a merge file, representing the final results returned to the user. The merge files do not include a partition, host, or port number, since they include data from multiple partitions, and potentially multiple hosts.
- 2009-04-21-101254—Timestamp in YYYY-MM-DD-HHMMSS format
- 26200058—Partition ID
- mymachine—Host name
- 6100—Port number
- search—Request type
- request—File type

To change the directory where activity files are stored, edit the `/atg/search/routing/Configuration.engineNameActivityPath` property value and set it to the directory you want to use.

You can view the most recent activity logs directly in ATG Dynamo Server Admin. To view activity logs, navigate to

`http://host:port/dyn/admin/nucl eus/atg/search/routing/SearchEngineService`. Note that the log history visible in ATG Dynamo Server Admin is reset when the application server or search engine is restarted; however, files in the `SearchEngineActivity` directory are not affected by restarts.

Enabling Search Engine Logs

Each search engine has an associated engine log file. Errors in the log files can help with troubleshooting, and a missing log file can indicate a problem with starting that engine.

The engine log contains platform, build, and system information at the beginning of the file:

```

2009/05/04 15: 56: 14 : LOGGI NG : Server : 0 :      Platform: x86-Windows32
2009/05/04 15: 56: 14 : LOGGI NG : Server : 0 :      Build: ATGSearch_10_090504-
002806
2009/05/04 15: 56: 14 : LOGGI NG : Server : 0 :      Unicode Support: utf8
2009/05/04 15: 56: 14 : LOGGI NG : Server : 0 :      Processors: 2
2009/05/04 15: 56: 14 : LOGGI NG : Server : 0 :      Processor speed (MHz): 2993
2009/05/04 15: 56: 14 : LOGGI NG : Server : 0 :      Total physical memory (MB): 4029
2009/05/04 15: 56: 14 : LOGGI NG : Server : 0 :      Avail physical memory (MB): 1399
2009/05/04 15: 56: 14 : LOGGI NG : Server : 0 :      Total swap space (MB): 6238
2009/05/04 15: 56: 14 : LOGGI NG : Server : 0 :      Avail swap space (MB): 3744
    
```

The platform and build information can help you determine whether the correct version of the engine is installed. The system information can show whether the system has sufficient processors, physical memory, and swap space.

A search engine log also contains the index filename, partition name, and the Content ID (also known as Partition GUID) associated with it, as shown in the following example:



```
2009/05/04 15:56:19 : INFO : Server : 0 : Starting copy of
'C:\ATG\ATG10\Search10\SearchEngine\x86-win32\data\initial.index' to
'../indexFiles\400001\initial.index'
```

```
2009/05/04 15:56:25 : INFO : Server : 0 : Content ID: 0fd62287-c697-4fe4-
8639-716e2c0e2f07
```

This information can show whether the path to the deployment share directory is correct, and can be used to match the partition listed in the search activity logs.

Each step in the indexing process generates a separate log; a full index task generates a minimum of three logs, corresponding to the estimation, indexing, and deployment steps. If the index spans multiple physical partitions, each additional partition generates additional logs for the indexing and deployment steps.

For an estimation or indexing step, the search engine is launched with an initial partition. This is reflected in the logs by the index filename `initial.index`. For a deployment step, the engine log starts with an existing partition. This is reflected in the logs by filenames with `idx` and `stg` extensions. The content ID from the deployment step matches that of the prior indexing step; however the estimation step uses a different content ID. See the following example:

```
2009/05/04 15:57:14 : INFO : Server : 0 : Starting copy of 'C:\ATG\ATG10\
Search10\bol.t.100002.ContentSet1.400020.idx' to ' ../indexFiles\400003\
bol.t.100002.ContentSet1.400020.idx'
```

```
2009/05/04 15:57:14 : INFO : Server : 0 : Starting copy of 'C:\ATG\ATG10\
Search10\bol.t.100002.ContentSet1.400020.stg' to ' ../indexFiles\400003\
bol.t.100002.ContentSet1.400020.stg'
```

```
2009/05/04 15:57:15 : INFO : Server : 0 : Content ID: 9b6342d1-bfb2-4010-
992c-9793c7ff0cae
```

The default logging level is `I` (information). This level includes warnings and errors, but does not include requests and responses from the engine. Set the logging level to `d` (debug) to include the request and response XML in the output, as in the following example:

```
2009/05/04 18:11:09 : DEBUG : Server : 0 : request RequestXML: <query
minScore='300' RQText='ANSWER' exclusion=' ' sorting='SCORE' suggestion=
'TERMSET' version='2.0.1' autoRefine='false' docSetSort='None' debug='0' >
```

```
2009/05/04 18:11:09 : DEBUG : Server : 0 : response RequestXML: <?xml
version="1.0" encoding="utf-8"?>
```

```
<answer contentID="d2127cff-6209-4e3a-8ed4-
3800461c33ad" sorting="score" docSetSort="None"
```

To enable engine logging:



1. Go to `http://host:port/dyn/admin/nucleus/atg/search/routing/Configuration` (see the *ATG Installation and Configuration Guide* for information on ATG Dynamo Server Admin).
2. Change the `engineLogLevel` property value to `D` to enable debug logging. This will log all requests and takes effect upon the next engine restart.

The search engine writes out files named `atgsearch_*.log` to the following directory:

```
<ATG10dir>\Search10\SearchEngine\platform\bin
```

To change the directory where engine log files are stored, create a `filelog_basepath` environment variable and set it to the directory you want to use.

By default, engine log files are retained for two days. To configure how long the files are retained, edit the `params` property value of the `/atg/search/routing/LaunchingService.properties` file and add a `-t` flag as shown:

```
params=atgsearch, -p, $port$, -s, AESoapWaspConfig.xml, -a, $AEConfigXml$, -m, Messages.xml, -l, $loglevel$, -t, ///
```

Controlling Slow Response Logging

Among other data, the search logs include warning information on search requests that take an unusual amount of time to process. An example of the warning logged is shown:

```
2009/08/03 13:34:13 : WARNING : Server : 0 : Slow request (5.18):
<query minScore='300' RQText='ANSWER' exclusion='' sorting='SCORE'
suggestion='TERMSET' version='2.0.1' autoRefine='false' docSetSort='None'
debug='0' sleepTimeMs="5000">
```

The `Slow request (time)` element shows the actual search processing time. Processing time excludes time spent in the queue before execution, so total elapsed time for the search could be longer under certain conditions.

If your logs are cluttered with extraneous warnings, or you want to troubleshoot particular slow searches, you can raise or lower the logging threshold for slow responses in the `AEConfig.xml` file located in `<ATG10dir>/Search10/SearchEngine/platform/bin`.

```
<SlowRequestThreshold>5000</SlowRequestThreshold>
```

The default is 5000 milliseconds.

Enabling Auxiliary Data Error Reporting

ATG Search includes several sorts of auxiliary data, including type-ahead and stock availability data. By default, if you have a multi-partition index, ATG Search suppresses errors that occur during loading for auxiliary data. If you have multiple partitions and suspect that this data is causing problems, turn on error reporting to see any issues.



To enable auxiliary data error reporting, set the enableAuxiliaryDataErrorsSuppress property of the RoutingIndexService component to false.

Search Engine Performance Logging

ATG Search automatically balances load across your engines. If you suspect your search engines may be under- or over-utilized, you can check values in the search response XML.

```
<answer ... maxThreads="10" loadFactor="0.10" throughputQps="0.13"
  avgResponseTimes="74.60" expectedResponseTimes="149.21">
```

The loadFactor value provides a summary of search engine load. The formula for calculating the loadFactor is:

$$(\text{request_queue_length} + \text{active_threads}) / \text{max_active_threads}$$

A load factor greater than 1 means that the engine is overloaded at the time of the request. Load information is periodically logged; see Controlling Load Logging in this chapter for information.

The throughputQps (queries per second), avgResponseTimes, and expectedResponseTimes can also provide information on how well the server is performing.

If you have reason to be concerned about your search server load, you can adjust the settings used to control how often load information is recorded in the search logs. To change this setting, configure the PerformanceLoggingPeriod element in the AECConfig.xml file located in <ATG10dir>/Search10/SearchEngine/platform/bin. For example:

```
<PerformanceLoggingPeriod>600</PerformanceLoggingPeriod>
```

The default is to log every 10 minutes.

Using Diagnostics Capture

Diagnostics capture allows you to easily take snapshots of your diagnostic information for troubleshooting purposes.

1. On your Search Administration instance, go to `http://hostname:port/dyn/admin/nucleus/atg/search/routing/RoutingSystemService/`.
1. Click the RoutingSystemService Capture Diagnostics link.



[Service /atg/search/routing/RoutingSystemService/](#)

Class [atg.search.routing.RoutingSystemService](#)

[View Service Configuration](#)

Service Info

[RoutingSystemService Capture Diagnostics](#)

2. Select the data you want to capture.

Select options for capturing diagnostics

Capture the following information on all routing and launcher instances:

- System Summary
- Engine Logs
- Engine Configuration Files
- Engine Activity files from all Routing Instances
- Application Server Logs
- Archived Application Server Logs (Warning: this option may result in a very large capture file.)

Retrieve the current index files for the following environments:

- indexing environment
- commerce

- System Summary—An index.html file and child pages describing the Routing system, providing a simplified version of what you see in RoutingSystemService and SearchEngineService. At the top-level are the environments, with the current index, hosts, and engines. The engine pages include the initial and recent commands for every Routing instance at the time of the capture.
- Engine Logs—All engine logs.
- Engine Configuration Files—All XML configuration files for those machines with engines.
- Engine Activity Files—Contents of the searchEngineActivity directory from every Routing instance.
- Application Server Logs—All application server logs from every Routing instance.
- Archived Application Server Logs—All archived application server logs from every Routing instance.
- Current Index files for Selected Environments—Check off an environment to include the index files (.idx and .stg) currently served by the environment. If



the files are missing from the deploymentShare directory, then they will not be included. It is not possible to include old indexes.

Note that all Routing instances must be running; if a Routing instance is not running, you cannot gather diagnostic information from that instance.

Note that the more information captured, the larger the archived files will be; indexes and application server logs are particularly large, but are also extremely important information sources. Capture only the necessary information, and be sure that your deployment share has enough space.

3. Click Capture. The files are zipped together and saved in your deployment share directory.

Once the archives are no longer needed, you can delete the files to save space.

Viewing Index Data

The Index Review pages in Search Administration provide statistics about the index as a whole and about individual content items. You can use this information for troubleshooting; for example, if a search does not return the expected results, has the index size changed? Are all expected languages and file formats included? Has the customization data changed? Are the expected content items included in the index? See the [Using Index Review](#) section for information.

You can also use IndexReport, a command-line utility that reports on index data. IndexReport allows you to answer questions such as:

- Did customization data load?
- What content is in this index?
- What text fields or meta-properties were indexed?

To use IndexReport, in a command-line window go to <ATG10dir>/Search10/SearchEngine/operating_system/bin and enter the following command:

```
IndexReport [-html] [-o outputDir] [-c contentFile] [-r requestFile]
idxFile
```

For syntax and usage information, enter the following command:

```
IndexReport -help
```

The name of the IDX file you want to examine is a required parameter, and must include the .idx file extension. IDX files are located at

<ATG10dir>/Search10/SearchEngine/operating_system/indexFiles.

Note: Index files are operating-system-specific. You cannot run IndexReport against a file created by an engine on a different operating system.



Troubleshooting Repository Indexing

If you run into problems specifically with indexing repositories, rather than file system content, you can try the following methods to locate the problem's source:

- Check for warnings or errors from your `IndexingOutputConfig` at startup.
- Use the `bulkload` property of the `IndexingOutputConfig` component and a `FileDocumentSubmitter` to capture XHTML to a directory. To capture XHTML with the `SearchAdmin` module, turn on `echoDocumentsToStdout` on the `BulkLoader` or `Incremental Loader`.
- If you get no documents when using the ATG Commerce `ProductCatalogOutputConfig` component, make sure that the `Catalog Maintenance Service` has been run.

Troubleshooting Connection Problems

The information in this section concerns situations in which search routing appears to be correctly configured, but the production site is unable to connect to the search environment. The main symptom of this problem is that you are unable to perform a search from the production site; you may also encounter an "environment not found" error.

When attempting to diagnose a connection problem, first perform the following checks:

1. Check the ATG console for startup errors related to repositories or lock managers.
2. Verify the database host and schema that contains the shared search repository data

If neither step makes the source of the problem clear, continue through the checks described in the sections that follow.

Check Data Source Configuration

One possible reason for search failure is an incorrectly configured data source. To examine your data source configuration, you can use the ATG Dynamo Server Admin (see the [ATG Installation and Configuration Guide](#) for information on this tool) to view each ATG instance in which the DAF. Search module is running. Check the configuration of the `/atg/search/service/SearchJTDataSource` component.

- If `SearchJTDataSource` is a `GenericReference`, find the value of its `componentPath` property by clicking `View Service Configuration`, and resolve that component.
- If `SearchJTDataSource` uses a `DirectJTDataSource`, find the `JNDIName` property of the `DirectJTDataSource`:

```
/atg/dynamo/service/jdbc/JTDataSource.properties:
dataSource=/atg/dynamo/service/jdbc/DirectJTDataSource
```




```
/atg/dynamo/service/jdbc/DirectoryDataSource.properties:
$class=atg.nucleus.JNDIReference
JNDIName=java:/ATGSolidsDS
```

Then make sure the corresponding application server's data source is configured properly. For example, on JBoss, you can find this in `<JBdi r>/server/server-name/deploy/atg-xxxx-ds.xml`:

```
<local-tx-datasource>
  <!-- This value matches JNDIName in DirectoryDataSource -->
  <jndi-name>ATGSolidsDS</jndi-name>
  <!-- Make sure this URL points to the correct db+schema -->
  <connection-url>jdbc:solids://localhost:1313</connection-url>
  <!-- The Login and password -->
  <user-name>admin</user-name>
  <password>admin</password>
</local-tx-datasource>
$JBOSS_DIR/server/<server-name>/deploy/atg-xxxx-ds.xml
<local-tx-datasource>
  <!-- This value matches JNDIName in DirectoryDataSource -->
  <jndi-name>ATGSolidsDS</jndi-name>
  <!-- Make sure this URL points to the correct db+schema -->
  <connection-url>jdbc:solids://localhost:1313</connection-url>
  <!-- The Login and password -->
  <user-name>admin</user-name>
  <password>admin</password>
</local-tx-datasource>
```

Check Lock Management Configuration

Misconfigured lock management can cause extensive problems for ATG applications. To check your lock management configuration:

1. On all ATG instances running the DAF Search module—which should include asset management, production, and staging if used—use ATG Dynamo Server Admin (see the [ATG Installation and Configuration Guide](#) for information on using this tool) to view the `/atg/search/routing/repository/SearchConfigurationRepository`.
2. Find the `LockManager` property, and click on the component value (for example, `/atg/dynamo/service/ClientLockmanager`).
3. If no lock manager has been assigned, assign one now.
4. In the lock manager component, ensure that `useLockServer` is set to `true`, and that the same values for `LockserverPort` and `LockServerAddress` are used on all ATG instances.
5. Ensure that a `ServerLockManager` is running on the host and port referred to by the `ClientLockManager` components.

See the *Locked Caching* section in the [ATG Repository Guide](#) for more information.



Check Event Server Connections

Event server problems are relatively rare, but if the other troubleshooting measures described in this section do not provide answers, you should look into your configuration. To verify your event server configuration:

1. For each ATG instance running the DAF Search module (asset management, production, and staging if used), use ATG Dynamo Server Admin to view the `/atg/search/routing/repository/SearchConfigurationRepository`. See the [ATG Installation and Configuration Guide](#) for information on accessing ATG Dynamo Server Admin.
2. Find the `eventServer` property and click on the component value, for example `/atg/dynamo/server/SQLRepositoryEventServer`.
3. Record the `LocalPort` and `hostAddr` or `hostName` properties of the server.
4. Ensure that the event server's `autoUpdateSubscribers` property is true.
5. Invoke the `checkSubscriptions` method
6. Select a single `/atg/dynamo/service/jdbc/SQLRepository` component on one ATG instance.
7. In the `Examine the Repository, Control Debugging` section of the page, select `List all Items` on the `das_gsa_subscriber` row.
8. Find all instances of `SearchConfigurationRepository` in the resulting list, and ensure that each host and port value you recorded previously are found in the address and port values.

Troubleshooting Search Results

This section provides suggestions for what to do when your search results do not match what you expect to see; for example, a search returns no results at all when results are expected, or does not include expected content items.

ATG Search offers the following tools for testing and investigating search results:

- **Dictionary Inspection**—Within Search Administration, you can use Dictionary Inspection to view terms in the ATG Search core dictionary and any custom dictionaries you have added, and experiment with processing options to see how they affect indexing. Search Text Analysis shows how a search is interpreted. See [Using Dictionary Inspection Tools](#) in this guide
- **Index Review**—Within Search Administration, you can use index Review to report on index metrics (how many items indexed, of what types), and to view individual content items. See [Using Index Review](#) in this guide
- **Merchandising Search Test**—From Merchandising, you can simulate a search performed on a commerce site without having to go to the site itself. More importantly, it allows you to choose search criteria to test, such as search configurations or facets, and the types of test output you want to see. The test output



shows detailed search processing information. See *Testing Search Queries* in the [ATG Merchandising Guide for Business Users](#).

- Query Console—ATG Search includes a lightweight sample application you can use for submitting test queries. The JSPs enable you to directly set search attributes that are rarely set by individual queries, in addition to those that would typically be determined by logic in the page or the form handler. See the *Sample Application* section of the [ATG Search Query Guide](#)

The sections that follow provide additional suggestions for troubleshooting your search results.

Search Returns No Results

If ATG Search returns no results at all when results are expected, consider the following checklist of possible causes:

- Was the index created and deployed without errors? Use Search Administration to view the indexing history.
- Is the correct index being searched? The correct partition? Check the `home/logs/searchEngineActivity/engineInfo.xml` activity log.
- Are the items in the index? If not, it might be a problem with your catalog or repository, or with the `IndexingOutputConfig` configuration. You can view the XHTML files and use index inspection tools (see [Using Index Review](#)) to see if the content was processed.
- Was the content indexed in the language being searched in? Check your Text Processing Option sets, selected language customizations. You can use index inspection tools (see [Using Index Review](#)) to highlight language usage, and the `request.xml` to pinpoint query options (see the [ATG Search Query Reference Guide](#) for details on the XML).
- Is the item text in question being searched? Check for errors in the query form handlers, and look at the `request.xml` file for `activeSolutilonZones` settings that may be excluding content items.
- Do the terms in the query match any item text? There are many possible causes for this problem, which can be diagnosed by viewing the `request.xml`, `response.xml` (`docMin=0` means no items found by any term), and Query Console:
 - boolean logic failure—Only one of the entered terms matches, when your settings require all terms to match. Check the `mode` attribute on the query.
 - term operators—Query operators might require a match on multiple terms, where only one term matches.
 - stop words—The user may be searching on a term that has been excluded from indexing (see the [Dictionaries](#) chapter of this guide for information). Use Dictionary Inspection to resolve this issue.
 - relevancy threshold—Check the `minScore` attribute in the `request.xml` file. The default is 0, which is fine for most circumstances; if it is set higher, you may be excluding results that should be returned.



- constraints—Constraints on the query may eliminate all results, or restrict the search to particular catalogs, categories, or facet values. See the request.xml file.
- Have values that limit the query been set too low? Check the request.xml file for the following possible problems (see the [ATG Search Query Reference Guide](#) for details on the XML):
 - pageSize=0
 - responseNumberSettings
 - requestSettings
- Search configurations (which are created in ATG Merchandising and also known as Merchandising Search Configurations) or query rules may exclude results; see the request and response XML files to view constraints and rules that have been used.

Search Returns Unexpected Results

If you find that searches return results, but the results are unexpected, consider the following checklist of potential problems:

- The items may have been found, but are ranked low in the results.
 - Use Content Inspection to make sure the item is included in the index.
 - Examine the response XML file for relevancy (score=N).
 - Check your search (rank) configurations.
- Items may be filtered out by mergeSettings. The mergeSettings attribute on the query can be used with multi-partition indexes to retrieve a different number of results than the per-partition requests. For example, if you have three partitions and are fetching 100 from each but want 200 total results, you need to set the responseNumberSettings for 100 and the mergeSettings for 200. Otherwise the mergeSettings will default to be the same as the responseNumberSettings, giving only 100.

Search Error Messages

This section lists specific error messages you may encounter while using ATG Search, and suggested remedies.

Connection Refused

If you are using ATG Search to index remote repositories, such as those used by an ATG Commerce installation, you may encounter the following error:

```
Connection refused to host: 127.0.0.1
```

If this happens, modify the application server startup script to include the argument:



```
-Djava.rmi.server.hostname=XXX.XXX.XXX.XXX
```

Replace the `XXX.XXX.XXX.XXX` value with an IP address which other ATG instances can communicate with on this machine.

No ManagedConnections Available Within Configured Blocking Timeout

If the `maxNumberProcessItemsThreads` specified in your `ProductCatalogOutputConfig` component is higher than the `max-pool-size` setting configured in your JBoss datasource, you may see an error similar to the following:

```
[ProductCatalogOutputConfig]
Starting bulk load 13:05:52,067 ERROR [ProductCatalog]
CAUGHT AT: CONTAINER: atg.repository.RepositoryException;
SOURCE: org.jboss.util.NestedSQLException:
No ManagedConnections available within configured blocking timeout
```

To resolve this issue, increase the `max-pool-size` setting in your JBoss datasource.

Could Not Prepare Environment

If you see the following exception when during the preparing content phase of indexing, it means that the ATG Search executable was unable to find the appropriate library dependencies in its search path.

```
14:35:06,189 WARN [IndexDeploymentService] Potential problem with
SearchEngine 300002 atg.search.routing.SearchEngineException:
atg.search.routing.SearchEngineLaunchException:
/ATG/ATG10.1/Search10/SearchEngine/i686-linux-gcc345/bin/atgsearch.exe
with error code 127
```

This can happen if the wrong version of ATG Search was installed for your operating system, or (on AIX) if the `LIBPATH` environment variable is missing or incorrect.

Verify that the executable is for the correct platform and (on AIX) that the search path `LD_LIBRARY_PATH` or `LIBPATH` is correctly set. To check the search path, run the `ldd` tool against the executable and ensure that all library dependencies can be found, and that the ones in the `bin` folder are selected. You can also check the executable and dependencies by running the following command at a command line:

```
atgsearch -v
```

Cannot Find Remote IndexingOutputConfig

If you see the following error, make sure the `IndexingOutputConfig` component is running on the remote server:

```
ERROR [RepositoryAdaptorService] Cannot find remote IndexingOutputConfig
at the following location:
rmi://host:port/atg/commerce/search/ProductCatalogOutputConfig
```



Locale Errors

If you see the following error, or any others including locale information, make sure you have appropriate language customizations configured.

```
ERROR [IndexLoader] Error occurred in step indexing_step on
contentatgrep: /CustomProductCatalog/product/xprod1030?catalog=
masterCatalog&locale=fr_FR for sync task 500008 of type incremental.
```

Not Enough Space

ATG Search requires a certain amount of virtual memory (also known as swap space) in order to carry out indexing tasks. If ATG Search does not have sufficient memory, you may see the following error:

```
java.io.IOException: Not enough space
```

You can estimate the required amount using the following formula:

```
<memory reserved by application server> + (<memory
reserved by search engine> * <number of search engines>)
```

The amount reserved by the search engine is controlled by the `MemoryReserveSize` setting in the `AEConfig.xml` file located in `<ATG10dir>/Search10/SearchEngine/platform/bin`.

For most ATG Search installations on JBoss, 2 GB minimum free is required. If your index is very large, you may require more.

Out of Memory Errors

If you have a catalog that includes millions of items and run out of memory during indexing, consider adjusting the `gcCallingRangeSize` property of the `/atg/search/repository/BulkLoader` component. This component controls how often Java garbage collection is called during indexing. The default interval is 10000 items. Decrease the setting to run garbage collection more frequently and reduce the risk of out of memory errors.

Note that this change will slow the indexing process somewhat.

This Server is Under Too Much Load for the Multiplexed...

In rare cases, due to a problem in the Java 6 JDK, if you are using multiplexed routing under extremely heavy load, ATG Search may throw a Null Pointer Exception. If this happens, your error logs will contain this message:

```
This server is under too much load for the multiplexed strategy on
Windows. This is a known problem with Sun's JDK (sun bug 5076772); reduce
load IMMEDIATELY or switch to searchStrategy=serialized.
```

Multiplexed and *serialized* are options used by search routing to optimize handling of multi-partition indexes. If using the multiplexed strategy (the default for a multi-partition index), routing addresses all



search engines simultaneously. If using the serialized strategy, routing contacts engines one after another.

To change your routing strategy to serialized, create a Configuration.properties file in `/atg/servername/localconf/atg/search/routing` with this line:

```
strategyType=serialized
```

Then rebuild and redeploy your ATG application EAR file.

ConfigurationClaimedException or IndexingException

The full error text most often encountered is:

```
ERROR [nucl eusNamespace. atg. search. reposi tory. I ncremental Loader]
atg. reposi tory. search. i ndexi ng. I ndexi ngExcepti on: Cannot complete
i ndexi ng update because we do not currently own the confi gurati on,
owned by "nul l"
```

Each ATG instance in which the DAF. Search. I ndex module is running performs a periodic check of the searchConfIg items in the I ncremental I temQueue repository, checking for searchConfIg items that are claimed, but for which the I astActi vi ty has not been updated in 10 minutes (by default). The error indicates that one ATG instance removed the lock held by the ATG instance doing the indexing.

The most common causes of this error are:

- Two ATG instances are running on the same machine with the same DRP port.
Check the `/atg/dynamo/ConfIgurat i on` component for each instance to make sure that the DRP ports are unique.
- Another machine clears out the lock that the indexing machine holds in the database table.
- Lock managers and related repository properties are not configured correctly, so other instances see stale data.

Make sure that lock managers are configured and shared correctly.

You should also check the other servers in your cluster for warning messages on the other servers in your cluster that begin with the words "Releasing configuration" and name the I ndexi ngOutputConfIg in question, and verify their DRP port and lock management configuration on those servers.

If none of the above problems is the cause, it may be that the application hung for more than ten minutes during indexing. To investigate this, collect and examine thread dumps from the indexing instance to identify where indexing is stuck. If indexing does not appear stuck, you may need to increase the timeout period configured in the `ti meOutMsecs` property of the `/atg/search/reposi tory/ConfIgStatePersi ster` component from the default ten minutes to a longer interval.

Caution: Increase this setting only if you are absolutely certain that no other problem is the cause.



Unable to Suggest Port for Engine

This error can appear in the search indexing error log if you do not have sufficient ports for the number of search engines running.

Every search indexing environment requires $2 * (\text{content set count})$ free ports for successful indexing. ATG Search automatically reserves 28 ports per host, but if you have a large number of content sets, that may not be sufficient.

If you encounter this error, ATG recommends increasing the number of hosts. You can also increase the available port range on an existing host:

1. On the Projects tab, go to Projects > *Your Project* > Environments.
2. Click the environment name.
3. Click the host machine name to which you want to add ports.
4. Click the Show link for Advanced Settings.
5. In the Port Numbers for Search Engines option, select Range and enter the new range to use.
6. Click Save.

Search Performance Tuning

By default, ATG Search is configured in what should be the optimal manner for an average site. If your site has unusual requirements, or you run into performance issues during testing, consider these possible configuration changes.

Configuring Multi-Threaded Indexing

Multi-threaded indexing in ATG Search is controlled by the `maxNumberProcessItemThreads` property of the `IndexingOutputConfig` component. This property is set to 5 by default. You can turn off multi-threaded indexing by setting `maxNumberProcessItemThreads` to 1.

Configuring the Number of Threads per Engine

If you have a single search engine per physical machine, the number of processing threads per engine is dynamically determined for optimum performance, and should not require adjustment. If for some reason the number of CPUs changes, restart the engine to recalculate thread allocation.

If you are running multiple engines per machine, you may want to adjust the ratio of threads to engines for best performance. For example, if CPU utilization is always 100% and response times are slow, there may be too many threads.

The total number of threads across all engines should equal the number of processors. You can set this value in the `maxSpareThreads` property of the `<ATG10dir>\Search10\SearchEngine\operating_system\bin\AESoapWaspConfig.xml` file



$$\text{maxSpareThreads} = \text{number_of_processors} / \text{number_of_engines}$$

Using a lower total number of threads may result in limited CPU usage; a higher number, in increased response times. If you have adjusted the default settings to run multiple engines per machine and the number of CPUs changes, adjust maxSpareThreads accordingly.

You must restart the engines in order for changes to take effect.

Adjusting TCP Settings for Heavy Load on Windows

The underlying ATG Search architecture that directs searches across multiple physical partitions uses TCP/IP ports and non-blocking NIO SocketChannels to connect to the search engines. These connections remain open in the TIME_WAIT state until the operating system times them out. Consequently, under heavy load conditions, the available ports on the machine running the Routing module can be exhausted.

On Windows platforms, the default timeout is 120 seconds, and the maximum number of ports is approximately 4,000, resulting in a maximum rate of 33 connections per second. If your index has four partitions, each search requires four ports, which provides a maximum query rate of 8.3 queries per second.

$$(\text{maximum ports}/\text{timeout period})/\text{number of partitions} = \text{maximum query rate}$$

If this rate is exceeded, you may see failures as the supply of TCP/IP ports is exhausted. Symptoms include drops in throughput and errors indicating failed network connections. You can diagnose this problem by observing the system while it is under load, using the `netstat` utility provided on most operating systems.

To avoid port exhaustion and support high connection rates, reduce the TIME_WAIT value and increase the port range.

Note: This problem does not usually appear on UNIX systems due to the higher default connection rate in those operating systems.

To set TcpTimedWaitDelay (TIME_WAIT):

1. Use the `regedit` command to access the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TCPIP\Parameters` registry subkey.
2. Create a new REG_DWORD value named `TcpTimedWaitDelay`.
3. Set the value to 60.
4. Stop and restart the system.

To set MaxUserPort (ephemeral port range):

1. Use the `regedit` command to access the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TCPIP\Parameters` registry subkey.
2. Create a new REG_DWORD value named `MaxUserPort`.

3. Set this value to 32768.
4. Stop and restart the system.

Disabling Index Compression

By default, search indexes are stored in the GZIP compression format. This compression allows you to retain more historical indexes, and provides for faster deployment of large indexes over the network. However, if your index is very small or is deployed locally, you may achieve faster performance without the compression.

To turn off compression, set `CompressDeploymentShare=0` in the `AEConfig.xml` file located in `<ATG10dir>/Search10/SearchEngine/platform/bin`.

After making any compression changes, you must do a full index of your content.

Adjusting Physical Partition Size

If you have reason to think that ATG Search is underestimating the size of your index, you can manually adjust the size of physical partitions.

For example, estimation says that your content requires three physical partitions, but you know that you are very close to needing four. To rule out any potential performance problems, you overrule the estimation and create four partitions instead. If this happens often, you may want to adjust the `fulnessOfRecommendedPartitions` setting in the `\atg\search\routing\Configuration.properties` file to some number less than one, as shown:

```
fulnessOfRecommendedPartitions=0.7
```

Going forward, ATG Search will estimate your content based on 70% of the maximum physical partition size, making it less likely that your partition will fill up.

To adjust the partition size, change the `MemoryReserveSize` property in the `AEConfig.xml` file. This setting controls how large a partition can be. For 32-bit Windows, the default setting is `0x54000000` (1.4 GB); 64-bit systems default to as much physical memory as possible.

Configuring Indexing for Large Content Sets

This section addresses indexing configuration for large individual content sets, such as those containing repositories or file sets involving millions of items.

You may want to configure ATG Search to perform *parallel indexing*. This configuration uses more than the normal resources to perform indexing tasks, which might otherwise take longer than is convenient.

Multiple content sets cannot be indexed in parallel. If you have a large individual content set, configure parallel indexing by changing the default `ActivePhysicalPartitions` property of the `\atg\search\routing\routingIndexService` component. This setting applies to all search projects.

ATG Search uses the number of search engines you specify to index content. These engines are distributed across the indexing environment's available host machines. Make sure that you have enough



host machines configured for the indexing environment; for example, if you configure defaultActivePhysicalPartitions=4, make sure that your machines have at least 4 free CPUs between them. ATG strongly recommends using dedicated machines for indexing; do not use machines for both indexing and answering.

Parallel indexing with multiple machines requires that all machines have access to the content to be indexed. Repository content is automatically streamed to the engines; however, if you are indexing file system content, the path to the content must be a shared location accessible to all of the engine machines.

Note that while parallel indexing can significantly improve performance, there is not a 1:1 ratio between the number of engines and processing speed, due to disk and network overhead, and returns diminish as more engines are added. Moving from one to two engines may cut indexing time in half, but adding another two engines may not lead to such a significant gain.

Updating Search Settings on a Live Site

It is preferable to make configuration changes to your search application in a development or staging environment and then deploy the changes to your production environment. However, under some circumstances it may be necessary to change settings in the AECConfig.xml file while your application is up and running.

In this case, you will need to shut down your search engines so that they can be restarted using the new settings. To make sure that users can continue to search your site while this update takes place, do the following:

1. On the machine on which your search application is running, go to `http://hostname:port/dyn/admin/nucleus/atg/search/routing/RoutingSystemService/`.
2. Select the environment with the engines you want to restart.
3. Click Cycle Engines.

When the page reloads, the engines have cycled. The page will display any errors that occur.

Warning: This procedure works **only** if you have more than one search engine per partition (which should always be the case to ensure good performance and continuous support for user queries). If you have only one engine per partition, your site will experience search interruptions.





Appendix A: Indexable File Types

ATG Search can support the following formats by converting them to HTML and indexing the resulting files. File formats are grouped by application type:

Word Processing

Text

Spreadsheet

Database

Raster Image

Vector Image

Presentation

Compressed Formats

Email Formats

Multimedia

Other

Word Processing

Product	Version
Adobe FrameMaker (MIF only)	3.0-6.0
Adobe Illustrator Postscript	Level 2
Ami	
Ami Pro for OS/2	
Ami Pro for Windows	2.0, 3.0
DEC DX	Through 4.0
DEC DX Plus	4.0, 4.1
Enable Word Processor	3.0-4.5



Product	Version
First Choice WP	1.0, 3.0
Framework WP	3.0
Hangul	97-2007
IBM DCA/FFT	
IBM DisplayWrite	2.0-5.0
IBM Writing Assistant	1.01
Ichitaro	5.0, 6.0, 8.0-13.0, 2004
JustWrite	Through 3.0
Kingsoft WPS Writer	2010
Legacy	1.1
Lotus Manuscript	Through 2.0
Lotus WordPro	9.7, 96-Millennium 9.6
Lotus WordPro (non Win32)	97-Millennium 9.6
MacWrite II	1.1
Mass 11	through 8.0
Microsoft Publisher (file ID only)	2003-2007
Microsoft Word for DOS	4.0-6.0
Microsoft Word for Macintosh	4.0-6.0, 98-2008
Microsoft Word for Windows	1.0-2007
Microsoft Wordpad	
Microsoft Works WP for DOS	2.0
Microsoft Works WP for Macintosh	2.0
Microsoft Works WP for Windows	3.0, 4.0
Microsoft Write for Windows	1.0-3.0
MultiMate	Through 4.0
MultiMate Advantage	2.0
Navy DIF	
Nota Bene	3.0



Product	Version
Novell PerfectWorks Word Processor	2.0
OfficeWriter	4.0-6.0
OpenOffice Writer	1.1-3.0
Oracle Open Office Writer	3.x
PC File Doc	5.0
PFS:Write	A, B
Professional Write for DOS	1.0, 2.0
Q&A Write	2.0, 3.0
Samna Word IV	1.0-3.0
Samna Word IV+	
Samsung JungUm Global (file ID only)	
Signature	1.0
SmartWare II WP	1.02
Sprint	1.0
StarOffice Writer	5.2-9.0
Total Word	1.2
Wang IWP	Through 2.6
WordMarc Composer	
WordMarc Composer+	
WordMarc Word Processor	
WordPerfect for DOS	4.2
WordPerfect for Macintosh	1.02-3.1
WordPerfect for Windows	5.1-X4
Wordstar 2000 for DOS	1.0-3.0
Wordstar for DOS	3.0-7.0
Wordstar for Windows	1.0
XyWrite	Through III+



Text

Product	Version
ANSI Text	7 and 8 bit
ASCII text	7 and 8 bit
EBCDIC	
HTML (CSS rendering not supported)	1.0-4.0
IBM DCA/RFT	
Macintosh Character Set	
Rich Text Format (RTF)	
Unicode	3.0, 4.0
UTF-8	
Wireless Markup Language	
XML (text only)	
XHTML (file ID only)	1.0

Spreadsheet

Product	Version
Enable Spreadsheet	3.0-4.5
First Choice SS	Through 3.0
Framework SS	3.0
IBM Lotus Symphony Spreadsheets	1.x
Kingsoft WPS Spreadsheets	2010
Lotus 1-2-3	Through Millennium 9.6
Lotus 1-2-3 Charts (DOS and Windows)	Through 5.0
Lotus 1-2-3 for OS/2	2.0
Microsoft Excel Charts	2.x-2007



Product	Version
Microsoft Excel for Macintosh	98-2008
Microsoft Excel for Windows	3.0-2007
Microsoft Excel for Windows (.xlsb)	2007 (binary)
Microsoft Works SS for DOS	2.0
Microsoft Works SS for Macintosh	2.0
Microsoft Works SS for Windows	3.0, 4.0
Multiplan	4.0
Novell PerfectWorks Spreadsheet	2.0
OpenOffice Calc	1.1-3.0
Oracle Open Office Calc	3.x
PFS: Plan	1.0
QuattroPro for DOS	Through 5.0
QuattroPro for Windows	Through X4
SmartWare Spreadsheet	
SmartWare II SS	1.02
StarOffice Calc	5.2-9.0
SuperCalc	5.0
Symphony	Through 2.0
VP-Planner	1.0

Database

Product	Version
DataEase	4.x
DBase	III, IV, V
First Choice DB	Through 3.0
Framework DB	3.0



Product	Version
Microsoft Access	1.0, 2.0
Microsoft Access Report Snapshot (File ID only)	2000-2003
Microsoft Works DB for DOS	1.0, 2.0
Microsoft Works DB for Macintosh	2.0
Microsoft Works DB for Windows	3.0, 4.0
Paradox for DOS	2.0-4.0
Paradox for Windows	1.0
Q&A Database	Through 2.0
R:Base	R:Base 5000; R:Base System V
Reflex	2.0
SmartWare II DB	1.02

Raster Image

Product	Version
CALS Raster (GP4)	Type I, II
Computer Graphics Metafile	ANSI, CALS, NIST
Encapsulated PostScript (header only)	TIFF
GEM Image (bitmap)	
GIF	
IBM Graphics Data Format (GDF)	1.0
IBM Picture Interchange Format	1.0
JBIG2 (graphic embeddings in PDF)	
JFIF (JPEG not in TIFF format)	
JPEG	
JPEG 2000	JP2



Product	Version
Kodak Flash Pix	
Kodak Photo CD	1.0
Lotus PIC	
Lotus Snapshot	
Macintosh PICT (BMP only)	
Macintosh PICT2 (BMP only)	
MacPaint	
Microsoft Windows Bitmap	
Microsoft Windows Cursor	
Microsoft Windows Icon	
OS/2 Bitmap	
OS/2 Warp Bitmap	
Paint Shop Pro (Win32 only)	5.0, 6.0
PC Paintbrush (PCX)	
PC Paintbrush DCX (multipage PCX)	
Portable Bitmap (PBM)	
Portable Graymap (PGM)	
Portable Network Graphics (PNG)	
Portable Pixmap (PPM)	
Progressive JPEG	
StarOffice Draw	6.x-9.0
Sun Raster	
TIFF	Group 5 and 6
TIFF CCITT	Group 3 and 4
TruVision TGA (Targa)	2.0
Word Perfect Graphics	1.0-10.0
WBMP wireless graphics format	
X-Windows Bitmap	x10 compatible



Product	Version
X-Windows Dump	x10 compatible
X-Windows Pixmap	x10 compatible

Vector Image

Product	Version
Adobe Illustrator	4.0-7.0, 9.0
Adobe Illustrator (XMP only)	11-13 (CS 1-3)
Adobe InDesign (XMP only)	3.0-5.0 (CS 1-3)
Adobe InDesign Interchange (XMP only)	
Adobe Photoshop (XMP only)	8.0-10.0 (CS 1-3)
Adobe PDF	1.0-1.7 (Acrobat 1-9)
Adobe PDF Package	1.7 (Acrobat 8-9)
Adobe Photoshop	4.0
Ami Draw	SDW
AutoCAD Drawing	2.5, 2.6, 9.0-14.0, 2000i-2007
AutoShade Rendering	2
Corel Draw	2.0-9.0
Corel Draw Clipart	5.0, 7.0
Enhanced Metafile (EMF)	
Escher graphics	
FrameMaker Graphics (FMV)	3.0-5.0
Gem File (Vector)	
Harvard Graphics Chart DOS	2.0-3.0
Harvard Graphics for Windows	
HP Graphics Language	2.0
IGES Drawing	5.1-5.3



Product	Version
Micrografx Designer	through 3.1; 6.0
Micrografx Draw	through 4.0
Microsoft XPS (text only)	
Novell PerfectWorks Draw	2
OpenOffice Draw	1.1-3.0
Oracle Open Office Draw	3.x
Visio (Page preview mode WMP/EMF)	4.0
Visio	5.0-2007
Windows Metafile	

Presentation

Product	Version
Corel Presentations	6.0-X3
Harvard Graphics Presentation DOS	3.0
IBM Lotus Symphony Presentations	1.x
Kingsoft WPS Presentation	2010
Lotus Freelance	1.0-Millennium 9.6
Lotus Freelance for OS/3	2
Lotus Freelance for Windows	95, 97
Microsoft PowerPoint for Macintosh	4.0-2008
Microsoft PowerPoint for Windows	3.0-2007
Microsoft PowerPoint for Windows Slideshow	2007
Microsoft PowerPoint for Windows Template	2007
Novell Presentations	3.0, 7.0
OpenOffice Impress	1.1, 3.0



Product	Version
Oracle Open Office Impress	3.x
StarOffice Impress	5.2-9.0
WordPerfect Presentations	5.1-X4

Compressed Formats

Product	Version
LZA Self Extracting Compress	
LZH Compress	
Microsoft Office Binder	95-97
Self-extracting .exe	
UNIX Compress	
UNIX GZip	
UNIX tar	
Uuencode	
Zip	PKZip, WinZip

Email Formats

Product	Version
Encoded mail messages	MHT; Multi Part Alternative; Multi Part Digest; Multi Part Mixed; Multi Part News Group; Multi Part Signed; TNEF
IBM Lotus Notes Domino XML Language DXL	8.5
IBM Lotus Notes NSF (Windows only with Notes Client or Domino Server)	8.x



Product	Version
Microsoft Outlook (MSG)	97-2007
Microsoft Outlook Express (EML)	
Microsoft Outlook Forms Template (OFT)	97-2007
Microsoft Outlook OST	97-2007
Microsoft Outlook PST	97-2007
Microsoft Outlook PST (Mac)	2001

Multimedia

Product	Version
AVI (metadata extraction only)	
Flash (text extraction only)	6.x, 7.x, Lite
Real Media (file ID only)	
MP3 (ID3 metadata only)	
MPEG-1 Audio layer 3 ID3 v1 (file ID only)	
MPEG-1 Audio layer 3 ID3 v2 (file ID only)	
MPEG-1 Video V 2 (file ID only)	
MPEG-2 Audio (file ID only)	
MPEG-4 (metadata extraction only)	
MPEG-7 (metadata extraction only)	
QuickTime (metadata extraction only)	
Windows Media AST (metadata extraction only)	
Windows Media DVR-MS (metadata extraction only)	
Windows Media Audio WMA (metadata extraction only)	
Windows Media Playlist (file ID only)	
Windows Media Video (metadata extraction only)	
WAV (metadata extraction only)	



Other

Product	Version
Microsoft InfoPath (file ID only)	2007
Microsoft OneNote (file ID only)	2007
Microsoft Project (text only)	98-2003
Microsoft Windows DLL	
Microsoft Windows Executable	
vCalendar	2.1
vCard	2.1
Yahoo! Messenger	6.x-8



Appendix B: Using WGet for Web Site Indexing

ATG recommends using the free WGet utility to retrieve Web pages for indexing. To use:

1. Download WGet from

```
http://www.christopherlewis.com/WGet/wget_SVN.zip
```

2. Unzip the file. See the wget.help file for information on using the utility.
3. Run WGet against the Web site you want to retrieve. For example:

```
wget -P atg -r -l 2 http://www.mycorp.com
```

This example downloads the URL to an /atg directory, and specifies two levels of recursion; the maximum supported is five levels.

4. Add the directory that WGet created (in this example, the atg directory) as file system content for your project. Be sure to specify the URL of the Web site you are retrieving as the external access URL in the content Advanced Settings. In this example, you would specify <http://www.mycorp.com>.
5. Index the project.





Appendix C: Loading Customizations by Reference

Customizations are used to enhance the out-of-the-box configuration for features such as languages, dictionaries, and topic sets. Most customizations are created and edited using the Workbench. However, customization types can also be loaded *by reference*.

To add customizations by reference, create an XML file of the customization data and place that file in the correct folder in your ATG installation. Search Administration automatically picks up the XML files and makes the customizations available to add to your search projects.

Note: By default, search customization data uses UTF-8 encoding. If your customization data file is encoded in something other than UTF-8, specify the correct encoding in the XML file header. For example:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

Auxiliary Data by Reference

Auxiliary data in ATG Search provides a way to add or change property values on content items that have already been indexed. Reasons you may want to use this feature include:

- The property values for your content items come from a source external to the catalog or repository you are indexing. In this case, you'll have a separate process that prepares the auxiliary data, and use the process described in this appendix to load the data into the index. (An alternative approach is to use custom property accessors in `IndexingOutputConfig` to get these properties into the originally indexed XHTML; see [Customizing the XHTML Output](#)).
- Content items need to have certain properties updated on a more frequent basis than others (for example: inventory status, sale price, "hot" item, "hot" solution, etc.). These property values can change rapidly, and you do not want to fully reindex all of your content just to update them.

To use auxiliary data, you first need to create an XML file, either manually or using your preferred automation method, that uses the following format:

```
<queryReport>  
<propertyReport>
```



```
<property name="prop_name" value="value" type="type">http://docurl</property>
</propertyReport>
</queryReport>
```

Each property tag corresponds to a property value to be added or updated. Attributes on the tag indicate the name, value, and type of the property. The contained text is the URL of the content item that should have the specified property value assigned, and should be the URL of the content item as it is known to ATG Search.

For example:

```
<queryReport>
  <propertyReport>
    <property name="hotScore" value="3885"
type="integer">atgrep: /ServiceRepository/Solution/11</property>

    <property name="hotScore" value="2884"
type="integer">atgrep: /ServiceRepository/Solution/111</property>

    <property name="hotScore" value="1883"
type="integer">atgrep: /ServiceRepository/Solution/121</property>

  </propertyReport>
</queryReport>
```

The text of the tag can also be of the form `property==value`, in which case every content item meeting that property constraint will have the indicated property value added or updated to it.

For example:

```
$solution_id==primus18537
```

Place the XML file in your `<ATG10dir>/home/localconfig/atg/searchadmin/customizations/auxiliary_data` directory.

If you are making these changes as a routine part of your content updates and site maintenance, consider a scripting solution to both generate the XML file and place it in the correct directory.

You can then use Search Administration to select the auxiliary data customization item on the project to which you want to apply the data. You can set up an indexing job using the Load Post-Index Customizations Only option (see the [Building Indexes](#) section of this guide) to apply your auxiliary data to your content without doing a full reindex.

Note: The naming convention for properties used in this XML file format does **not** include the `atg: type` prefix information used by the structured XHTML format (see the [ATG Search Query Guide](#)). If you are loading a repository item XHTML that refers to property `atg: integer: $repositoryId`, then in this file the name to use is `$repositoryId`.



Merchandising Search Configurations by Reference

Search configuration data is created using ATG Merchandising; you should not attempt to create these documents manually.

By default, the file customization adapters look in the following directories in your <ATG10dir>/home/localconfig/atg/searchadmin/customizations directory to find by reference file customization items:

`/rank_configs`





Appendix D: Implementing Type-Ahead in ATG Search

ATG Search supports integration with a type-ahead feature that attempts to auto-complete user-entered text on your Web site. This appendix is concerned with how to set up type-ahead in Search Administration; in addition, you must configure your JSP pages to support this feature before your customers can use it (see the [ATG Search Query Guide](#) for information).

ATG recommends that you dedicate a server for answering type-ahead requests, to reduce load on the search engines. Otherwise, you may see performance degradation. If type-ahead performance is slow, you may need to add hardware resources to the search project.

This appendix includes the following sections:

[Creating a Type-Ahead Value List](#)

[Making Type-Ahead Available to Your Site](#)

Creating a Type-Ahead Value List

Type-ahead in ATG Search works by referring to a pre-generated list of possible values that can be used to complete the user's input. For example, you might want to use a list of products, frequently asked questions, or most-commonly-searched strings. The list can be generated using database tools or can be populated manually. In addition to the text values, the file contains ranking information, which determines the order in which values are returned. The default is to return values in descending order by rank.

For example, your type-ahead file could contain the following product names and associated ranks:

bol t (small)	1
bol t (large)	2
bol t cutter	3
bottle opener	4
grommet	5

If the user types "bol" in the search field on your site, a dropdown list appears, populated with the following selections, in rank order:



```
bol t cutter
bol t (large)
bol t (small)
```

This information must be provided in the form of an XML file. The file can have any name, and is formatted as a simple key-value pair:

```
<typeAhead lang="/ language" min="3" max="10" top="5">
  <t v="1">bol t (small)</t>
  <t v="2">bol t (large)</t>
  ...
</typeAhead>
```

Replace */ language* with the name of the language you are using, such as English. The `t` element contains the text to use for the type-ahead, ordered by the ranking indicated in the `v` attribute.

The attributes of the `typeAhead` element are:

- `lang`—The language of the data, which should match the query language to use for searching. The LANGUAGE must be a loaded language that is loaded in the Search Administration project (see the next section). If you want to provide type-ahead in more than one language, you must generate separate files for each language.
- `min`—The minimum size of a key to store, and therefore the minimum entered characters before the UI should submit a request.
- `max`—The maximum size of a key to store. Requesting results with a longer key will simply truncate the key.
- `top`—The maximum text results to store under each key.

Remember that by default, XML uses UTF-8 encoding. If your customization data file includes non-UTF-8 characters, specify the correct encoding in the XML file header:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

If your XML file is extremely large, you can break it up into as many separate files as you want to make them more manageable. If you do have multiple XML files, attributes in the first file loaded are used for the entire set.

Once you have the list of ranked values, you can make it available to your site.

Making Type-Ahead Available to Your Site

To make your type-ahead text available to end users:

1. Place your type-ahead text XML file or files in the following directory of your Search Administration server:



<ATG10dir>/home/localconfig/atg/searchadmin/customizations/auxiliary_data

2. In Search Administration, create a new project. In the project's General settings, select No Content Items in This Search Project.
3. After creating the project, go to the project's Content page and select **Auxiliary Data** from the Post-Index Customization Add Item list.
4. Check the type-ahead search customization you want to associate with the project.
5. Click **Save** to save your selections and return to the Content page.
6. Build the index.
7. If you expect to update the type-ahead values regularly, set up automatic indexing rules. Note that you cannot use incremental indexing for customization data, but must use the Full Index option.

Your type-ahead values are now available for retrieval by your customer-facing site. You must implement code in your JSPs to issue the appropriate requests to retrieve those values when customers begin typing in a search field.





Glossary

Term	Definition	Replaces prior term
content	Raw data for indexing, such as html files, Word documents, or ATG repositories.	
content label	The content label identifies the purpose of a content set, such as a catalog or a collection of articles. If your search project includes different types of content that you want to search individually, you can use content labels to differentiate the content types.	
content set	Bundles content into a single organizational unit that can be indexed as one.	document repositories/sets
customizations	Blanket term for topic sets, query rules, dictionaries, facet sets, languages, parsing options, query rules, and preferred answers.	
deploying	The action of moving indexes and/or customization data from a staging to a production environment	
dictionary	Contains definitions of special words and the relationships between them. Users can add terms to improve the search responses.	
expansion	A synonym for a term, either defined in a custom dictionary or in the default language files	
facet set	Allows users to search within an existing result set.	
index	The end result of indexing content, and what is used to answer queries submitted by end-users.	IdeaMap
indexing environment	Search environment that indexes content.	indexing server



partition	An index is divided into one or more Logical Partitions, depending on the number of Content Sets. Each Logical Partition is composed of one or more Physical Partitions, depending on the size of the content.	
learning files	Rules that automatically apply topics to a content collection.	
preferred answer	Custom response to frequently-asked end-user questions. Can be linked to source text within content collections.	
project	Projects represent the top-level container for managing and indexing content. The project associates customization data and search environments with an index.	Job script (also included actions, which are now part of synchronization)
query rules	Link actions in ATG Search to certain types of user queries.	
Search Administration	The ATG Search user interface, including both project administration and the Workbench.	Management Console
search engine	Part of ATG Search that creates the index, receives an end-user query and returns the results of the search.	Answer Server
search environment	A combination of a single index and one or more machines and search engines used for indexing or searching.	Content clusters, also called clusters
stem	In linguistics, the stem is the part of a word that is common to all its inflected variants. For example, the stem of <i>banking</i> , <i>banker</i> , and <i>banks</i> is <i>bank</i> .	
structured data	Content that adheres to a well-defined structure; in ATG Search, typically repository data that has been exported to XHTML.	
target type	Identifies the general purpose of the search project. If you are using ATG Content Administration, the target types should correspond to your Content Administration targets.	
topic set	A hierarchy of topics.	topic taxonomy
term	Individual entry within a dictionary.	



term weight set/stop words	A set of term/weight pairs, in which the weight describes the amount of influence a term has in a search.	
topics	User-defined classification system that can be applied to indexes, allowing end-users to refine their search by finding related information.	
type-ahead	ATG Search feature that allows you to provide a list of terms that can be used in your UI to auto-complete search terms as they are entered.	
unstructured data	Content that has no governing structure, such as that typically found in file systems, which might include a variety of document formats and information types.	
Workbench	Part of the user interface where users create and edit customization data.	Management Console (partial)



Index

A

- acronyms, in dictionaries, 120, 121
- actions
 - deleting from rules, 164
 - syntax, 153
- Active Search Project, 13
- Add Content Set page, 22, 27
- Advanced Settings
 - for file system content, 25
 - in environments, 86
- ATG server, 1
- ATG10dir, 1
- audience, 1
- auxiliary data, 225
 - adding by reference, 225

B

- Blank Extension, 25
- bulk loading, 59
 - configuring, 60
- Business Control Center, 9

C

- child terms
 - adding to dictionaries, 120
- ConcatFilter, 70
- configuring
 - bulk loading, 60
- content, 8
 - adding customizations to, 28
 - adding to search projects, 20
 - automatically associating with topics, 98
 - removing customizations to, 28
 - viewing in search projects, 28
- Content Administration
 - triggering indexing from, 39
- Content Item Type, 23, 27
- content items, 3
- content sets, 7
 - adding file system content to, 22
 - adding repository content to, 27
 - creating, 21
 - deleting, 22

- Context object, 67
- conventions, 1
- Converted Document Access URL, 26
- Converted Document Output Directory, 26
- CustomCatalogPropertyAccessor, 75
- CustomCatalogVariantProducer, 75
- customization data, 8
 - post-index, 38
 - types, 8
- customizations
 - adding by reference, 225
 - adding to content, 28
 - applying to indexes, 35
 - removing from content, 28
- customizing XHTML output, 65

D

- data loading, 59
- default property values, 56
- definition file format, 50
 - locale attribute, 59
 - prefix element, 58
 - suffix element, 58
- Deploy indexing option, 38
- dictionaries, 105
 - acronyms as terms, 120, 121
 - adding, 117
 - adding child terms to, 120
 - adding terms to, 119
 - deleting, 119
 - duplicating, 118
 - editing, 118
 - importing, 119
 - load modes, 117
 - terms, 119
- dictionary adapters, 29
- dictionary inspection, 105
- dictionary terms. *See* terms
- docsets. *See* document sets
- document set, 100
- Document Set Name, 23, 27
- document sets, 3, 21
- document submitters
 - configuring, 61
- documents
 - URLs for, 26

**E**

- environments, 8
 - adding to search projects, 88
 - creating, 83
 - deleting, 91
 - moving, 90
 - viewing, 89
- extensions
 - mapping, 24
- External Access URL, 26

F

- facet sets, 139
 - adding child facets, 140
 - creating, 140
 - deleting, 143
 - duplicating, 145
 - exporting, 144
 - importing, 144
 - modifying, 143
- facets
 - adding to facet sets, 140
 - deleting, 143
 - modifying, 143
 - moving, 143
- features, 3
- fielded search, 21
- file system content
 - adding to content sets, 22
- File Type Mapping, 24
- FirstWithLocalePropertyAccessor, 66

G

- General File Suppression, 25

H

- History tab, 46
- host machine, specifying, 84
- HtmlFilter, 72

I

- image files
 - accessing, 26
 - storing, 26
- incremental loading, 59
 - monitored properties, 61
 - tuning, 61
- index, 3, 7, 15, 35
 - building, 36
 - customizing, 8
 - deleting histories, 46
 - deploying, 38
 - estimating final size, 33
 - Full indexing option, 37
 - Incremental indexing option, 37

- options, 36
 - saving options as rules, 39
 - triggering from Content Administration, 39
 - viewing histories, 46

- index history
 - deleting, 46
 - viewing, 46
- indexing
 - multi-threaded, 206
- indexing rules, creating, 39
- IndexingOutputConfig, 60
- IndexingOutputConfig Path, 27
- item subtypes
 - indexing, 55

L

- languages
 - selecting for synonyms, 124
 - supporting multiple, 29
- large content items, 208
- learning files. See auxiliary data
- Load Post-Index Customization Data indexing option, 38
- loading data, 59
- LocaleVariantProducer, 67
- logging
 - data for search reports, 167

M

- macros
 - adding global, 102
 - adding local, 103
 - deleting, 103
 - in query rules, 161
 - in rule patterns, 100
 - using in rule patterns, 102
- Map properties
 - indexing, 54
- maxNumberProcessItemThreads, 206
- Merchandising Search Configuration, 225
- metadata, 3
- metadata properties
 - indexing, 77
- metatags
 - in rule patterns, 100
 - using for search refinement, 139
- monitored properties, 61
- multisite catalogs
 - indexing, 79
- multi-threaded indexing, 206
- multi-value properties
 - indexing, 53
 - XHTML output, 52

N

- non-repository properties
 - indexing, 56



normalizing property values, 58

P

parallel indexing, 208

patterns

- deleting from rules, 164
- syntax, 151

percent quoted string, 100

port numbers

- for environments, 86

preferred answers, 160

price lists

- indexing data in, 80

PriceListMapPropertyAccessor, 81

PriceListPropertyProvider, 80

ProductCatalogOutputConfig, 49, 73

product-catalog-output-config.xml file, 73

production environments, 8

Project Administration, 7

property accessors, 65

- CustomCatalogPropertyAccessor, 75
- FirstWithLocalePropertyAccessor, 66

property values

- default for indexing, 56
- normalizing, 58
- translating, 58

PropertyFormatter, 68

PropertyValuesFilter, 69

Q

query rule sets

- creating, 147
- deleting, 163

query rules, 147

- enabling and disabling, 161
- exporting, 163
- importing, 162

quoted string, 100

R

refineConfig.xml, 139

renaming index properties, 57

reporting

- data logging, 167

repository content

- adding to content sets, 27

repository indexing, 49

- ConcatFilter, 70
- Context object, 67
- customizing output, 65
- default property values, 56
- definition file format, 50
- determining the properties to index, 76
- excluding uncategorized products, 78
- HtmlFilter, 72
- item subtypes, 55
- loading data, 59
- Map properties, 54

metadata properties, 77

multisite catalogs, 79

multi-value properties, 53

non-repository properties, 56

property accessors, 65

PropertyFormatter, 68

PropertyValuesFilter, 69

renaming output properties, 57

suppressing properties, 57

text properties, 77

transforming data, 49

translating property values, 58

UniqueFilter, 70

UniqueWordFilter, 71

variant producers, 67

requests, 3

responses, 3

rollback policy, setting, 85

root directory, 23

rule groups

- adding to query rule sets, 148
- deleting, 164
- moving, 163

rule patterns

- adding to topics, 98
- deleting rules, 102
- enabling and disabling, 102
- examples, 101
- syntax, 100

rule sets

- adding rule groups to, 148
- adding rules to, 149
- moving, 163
- rule syntax, 151

rules

- adding to query groups, 149
- deleting, 164

S

Search Administration, 7

- accessing, 12
- search projects in, 15

search environment. See environment

search projects, 7, 15

- adding content to, 20
- creating, 17
- deleting, 20
- duplicating, 18
- indexing, 33
- viewing, 19



- Search Query request, 3
- search results, refining. See facet sets
- Search Workbench. See Workbench
- SearchAdmin role, 9
- searching, definition, 3
- security, 9
- site-specific content, 31
- SKU-based indexing, 73
- stem, 100
- structured content, 21, 23, 27
- subtypes
 - indexing, 55
- suppressing properties from indexes, 57
- Suppression, 25
- synonyms
 - adding to terms, 122
 - deleting, 124
 - modifying, 124
 - selecting language for, 124
 - types, 123

T

- term dictionaries. See dictionaries
- term weight sets. See weight sets
- terms
 - adding synonyms to, 122
 - adding to dictionaries, 119
 - adding to weight sets, 127
 - deleting, 122
 - deleting from weight sets, 127
 - modifying, 121
 - moving, 122
 - overview, 119
- text processing option sets, 129
 - adding content-level, 131
 - adding project-level, 130
 - deleting, 136
 - duplicating, 135
 - exporting, 136
 - importing content-level, 132
 - importing project-level, 131
 - modifying, 136
- text properties
 - indexing, 77
- topic sets, 93
 - adding to search projects, 95
 - adding topics to, 95
 - creating, 94
 - deleting, 97
 - exporting, 95, 118
 - importing, 94
 - modifying, 95
 - renaming, 97
- topics
 - adding rule patterns to, 98
 - adding to topic sets, 95
 - assigning weights, 99

- automatically associating with content, 98
- deleting, 97
- moving, 96
- renaming, 97
- rule pattern examples, 101
- rule pattern syntax, 100
- sorting, 96

- TPO sets. See text processing option sets
- transforming catalog data
 - ProductCatalogOutputConfig, 49, 73
- transforming repository data, 49
- translating property values, 58

U

- uncategorized products
 - excluding from indexing, 78
- UniqueFilter, 70
- UniqueSiteVariantProducer, 79
- UniqueWordFilter, 71
- Unspecified Extensions, 25
- unstructured content, 23, 27
- URLs
 - in rule patterns, 100

V

- variant producers, 67
 - CustomCatalogVariantProducer, 75
 - LocaleVariantProducer, 67
 - UniqueSiteVariantProducer, 79
- View Item request, 3

W

- weight sets, 126
 - adding, 126
 - adding terms to, 127
 - copying, 126
 - default, 126
 - deleting, 128
 - deleting terms from, 127
 - importing, 127
- weights
 - assigning to topics, 99
- wildcard, 100
- Workbench, 8
 - accessing, 12

X

- XHTML output
 - customizing, 65
 - directing to file, 61
 - document format, 51
 - viewing in Component Browser, 63