

Content Categorizer Administration Guide
10g Release 3 (10.1.3.3.0)

March 2007

Content Categorizer Administration Guide, 10g Release 3 (10.1.3.3.0)
Copyright © 2007, Oracle. All rights reserved.

Contributing Authors: Deanna Burke

Contributors: Evan Suits

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

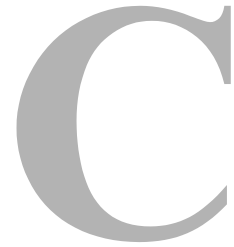
U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Table of Contents



Chapter 1: Introduction

Overview	1-1
Introduction to Content Categorizer	1-2
Search Rules	1-2
Search Rule Override	1-2
XML Conversion	1-3
Flexiondoc XML Converter	1-3
SearchML XML Converter	1-4
Operating Requirements	1-4
Operating Modes	1-5
Interactive Mode Process	1-5
Batch Mode: Process	1-6
Supported Platforms	1-8

Chapter 2: Setting Up Content Categorizer

Overview	2-1
Setting XML Conversion Method	2-1
Defining Field Properties (Optional)	2-2

Chapter 3: Using the CC Admin Applet

Overview	3-1
Introduction to CC Admin Applet	3-2
Log Levels	3-6
Setting Log Levels	3-7
Key Field	3-11
Count Field	3-13
Needs Rebuild? and Is Orphan? Status	3-27

Chapter 4: Using Batch Categorizer

Overview	4-1
----------------	-----

Chapter 5: Search Rules

Overview	5-1
Understanding Search Rules	5-1
Search Rule Types	5-2
Search Rule Guidelines	5-2
Pattern Matching Search Rules	5-3
Rule Types	5-3
Sub-Types	5-3
Key	5-4
Count	5-4
Examples	5-4
TAG_TEXT	5-4
TAG_ALLTEXT	5-5
TEXT_REMAINDER	5-5
TEXT_ALLREMAINDER	5-6
TEXT_FULL	5-6
TEXT_ALLFULL	5-6
TEXT_NEXT	5-7
TEXT_ALLNEXT	5-7
Abstract Search Rules	5-8
Rule Types	5-8
Key	5-8
Count	5-9
First Paragraph	5-9
First Sentence	5-9
Examples	5-9
FIRST_PARAGRAPH	5-9
FIRST_SENTENCE	5-10
Option List Search Rule	5-11
Rule Types	5-11
Key	5-12
Count	5-12
Examples	5-12
Categorization Engine Search Rule	5-15
Rule Types	5-15
Key	5-15
Count	5-15

Filetype Search Rule	5-16
Rule Types.....	5-16
Key.....	5-16
Count.....	5-17
Examples.....	5-17
Defining Search Rules	5-17
Defining Search Rules.....	5-18
Defining Option List Keywords.....	5-19
Applying Rules to the Type Field.....	5-20

Chapter 6: Using Categorizer Engines

Overview	6-1
Categorizer Engine Registration	6-1
Building Query Trees	6-2
Hierarchical Browsing	6-4
Browsing the Taxonomy	6-4
Browsing a Category in the Taxonomy	6-5
Browsing a Sub-Category	6-6

Appendix A: Adaptor Modules

Adaptor Modules for Autonomy and APR Smartlogik	A-1
---	-----

Appendix B: Sample *doc_config.htm* page

Appendix C: Configuration Variables

Overview	C-1
MaxQueryRows	C-1

Appendix D: XSLT Transformation

Overview	D-1
Translation Using OutsideIn XML Export Filters	D-2
Transformation Using XSLT Stylesheets.....	D-2
SearchML Transformation	D-3
Document Properties and Text Style Examples	D-3
Flexiondoc Transformation	D-4
Document Properties Example	D-4
Text Style Example	D-5
Example Files	D-8

Appendix E: Third Party Licenses

Overview E-1

Apache Software License E-1

W3C® Software Notice and License E-2

Zlib License E-4

General BSD License..... E-5

General MIT License E-5

Unicode License..... E-6

Miscellaneous Attributions E-7

INTRODUCTION

OVERVIEW

Content Categorizer (CC) suggests metadata values for documents being checked into Content Server, and can be used to recategorize the metadata of documents that are already in Content Server. The metadata values are determined according to search rules provided by the system administrator. Third-party categorization engines can be integrated with CC to categorize documents based on taxonomies defined through the engine.

Content Categorizer includes a Batch utility that can search a large number of files and create a Batch Loader control file containing appropriate metadata field values. The Batch utility can be used to recategorize existing content (already checked into the content server repository).

This section includes the following topics:

- ❖ [Introduction to Content Categorizer](#) (page 1-2)
- ❖ [Operating Modes](#) (page 1-5)
- ❖ [Supported Platforms](#) (page 1-8)

INTRODUCTION TO CONTENT CATEGORIZER

Content Categorizer (CC) suggests metadata values for new documents being checked into Content Server, and for existing documents into Content Server that may or may not already have metadata values. These metadata values are determined according to search rules provided by the System Administrator.

Search Rules

Content Categorizer executes its search rules depending on the type of rule defined:

- ❖ **Pattern Matching and Abstract Rules:** Content Categorizer scans a content document looking for “landmarks”. A landmark can be specific text, or it can be based on structural properties of the source document, such as styles, fonts, and formatting.
- ❖ **Option List Rule:** Content Categorizer searches for keywords whose cumulative score determines which option of an option list is selected. It does not look for either landmarks or specific XML tags.
- ❖ **Categorization Engine Rule:** Content Categorizer invokes a 3rd-party categorizer engine and taxonomy to categorize a content item.
- ❖ **Filetype Rule:** Content Categorizer looks for the document file type (the filename extension).

Search Rule Override

Normally, a user-entered value on the Content Check In Form prevents Content Categorizer from applying the search rules for that field. This is also true for option list fields that have a default value, such as the Type field.



Important: It is important to instruct contributors to leave blank any fields they want to have filled by search rules.

- ❖ The current version of Content Server automatically inserts a blank value as the default value in a custom option list field. In this case, the first value (by default, a blank value) will not be considered a user-entered value, and the Option List search rule will be applied. If you do not want the Option List search rule to override the first value in a custom option list field, you must provide a default value for that option list on the Configuration Manager Applet.

- ❖ To have Content Categorizer ignore the default value and apply the search rules to the Type field, you can edit the Content Server configuration file. See [Applying Rules to the Type Field](#) (page 5-20).

XML Conversion

For Content Categorizer to recognize structural properties, the content must be converted to XML (eXtensible Markup Language). The conversion method is a user-defined runtime configuration setting. The variable name is `sccXMLConversion` and its possible values include `Flexiondoc`, `SearchML` and `None`. The *None* value is used for files that are already XML.



Note: The `CC_Sample` directory that was installed with Content Categorizer includes a sample source file named *Wellington_WordStyle.doc* that is “artificially rich” with document properties and styles. The directory also contains sample XML files (*Wellington_WordStyle_flexion.xml* and *Wellington_WordStyle_searchml.xml*) that demonstrate the XML that results when the source document is converted by each of the available XML converters.



Important: There is a problem with the XSLT transformation used to post-process PDF content converted using the Flexiondoc schema. When Flexiondoc schema are used, single words are assigned to individual XML elements, making the final XML unusable. It is therefore necessary to use SearchML for categorizing PDF content.

Regardless of which XML converter is specified, the XML intermediate files are used only by Content Categorizer, so they are discarded after use, and documents are checked into Content Server in their original source form. The only exception is content that is already in XML format, which is not subjected to the translation process.

Flexiondoc XML Converter

The OutsideIn XML Export technology is used in combination with a custom XSLT style sheet (*flexiondoc_to_scc.xsl*) to produce XML in a two-stage process. In the first stage, the native document is converted to Flexiondoc-formatted XML. In the second stage, the style sheet is used to further refine the XML so that it is searchable by Content Categorizer. Native document properties and text segments are isolated in XML elements, which are named after the corresponding document property, paragraph style, or character style.



Note: Refer to [Supported Platforms](#) (page 1-8) for information about the current Content Server platforms on which SearchML and Flexiondoc are supported.

SearchML XML Converter

The OutsideIn technology is used in combination with a custom XSLT style sheet (*searchml_to_scc.xsl*) to produce XML in a two-stage process. In the first stage, the native document is converted to SearchML-formatted XML. In the second stage, the style sheet is used to further refine the XML so that it is searchable by Content Categorizer. Native document properties and text segments are isolated in XML elements, which are named after the corresponding document property or paragraph style. Character styles are not supported by SearchML.



Note: Refer to [Supported Platforms](#) (page 1-8) for information about the current Content Server platforms on which SearchML and Flexiondoc are supported.

Operating Requirements

To run Content Categorizer, these settings are required:

❖ Define the XML Conversion method in Content Categorizer as one of these:

- `sccXMLConversion=Flexiondoc`
- `sccXMLConversion=SearchML`

See [Setting XML Conversion Method](#) (page 2-1).

❖ Define search rules in the Content Categorizer Admin applet. See [Defining Search Rules](#) (page 5-18).

❖ **Optional:** Define field properties, including default values for the metadata fields in the Content Categorizer Admin Applet. See [Defining Field Properties \(Optional\)](#) (page 2-2) for more information.



Important: To use the CATEGORY search rule, you must install, set up and register a categorizer engine before you can define the CATEGORY rule for any metadata fields.

OPERATING MODES

Content Categorizer can operate in either Interactive mode or Batch mode. All modes require conversion of the source documents into XML intermediate form. However, the process flows of the modes are distinctly different.

- ❖ **Interactive** mode integrates Content Categorizer with the Content Check In Form and Info Update Form in Content Server. Users click the **Categorize** button on the form to run Content Categorizer on a single content item. Any value that is returned by Content Categorizer is a “suggested” value, because the contributor can edit or replace the returned value.
- ❖ **Batch** mode is used when recategorizing large numbers of documents that are already in the content server repository. The system administrator uses a stand-alone Batch Categorizer utility to run Content Categorizer, and then either performs a “live update” of content metadata or uses the output file from Batch Categorizer as input to the Batch Loader.

Interactive Mode Process

The following steps occur during the checkin process:

1. A contributor displays the Content Check In Form or the Info Update Form, selects a primary file (only on Content Check In Form), and clicks the **Categorize** button.
2. The Content Check In Form copies the primary file to the Content Server host and calls the Content Categorizer service.
3. Content Categorizer locates the source content.
4. If the content is already in XML format, no translation occurs, and the process continues at step 6.
5. If the content is not already in XML format, it will be converted using the specified conversion method:

Flexiondoc

- a. The content is converted into Flexiondoc-formatted XML.
- b. The XML is translated into “Content Categorizer-friendly” XML, using *flexiondoc_to_scc.xsl*.

SearchML

- a. The content is converted into SearchML-formatted XML.
- b. The XML is translated into “Content Categorizer-friendly” XML, using *searchml_to_scc.xml*.
6. Content Categorizer applies the search rules to the XML and obtains suggested values for the specified metadata fields.
7. Content Categorizer inserts the suggested metadata values into the Content Check In Form or Update Info Form, and returns the form to the contributor.
8. The contributor can check in or submit the document with the suggested values, revise the metadata values, or cancel the checkin or update.



Note: If the optional AddCCToNewCheckin component is installed and enabled, clicking Check In on the Content Check In Form performs steps 2 through 6, above, and automatically completes the check in process, provided the properties for dDocTitle are set to *Override Contents*. If the properties of dDocTitle are not set to *Override Contents*, then an alert is displayed requesting that the required field is completed. Field properties are set using the CC Admin Applet. See [Defining Field Properties \(Optional\)](#) (page 2-2).

Batch Mode: Process

The system administrator performs the following steps during this process:

1. Run the *BatchCategorizer* application. The application may also be started in Windows by clicking **Start—Programs—Content Server—<instance_name>—BatchCategorizer**.

The Batch Categorizer screen is displayed.

2. On the Batch Categorizer screen, define filters and release date information to display a list of content to be categorized.



Note: Options on the Batch Categorizer screen (see following steps) enable you to further define the exact content to be categorized.

3. Click **Categorize**.

The Categorize Existing screen is displayed.

4. Select Live Update or Batch Loader.
 - ❖ Use Live Update option to update the data in the repository immediately.

- ❖ Use Batch Loader option to create a control file, which is the output of the Content Categorizer process. The file contains an entry for each source document, and contains the values for each metadata field based on the search rules defined in Content Categorizer.



Tech Tip: You can edit/filter this file before submitting it to Batch Loader, or submit it directly to Batch Loader; see next step.

5. To run the Batch Loader utility automatically after the Content Categorizer process is complete, select the **Run Batch Loader** check box.
6. Enter the location and file name for the log file. The log file will contain error information about the Content Categorizer process.
7. Choose Categorize All or Categorize Selected.
 - ❖ Use Categorize All option to categorize all the content items displayed in the content list.
 - ❖ Use Categorize Selected option to categorize only the selected (highlighted) content items displayed in the content list.
8. Choose to categorize Latest Revision or All Revisions.
 - ❖ Use Latest Revision option to categorize only the most recent revision of the content items displayed in the content list.
 - ❖ Use All Revisions option to categorize all revisions of the content items displayed in the content list.
9. Choose to continue or discontinue the categorization process when Batch Categorizer encounters an error.
10. Click **OK**. The Progress bar shows the progress as the batch process moves through its steps:
 - a. Content Categorizer locates the source content.
 - b. If the content is already in XML format, no translation occurs, and the process continues at step d.
 - c. If the content is not already in XML format, conversion into XML occurs using the selected XML conversion method: Flexiondoc or SearchML.
 - d. Content Categorizer applies the search rules to the XML and obtains values for the specified metadata fields.
 - e. If Live Update was specified, database records are updated immediately. If Batch Loader was specified, an output control file is created, and the Batch Loader utility is run, if the option to do so after processing was specified.

11. When the batch process is complete, review the error logs. Errors encountered by Batch Categorizer are displayed on the console and also recorded in the batch categorizer log (if specified). Errors encountered by Batch Loader are displayed on the console and also recorded in the Content Server system log.



Note: If the optional AddCCToArchiveCheckin component is installed and enabled, all content loaded into content server using the Batchloader utility is categorized automatically, based on predefined rule sets. For more information about defining rule sets, see [Rule Sets Tab](#) (page 3-8).

For more information on batch loading files, see Working with Batches of Files in the *Content Server System Administration Guide*.

SUPPORTED PLATFORMS

The current version of Content Categorizer supports SearchML and Flexiondoc on the following Content Server platforms:

Platform	Version	SearchML	Flexiondoc
HP-UX (RISC)	11i v2	X	
IBM AIX (eServer pSeries)	5.2, 5.3	X	
Sun Solaris (SPARC)	9, 10	X	X
Sun Solaris (Intel)	9, 10	X	X
SuSE Linux (x86)	9, 10	X	X
SuSE Linux (IBM zSeries, 32-bit)	9 SP2	X	
SuSE Linux (Intel)	9 SP2, 10	X	
Red Hat Linux (x86)	ESn3, ES 4, AS 3, AS 4	X	X
MS Windows (32-bit)	2000	X	X
MS Windows (32-bit)	2003	X	X

SETTING UP CONTENT CATEGORIZER

OVERVIEW

Before using Content Categorizer, you must install and configure the necessary software.



Note: Refer to [Supported Platforms](#) (page 1-8) for information about the current Content Server platforms on which SearchML and Flexiondoc are supported.

This topic covers the following procedures:

- ❖ [Setting XML Conversion Method](#) (page 2-1)
- ❖ [Defining Field Properties \(Optional\)](#) (page 2-2)

SETTING XML CONVERSION METHOD

When operating in Interactive mode or Batch mode, the method that Content Categorizer uses to convert native documents into XML is set as a runtime configuration parameter.



Note: Refer to [Supported Platforms](#) (page 1-8) for information about the current Content Server platforms on which SearchML and Flexiondoc are supported.

To set the XML conversion method in Content Categorizer:

1. Log into the Content Server as the system administrator.
2. Click the **Administration** link.

3. Click the **Content Categorizer Administration** link (under Administration Pages for *instance_name*).

The CC Admin Applet screen is displayed.

4. On the Configuration tab, select the **sccXMLConversion** property and click **Edit**, or double-click the property.

The Property Config screen is displayed.

5. From the drop-down list, select the desired XML conversion method:

- ❖ Flexiondoc
- ❖ SearchML

6. Click **OK**.

7. Click **Apply** to save the changes, or click **OK** to save the changes and close the CC Admin Applet, Defining Field Properties

Batch Categorizer must be set up to handle assignment of metadata values for required and non-required fields. When batch recategorizing existing content, the Content Categorizer search rule for a particular field will either succeed or fail. Values will be determined as follows:

DEFINING FIELD PROPERTIES (OPTIONAL)

- ❖ When any rule for a field succeeds, the found value is used (in either Batch Loader operations or Live Update operations), unless you have chosen to override the found value with an existing value (Override is set to false).
- ❖ When all rules for a field fail, the default value defined in Field Properties is used (in either Batch Loader operations or Live Update operations), unless you have chosen not to use the default value (Use Default is set to true).



Important: The Content Server Batch Loader utility will fail any insert action that does not have a value for a required field.

To define field properties for the metadata fields in your system:

1. Open the Content Categorizer Admin Applet.
2. Click the Field Properties tab.
3. Select a metadata field to be edited and click **Edit**, or double-click the field.

The Field Properties screen is displayed.

4. Enter a default value for the field.

The default value for an option list field must match one of the values available for that field.

5. Select the **Override** check box if you want the value returned by the categorization process to override an existing value for the field.
6. Select the **Use Default** check box if you want the field's default value to be used if all rules fail (or are not defined) when the categorization process runs.
7. Click **OK**.
8. Repeat steps 3 through 7 for each field to be edited.
9. Click **Save Settings** to save the changes.

USING THE CC ADMIN APPLET

OVERVIEW

This section covers the following topics:

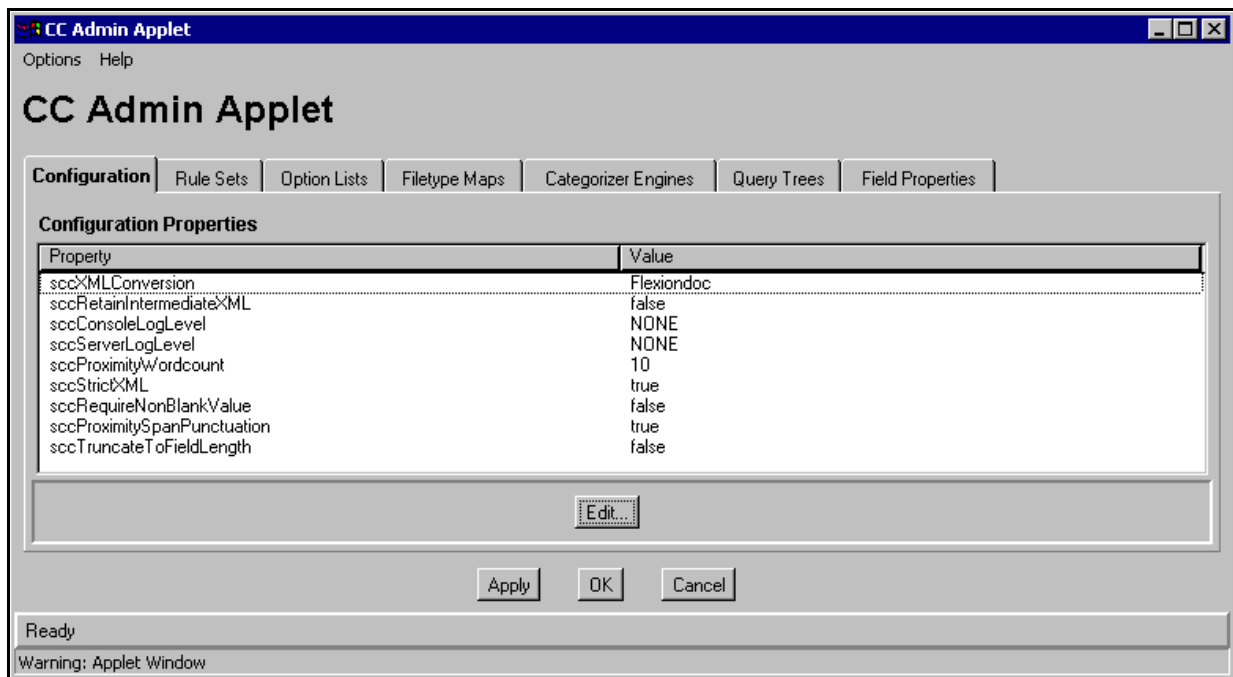
- ❖ [Introduction to CC Admin Applet](#) (page 3-2)
- ❖ [CC Admin Applet Page](#) (page 3-2)
- ❖ [Configuration Tab](#) (page 3-4)
- ❖ [Rule Sets Tab](#) (page 3-8)
- ❖ [Option Lists Tab](#) (page 3-15)
- ❖ [Filetype Maps Tab](#) (page 3-17)
- ❖ [Categorizer Engines Tab](#) (page 3-20)
- ❖ [Query Trees Tab](#) (page 3-23)
- ❖ [Field Properties Tab](#) (page 3-28)

INTRODUCTION TO CC ADMIN APPLET

The CC Admin Applet is accessed by clicking the Content Categorizer link on the Administration page in Content Server.



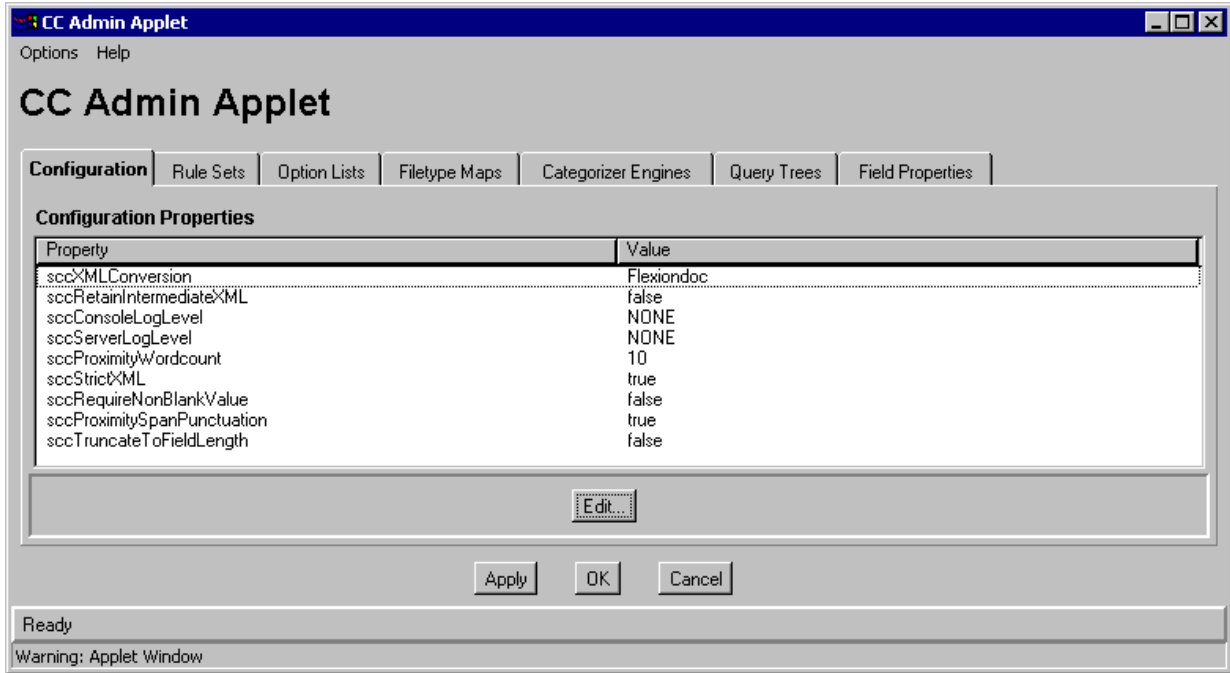
CC ADMIN APPLET PAGE



The CC Admin Applet is used to define the configuration and search rules for Content Categorizer, register categorizer engines, and build query trees.

Feature	Description
Options menu	<p>Save Tab Changes = Saves the settings on the current tab.</p> <p>Save All Tab Changes = Saves the settings on all tabs.</p> <p>Exit = Closes the CC Admin Applet screen.</p>
Help menu	<p>Help = Opens the online Help system.</p> <p>About Content Categorizer = Displays version information for Content Categorizer and Content Server.</p>
Tabs	<p>The applet screen contains 5 tabs:</p> <ul style="list-style-type: none"> ❖ Configuration Tab (page 3-4) ❖ Rule Sets Tab (page 3-8) ❖ Option Lists Tab (page 3-15) ❖ Filetype Maps Tab (page 3-17) ❖ Categorizer Engines Tab (page 3-20) ❖ Query Trees Tab (page 3-23) ❖ Field Properties Tab (page 3-28)
Apply button	Saves the settings on the current tab.
OK button	Saves the settings on all tabs and exits the CC Admin Applet screen.
Cancel button	Exits the CC Admin Applet screen without saving any changes to settings.

CONFIGURATION TAB



The Configuration tab is used to set Content Categorizer runtime configuration settings.

Feature	Description
Property column	<p>Lists the properties that can be configured:</p> <p>sccXMLConversion = Name of the XML conversion method used to convert native documents into XML when operating in Interactive mode or Batch mode.</p> <p>sccRetainIntermediateXML = If true, all intermediate files (temporary files created during the categorization process of a content document) are retained and INFO trace entries in the server log map the temp files to the content. If false, all intermediate files are deleted after use.</p> <p>sccConsoleLogLevel = Level of execution trace information that will appear in the console window when Content Server is run as a foreground process, not started as a background service.</p> <p>sccServerLogLevel = Level of execution trace information that will appear in the content server log (accessed from the Server Logs link on the Administration page).</p> <p>sccProximityWordcount = Number of words that can lie between two keywords in an Option List search rule for the keywords to be considered “near” each other. Default value is 10. Used with \$\$NEAR\$\$ operator.</p> <p>sccStrictXML = Set to true. (This property should be set to false only if the XML includes have leading whitespace.)</p> <p>sccRequireNonBlankValue = If true, search results will not return blank metadata values if XML tags are empty. If false, search results will include blank metadata values if XML tags are empty.</p> <p>sccProximitySpanPunctuation = If true, leading and trailing punctuation is eliminated from strings during the normalization process. If false, punctuation is retained in strings. Used in Option List search rule.</p> <p>sccTruncateToFieldLength = If true, XML tag contents that are too long for their target metadata fields will automatically be truncated to fit. If false, XML tag contents that are too long for their target metadata fields will produce errors.</p>

Feature	Description
Value column	Shows the current value for each property.
Edit button	Displays the Property Config screen for a selected property.

Property Config Screen



The Property Config screen is used to change the value of the property selected on the Configuration tab.

Feature	Description
Property field	Used to change the setting of the selected property. Either a text field or choice list, depending on the property.
OK button	Applies the change and closes the Property Config screen.
Cancel button	Closes the Property Config screen without applying changes.

Log Levels

The following log levels are listed in order from least to most log information:

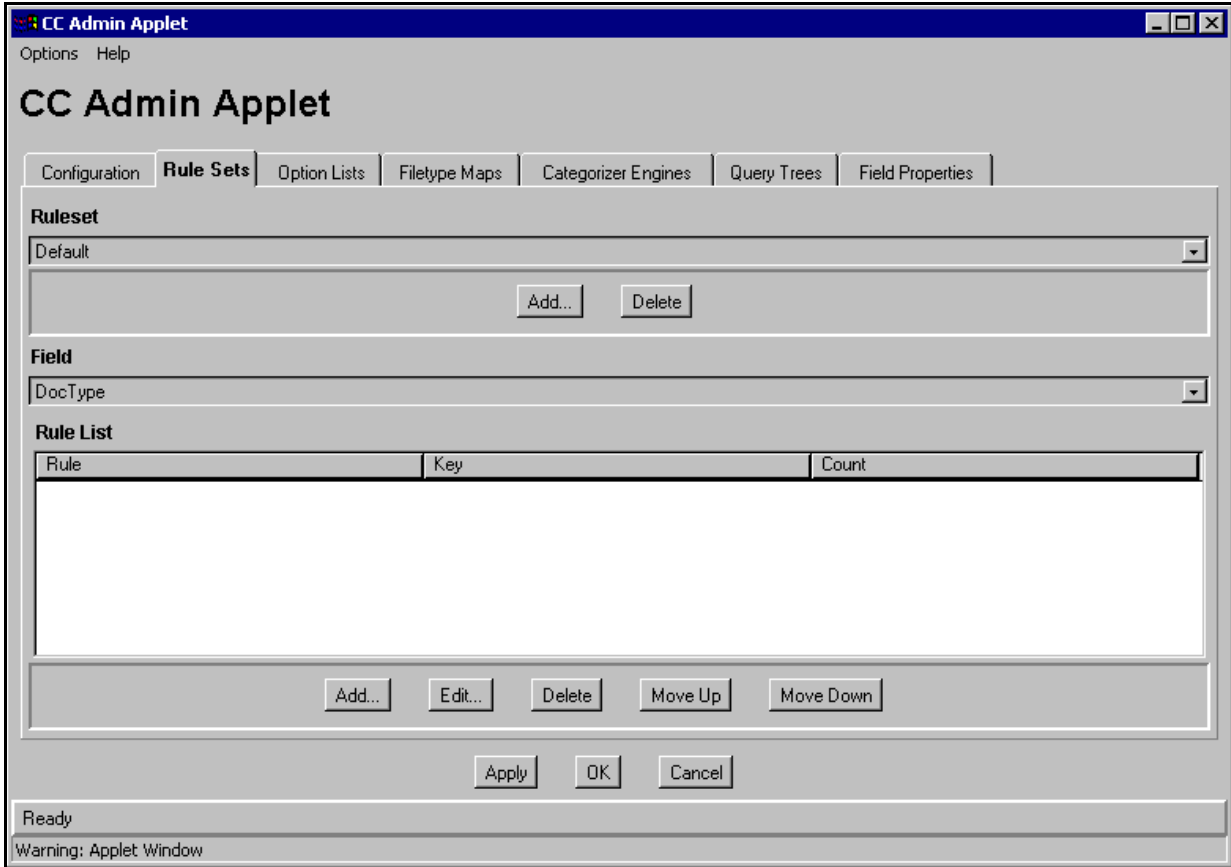
Setting	Level of Information
NONE	Fatal errors only.
ERROR	Operational errors.
WARNING	Actions that are well-defined but unusual.

Setting	Level of Information
INFO	All internal processes. Use for diagnosing problems with execution traces.

Setting Log Levels

1. On the CC Admin Applet Configuration tab, select *sccConsoleLogLevel* or *sccServerLogLevel*.
2. Click **Edit**.
3. Select the desired log level from the drop-down list.
4. Click **OK**.
5. Click **Apply** to save the changes.

RULE SETS TAB

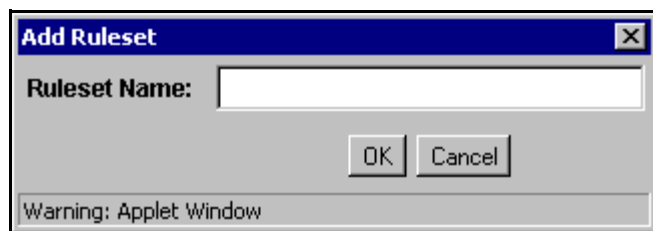


The Rule Sets tab is used to define the search rules that Content Categorizer uses to find metadata values. Named rulesets allows you to define multiple, independent rulesets for use with different types of content. The ruleset to be used for a given content item is determined by a set of specifiers that includes the document type (doc, txt, xls, etc.).

Feature	Description
Ruleset choice list	Lists the search rulesets currently defined. A defined ruleset contains multiple rules that apply to specific documents or a particular document type. If a specific ruleset is not defined for a given document or document type, the default ruleset is used. Rulesets are unitary and completely independent. One ruleset cannot be included in another ruleset and only one ruleset can be applied to a given document.
Add button (Ruleset)	Displays the Add Ruleset Screen (page 3-10), which is used to add a new search rule.
Delete button (Ruleset)	Deletes the selected search ruleset. Note: The default ruleset cannot be deleted.
Field choice list	Selects a metadata field for viewing or defining search rules. Only the metadata fields that allow search rules are included in the list.
Rule List	Lists the search rules currently defined for the selected metadata field. Content Categorizer runs the search rules in the order shown, from the top of the list to the bottom.
Rule column	Shows the type of search rule.
Key column	Shows the search key for the search rule.
Count column	Shows the count for the search rule.
Add button (Rule List)	Displays the Add/Edit Rule for Field Screen (page 3-11), which is used to add a new search rule.
Edit button (Rule List)	Displays the Add/Edit Rule for Field Screen (page 3-11), which is used to edit the selected search rule.
Delete button (Rule List)	Deletes the selected search rule.
Move Up button (Rule List)	Moves the selected search rule up in the list.

Feature	Description
Move Down button (Rule List)	Moves the selected search rule down in the list.

Add Ruleset Screen



The Add Ruleset screen is used to define a new search ruleset on the Rule Sets tab.



Important: Content Categorizer requires a non-empty rule set for any file type—.doc, .txt, .xml, etc.—it is called to examine. If no rules exist for a given file type, Content Categorizer will throw an exception. The easiest way to protect against this is to add at least one rule to the Default rule set. The Default rule set is used for all file types which do not have a custom rule set assigned.

Feature	Description
Ruleset Name field	Used to add a ruleset name.
OK button	Applies the change and closes the Add Ruleset Screen (page 3-10).
Cancel button	Closes the Add Ruleset Screen (page 3-10) without applying changes.

Add/Edit Rule for *Field* Screen

The Add/Edit Rule for *Field* screen is used to define a new search rule or edit the rule selected on the Rule Sets tab.

Feature	Description
Rule choice list	Selects the type of search rule.
Key field	Defines the search key for the search rule. See Key Field (page 3-11) for additional information.
Count field	Defines the count for the search rule. See Count Field (page 3-13) for additional information.
OK button	Applies the change and closes the Add/Edit Rule for Field Screen (page 3-11).
Cancel button	Closes the Add/Edit Rule for Field Screen (page 3-11) without applying changes.

Key Field

Search Rule	Key Description
TAG_TEXT	XML element [without angle brackets]
TAG_ALLTEXT	XML element [without angle brackets]
TEXT_REMAINDER	Text phrase

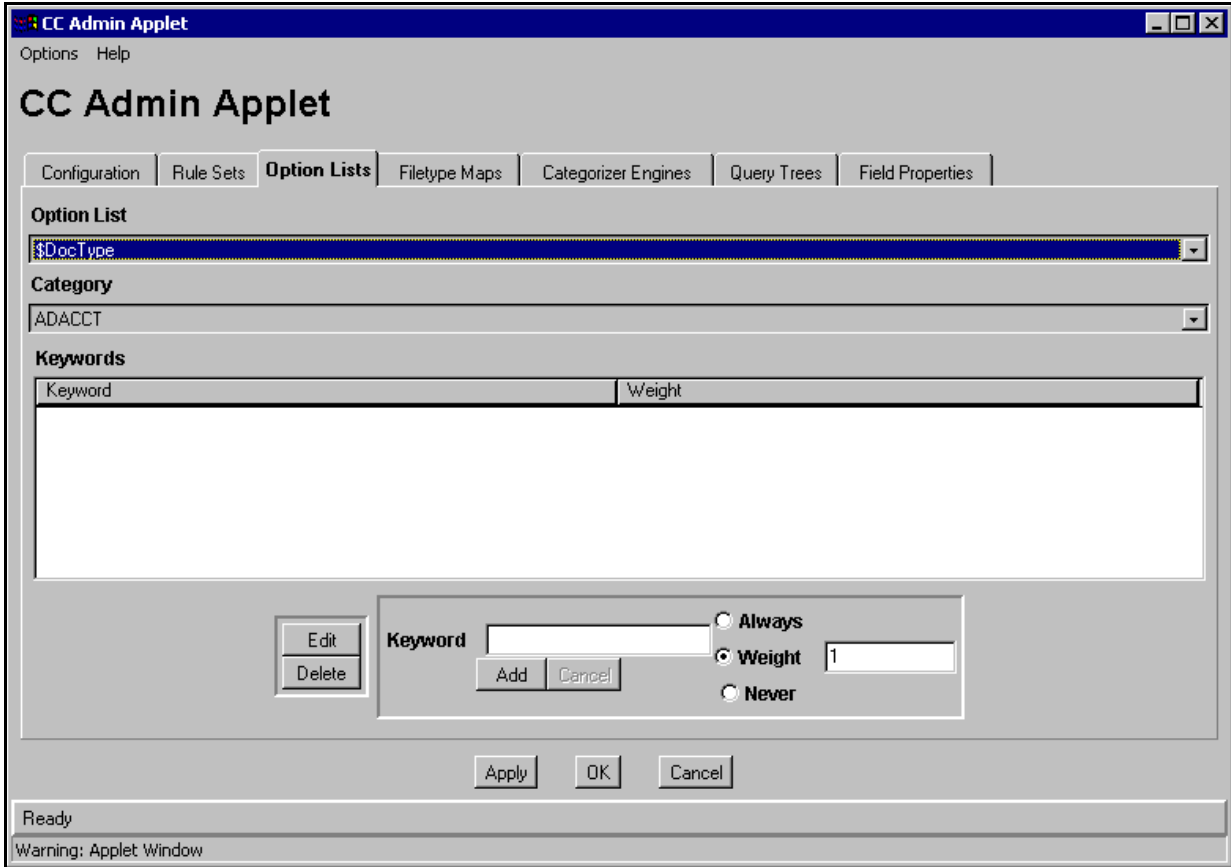
Search Rule	Key Description
TEXT_ALLREMAINDER	Text phrase
TEXT_FULL	Text phrase
TEXT_ALLFULL	Text phrase
TEXT_NEXT	Text phrase
TEXT_ALLNEXT	Text phrase
FIRST_PARAGRAPH	XML element [without angle brackets]
FIRST_SENTENCE	XML element [without angle brackets]
OPTION_LIST	Option List name (must match the Option List name on the Option Lists Tab (page 3-15).
CATEGORY	Categorization engine name (if more than one is defined in list of Categorizer Engines), followed by forward slash (/), followed by taxonomy name. For example: <i>EngineName/TaxonomyName</i>
FILETYPE	<i>Not applicable for this rule; leave this field blank.</i>

Count Field

Search Rule	Count Description
TAG_TEXT	<p>The number of tags or text phrases that must be matched before the rule returns results. For example, a count of 4 will look for the fourth occurrence of the key. If only three occurrences of the key are found in the document, the rule fails.</p> <p>The default count of 1 returns the first occurrence of the key.</p>
TAG_ALLTEXT	
TEXT_REMAINDER	
TEXT_ALLREMAINDER	
TEXT_FULL	
TEXT_ALLFULL	
TEXT_NEXT	
TEXT_ALLNEXT	
FIRST_PARAGRAPH	<p>Size threshold measured in percent. The first paragraph matching the key that is larger than the count percentage multiplied by the average paragraph size is returned.</p> <p>For example, if the count is set to 75 and the average paragraph size is 100 characters, the rule returns the first paragraph larger than 75 characters that matches the key.</p> <p>If the count is set to the default of 1, the rule is likely to return the first paragraph that matches the key.</p>
FIRST_SENTENCE	<p>The number of elements that have their first sentences returned.</p> <p>For example, if the count is set to 3, the rule returns the first sentence from each of the first three elements that match the key.</p>

Search Rule	Count Description
OPTION_LIST	<p>The minimum threshold score for the rule to return results.</p> <p>For example, if the count is set to 50, and the highest accumulated keyword score is 45, the rule fails.</p>
CATEGORY	<p>The minimum confidence level threshold for the rule to return results.</p> <p>For example, if the count is set to 50, and the highest-confidence category has a confidence level of 45, the rule fails.</p>
FILETYPE	<p><i>Not applicable for this rule; leave this field blank.</i></p>

OPTION LISTS TAB

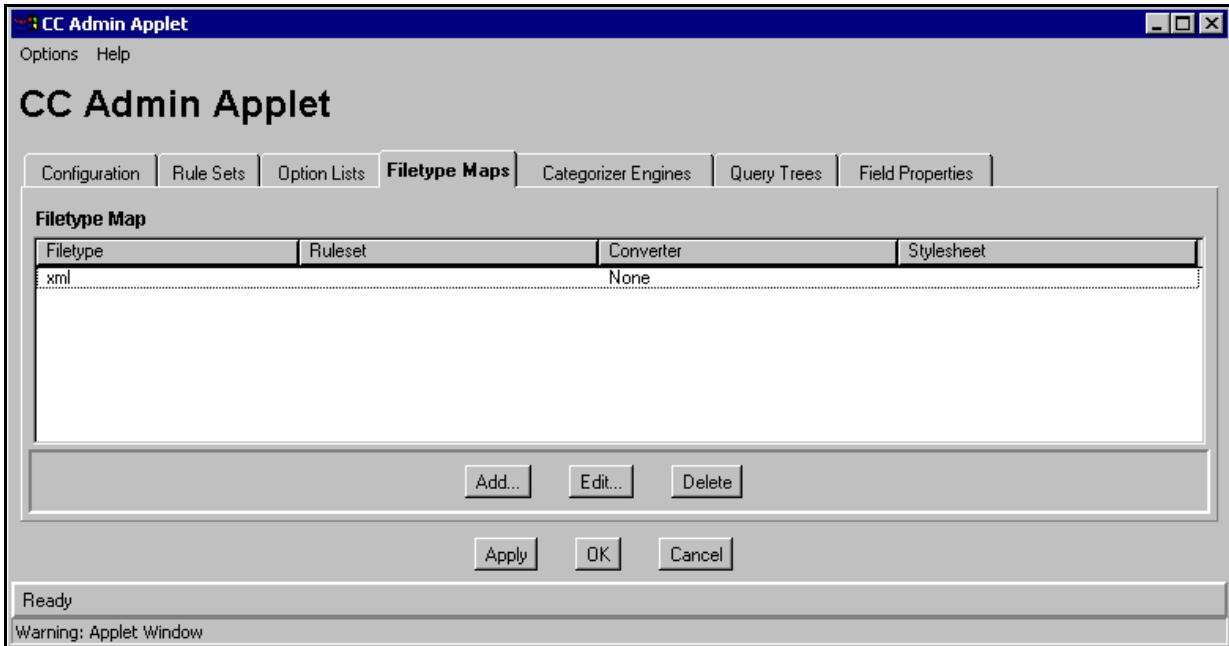


The Option Lists tab is used to define the keywords and weights for Option List search rules.

Feature	Description
Option List choice list	Selects the option list. The list includes the Type (\$DocType) option list, plus option lists of all custom metadata fields that have an option list defined in the Configuration Manager.
Category choice list	Selects a value on the selected option list. Only the pre-defined values for the option list are included.
Keyword column	Lists the keywords associated with the selected category.
Weight column	Shows the weights assigned to each keyword.

Feature	Description
Edit button	Changes the Add button to Update and enters the selected keyword and weight in the editing area.
Delete button	Deletes the selected keyword.
Keyword field	Used to add or edit a keyword.
Add/Update button	Adds a new keyword and weight value, or changes the selected keyword and weight value.
Cancel button	Changes the Update button to Add without saving changes to the selected keyword and weight value.
Weight options	<p>Defines the weight assigned to the keyword.</p> <p>Always = If the keyword is found, the selected category will be returned as the suggested value, regardless of the score.</p> <p>Weight = This number multiplied by the number of occurrences of the keyword contributes to the score for a category. The category with the highest score is returned as the suggested value for the option list field.</p> <p>Never = If the keyword is found, the selected category will not be returned as the suggested value, regardless of the score.</p>

FILETYPE MAPS TAB



The Filetype Maps tab is used to create and modify a filetype map, which determines the specific converter, stylesheet and ruleset are used for a particular content item type (such as doc, txt, xls). If no Filetype Map entry is defined for a particular content item, then the default ruleset, XML Converter, stylesheet, and maximum size values will be used.

Feature	Description
Filetype Map list	Lists the filetype maps currently defined.
Properties columns	Shows the following properties for a filetype map: Filetype name Ruleset Converter Stylesheet
Add button	Displays the Add/Edit Filetype Map Screen (page 3-18), which is used to add a filetype map.

Feature	Description
Edit button	Displays the Add/Edit Filetype Map Screen (page 3-18), which is used to edit the selected filetype map.
Delete button	Deletes the selected filetype map.

Add/Edit Filetype Map Screen

The Add/Edit Filetype Map screen is used to define a new filetype map or edit the map selected on the Filetype Maps tab. The filetype map entries determine which XML converter, which XSLT stylesheet, and which ruleset are to be used for a particular filetype (doc, txt, xle, evt.).

Defining a filetype entry for a particular content file type makes possible special, content-specific processing alternatives. For example, you may use a custom ruleset for a particular content type, set a maximum file size, or direct Content Categorizer to ignore such files altogether.

Feature	Description
Filetype field	Used to specify the filetype (such as doc, txt, xls). Enter \$\$NONE\$\$ to define a map entry for files with no filetype.

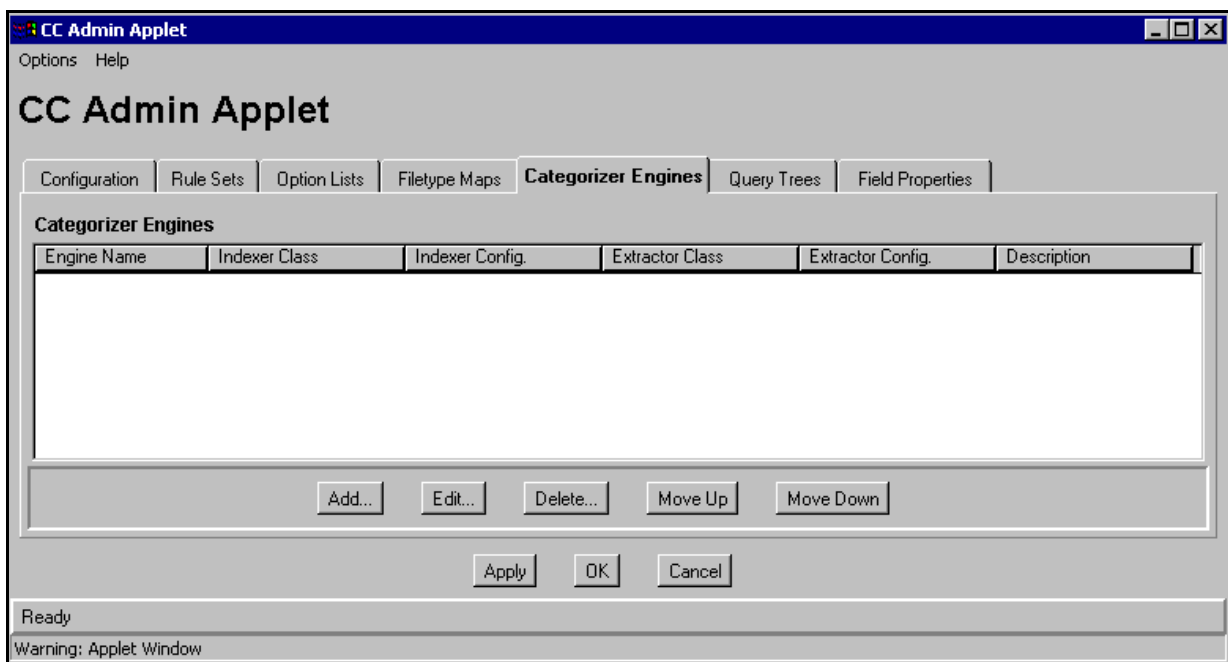
Feature	Description
Ruleset Name field	Used to specify the ruleset to be applied. If no ruleset is specified, the default ruleset will be used.
XML Converter list	Lists the applicable XML converter. Three values are available: Flexiondoc, SearchML, and None. (None is available if the content is already in XML format or if it will be when the custom XSLT stylesheet completes its processing.) The default value for the XML converter is set as a configuration parameter. See Configuration Tab (page 3-4).
XSLT Stylesheet field	<p>Used to specify the file path (absolute or relative) of the XSLT stylesheet to be applied. XSLT stylesheets are stored in the <code><install_dir_path>/custom/ContentCategorizer/stylesheets/</code> directory.</p> <p>Customized, content-specific XSLT translation stylesheets can be used as the back-end of the XML conversion. The selected stylesheet is used in a transformation operation after the XML conversion (if any) is complete. Stylesheets allow document properties and content to be isolated with the results extracted to a metadata field.</p> <p>The stylesheet to be used for a given content item is determined by a set of specifiers that includes the document type (doc, txt, xls, etc.). The default value for XSLT Stylesheet depends on the XML converter actually used. Flexiondoc uses <code>flexion_to_scc.xsl</code> and SearchML uses <code>searchml_to_scc.xsl</code>.</p>
Maximum File Size field	Used to establish the maximum file size for a given file type (doc, txt, xls, etc.).
Ignore check box	<p>Selected = directs Content Categorizer to disregard a particular file type (doc, txt, xls, etc.).</p> <p>Clear = directs Content Categorizer to consider all file types.</p>
OK button	Saves the settings in all the fields and exits the Add/Edit Filetype Map Screen (page 3-18).
Cancel button	Exits the Add/Edit Filetype Map Screen (page 3-18) without saving any changes to settings.

Filetype Mapping Operation

When a document is submitted to Content Categorizer for processing, the following shows the selection process to determine the applicable combination of a ruleset, XML converter, and XSLT stylesheet:

1. The check-in page is checked for certain Content Categorizer specific field values (sccRuleset, sccXMLConversion, and sccXSLTStylesheet). If the field exists and is non-blank, the contents of the field determine which ruleset, XML converter, or XSLT stylesheet is to be used.
2. If the form field is missing or empty, a filetype map entry for the content type is sought. If the map entry exists and if the relevant field is non-blank, the contents are used to determine which ruleset, XML converter, maximum file size, or XSLT stylesheet is to be used. If the map entry exists, and the Ignore flag is set, Content Categorizer will not perform the XML conversion or apply any search rules.
3. If the form field is missing or empty and the filetype map entry is missing or the relevant field of the map entry is blank, the default ruleset is used, the maximum file size is used, the default XML converter defined in the configuration settings is used, and/or the default XSLT stylesheet is used.

CATEGORIZER ENGINES TAB



The Categorizer Engines tab is used to register a third-party categorization engine for integration with Content Categorizer. The values used to register an engine will normally be provided by the third-party categorization vendor.

Feature	Description
Categorizer Engines list	Lists the Categorizer Engines currently defined.
Properties columns	Shows the following properties for a Categorizer Engine: Engine Name Indexer Class Indexer Configuration Extractor Class Extractor Configuration Description
Add button	Displays the Add Categorizer Engine screen, which is used to register a new engine.
Edit button	Displays the Edit Categorizer Engine screen, which is used to edit the selected engine.
Delete button	Deletes the registration for the selected engine.
Move Up button	Moves the selected engine up in the list.
Move Down button	Moves the selected engine down in the list.

Add/Edit Categorizer Engine Screen

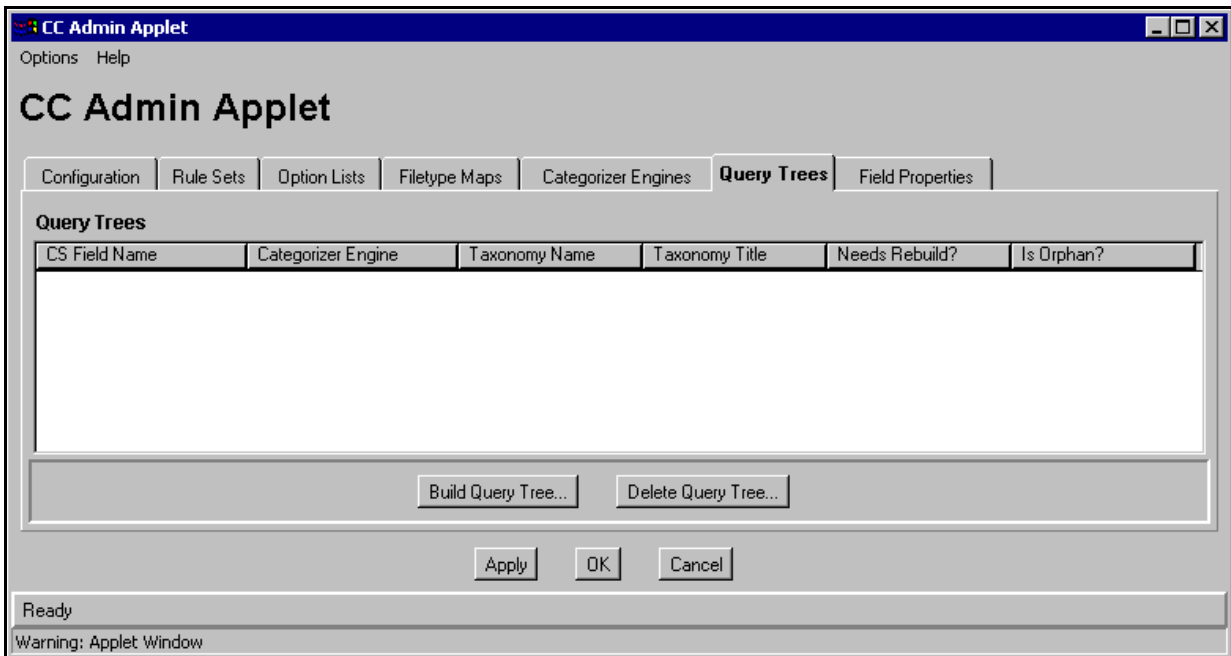


Note: Adaptor modules for Autonomy's *Categorizer* engine and APR Smartlogik's *Structure* engine have been developed and are available.

Feature	Description
Engine Name field	A unique identifier for a categorization engine for which SCC has an adaptor module. This identifier is used in the Key field of a CATEGORY rule, for example, " EngineName / <i>TaxonomyName</i> ".
Indexer Class field	The name of the Java class in an SCC adaptor module that is used to ask the categorizer engine to categorize a given document or set of documents.
Indexer Configuration field	A string that is passed to the Indexer Class's setup() method. It is usually a comma-separated list of engine-specific initialization parameters.
Extractor Class field	The name of the Java class in an SCC adaptor module that is used to ask the categorizer engine for the set of categories in a given taxonomy.
Extractor Configuration field	A string that is passed to the Extractor Class's doExtract() method. It is usually a comma-separated list of engine-specific initialization parameters.
Description	A description of the categorization engine.

Feature	Description
OK button	Saves the settings and exits the screen.
Cancel button	Exits the screen without saving changes to settings.

QUERY TREES TAB



The Query Trees tab is used to create a browsable hierarchy of categorized documents. A query tree consists of:

- ❖ a “taxonomy cache” of machine-generated files located in
`<install_dir_path>/data/contentcategorizer/taxonomies/<engine_name>/<taxonomy_name/`
- ❖ a “root link” on Content Server's Library page

Query trees are associated with a CATEGORY rule, and are generated from the taxonomy used in the CATEGORY rule definition. If you do not have a categorization engine defined, you will not be able to generate a Query Tree.

Feature	Description
Query Trees List	Lists the Query Trees currently defined.
Properties columns	Shows the following properties and status for a Query Tree: CS Field Name Categorizer Engine Taxonomy Name Taxonomy Title Needs Rebuild? Is Orphan?
Build Query Tree button	Displays the Build Query Tree Screen (page 3-24).
Delete Query Tree button	Displays the Delete Query Tree Screen (page 3-26).

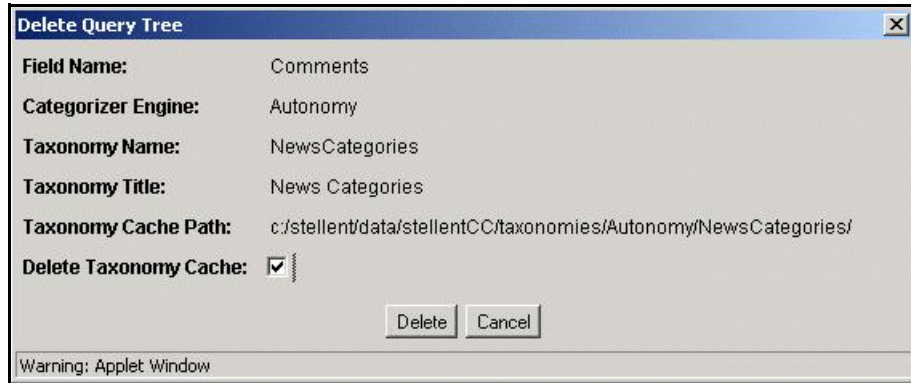
Build Query Tree Screen

The Build Query Tree screen is accessed from the Query Trees tab.

Feature	Description
Field Name	Name of the content information (metadata) field for which the CATEGORY search rule was defined.

Feature	Description
Categorizer Engine	Name of the categorizer engine specified for the CATEGORY search rule that was defined for the metadata field.
Taxonomy Name	Name of the taxonomy specified for the CATEGORY search rule that was defined for the metadata field. In addition to being displayed as a property in the Query Trees list, this name is displayed and used: in the Content Server Library navigation hierarchy. in the Value field of the Query Link Definition screen in Web Layout Editor.
Taxonomy Title field	Title of taxonomy. In addition to being displayed as a property in the Query Trees list, this name is displayed and used: as a root link on the Content Server Library main page. in the Page Links list of the Web Layout Editor screen. in the Link Title field of the Query Link Definition screen in Web Layout Editor.
Taxonomy Description field	Description of taxonomy, which is used: under the root link on the Content Server Library main page. in the Description field of the Query Link Definition screen.
Taxonomy Cache Path	Path to the taxonomy cache: <i><install_dir_path>/data/contentcategorizer/taxonomies/engine_name/taxonomy_name/</i>
Build Taxonomy Cache check box	Selected = Clicking OK builds (or rebuilds) the taxonomy cache. Clear = Clicking OK does not build (or rebuild) the taxonomy cache.
OK button	Saves the settings and builds (or rebuilds) the query tree.
Cancel button	Exits the screen without building (or rebuilding) the query tree and without saving changes to settings.

Delete Query Tree Screen



The Delete Query Tree screen is accessed from the Query Trees tab.

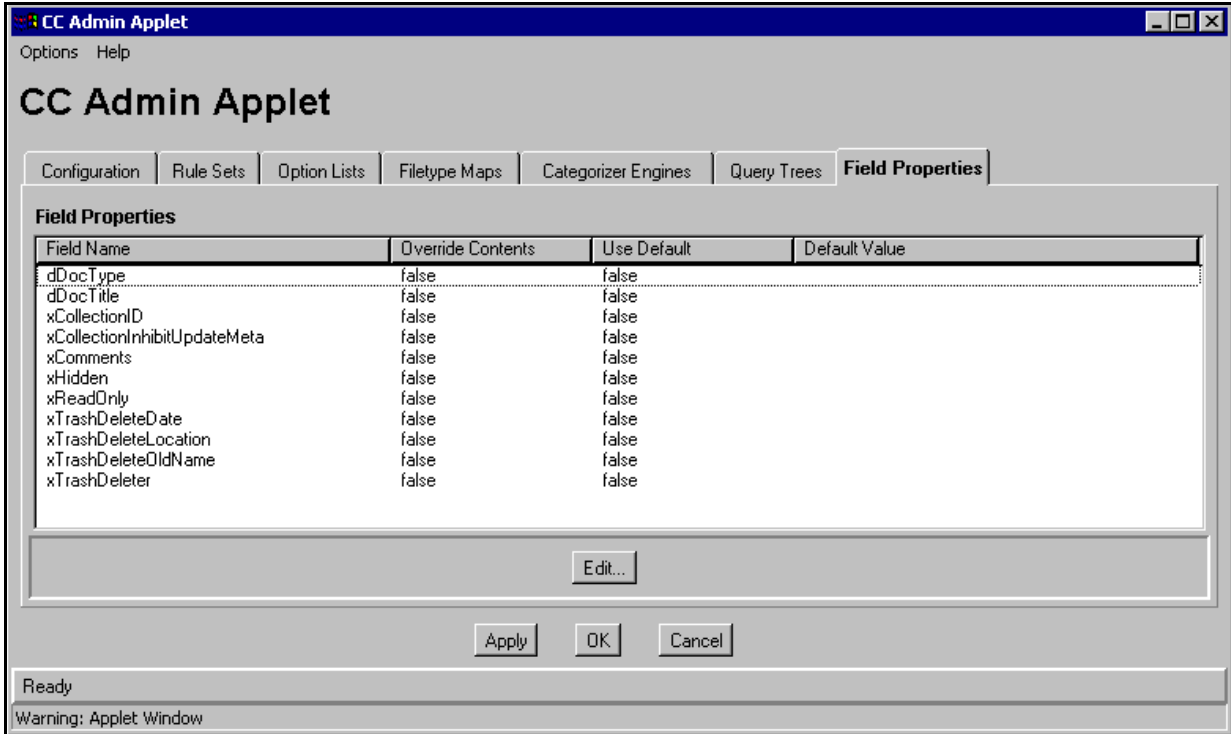
Feature	Description
Field Name	Name of the content information (metadata) field for which the CATEGORY search rule was defined.
Categorizer Engine	Name of the categorizer engine specified for the CATEGORY search rule that was defined for the metadata field.
Taxonomy Name	Name of the taxonomy specified for the CATEGORY search rule that was defined for the metadata field. In addition to being displayed as a property in the Query Trees list, this name is displayed and used: in the Content Server Library navigation hierarchy. in the Value field of the Query Link Definition screen in Web Layout Editor.
Taxonomy Title field	Title of taxonomy. In addition to being displayed as a property in the Query Trees list, this name is displayed and used: as a root link on the Content Server Library main page. in the Page Links list of the Web Layout Editor screen. in the Link Title field of the Query Link Definition screen in Web Layout Editor.

Feature	Description
Taxonomy Cache Path	Path to the taxonomy cache: <install_dir_path>/data/contentcategorizer/taxonomies/engine_name/taxonomy_name/
Delete Taxonomy Cache check box	Selected = Clicking Delete deletes the taxonomy cache. Clear = Clicking Delete does not delete the taxonomy cache.
Delete button	Deletes the selected taxonomy cache.
Cancel button	Exits the screen without building (or rebuilding) the query tree and without saving changes to settings.

Needs Rebuild? and Is Orphan? Status

Feature	Description
Needs Rebuild? column	No indicates that the query tree is current. No action is required. Yes indicates that the taxonomy cache, the Library page link to the cache, or both, are old or do not exist. A rebuild of the query tree is necessary.
Is Orphan? column	No indicates that the query tree is not an orphan. No action is required. Yes indicates that the taxonomy cache, the Library page link to the cache, or both, are orphans; that is, there are no longer any CATEGORY rules that refer to them. Check the rule associated with the query tree.

FIELD PROPERTIES TAB

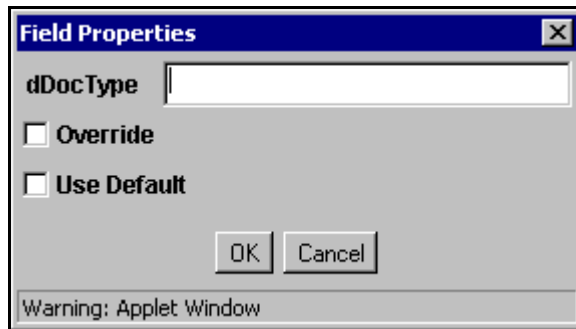


The Field Properties tab is used to set override and default properties that Content Categorizer will use during Batch mode.

Feature	Description
Field Properties List	Lists the standard and custom content information (metadata) fields and their property settings.
Field Name column	Lists the metadata fields that can have default values assigned.
Override Contents column	Shows whether a field's contents will be overridden if the field already has an existing value. True = value returned by the categorization process will override an existing value. False = value returned by the categorization process will not override an existing value.

Feature	Description
Use Default column	Shows whether a field's default value will be used if all rules fail (or are not defined) when the categorization process runs. True = default value will be used. False = default value will not be used.
Default Value column	Shows the default value for each field.
Edit button	Displays the Field Properties Screen (page 3-29).

Field Properties Screen



The Field Properties screen is used to edit settings for a field selected in the Field Properties list.

Feature	Description
Metadata field	Used to change the default value of the selected field.
Override check box	Selected = sets Override to true. Clear = sets Override to false.
Use Default check box	Selected = sets Use Default to true. Clear = sets Use Default to false.
OK button	Applies edits and closes the Field Properties screen.
Cancel button	Closes the Field Properties screen without applying changes.

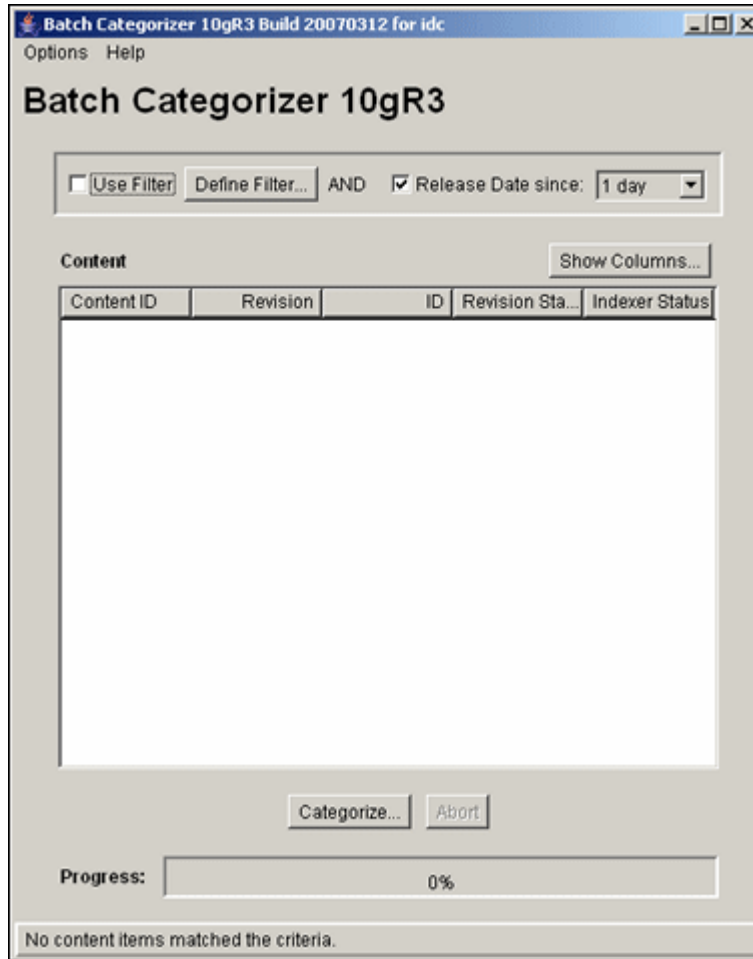
USING BATCH CATEGORIZER

OVERVIEW

This section covers the following topic:

- ❖ [Batch Categorizer Screen](#) (page 4-2)

BATCH CATEGORIZER SCREEN



The Batch Categorizer utility is used to run Content Categorizer in Batch mode and is accessed by running the *BatchCategorizer* application, which is located in the `<cs_install_dir_path>/custom/ContentCategorizer/` directory. The application may also be started in Windows by clicking **Start—Programs—Content Server—<instance_name>—BatchCategorizer**.

Feature	Description
Options menu	Save Settings = Saves all the current settings. Exit = Closes Batch Categorizer.

Feature	Description
Help menu	<p>Help = Opens the online Help system.</p> <p>About Content Categorizer = Displays version information for Content Categorizer and Content Server.</p>
Use Filter check box	<p>Selected = enables filtering of content in the repository based on any defined filters and release date filter (if specified).</p> <p>Clear = disables filtering of content in the repository based on defined filters.</p>
Define Filter button	Displays the Define Filter Screen (page 4-4), from which selections for filters can be made.
Release Date since check box and list	<p>Selected = enables filtering of content in the repository based on release date since 1 day, 1 week, or 4 weeks.</p> <p>Clear = disables filtering of content in the repository based on release date.</p>
Content list	Filtered lists of content items in the repository.
Show Columns button	Displays the Show Columns Screen (page 4-10), from which selections for displaying properties in the content list can be made.
Content list properties columns	Shows the following properties for each content item: Content ID Revision ID Revision Status Indexer Status
Categorize button	Displays the Categorize Existing Screen (page 4-11).
Abort button	Cancels the categorization batch process.
Progress bar	Shows the progress of the batch process.

Define Filter Screen

The Define Filter screen is accessed from the Batch Categorizer utility.

Feature	Description
Content ID check box and field	The unique identifier of the content item. Enter the value. Selected / Clear = Item is used / not used as a filter.
Title check box and field	The descriptive name identifying the file. Enter the value. Selected / Clear = Item is used / not used as a filter.

Feature	Description
Author check box and field	Name of the person who checked in the file. Enter the value. Selected / Clear = Item is used / not used as a filter.
Type check box and field	An identifier used to group files. Select value from list. Selected / Clear = Item is used / not used as a filter.
Security Group check box and field	A security group is a set of files with the same access privileges. Select value from list. Selected / Clear = Item is used / not used as a filter.
Checked out check box and field	Existing files that are checked out. Select value from list. Selected / Clear = Item is used / not used as a filter.
Checked out by check box and field	Person who has checked out the file(s). Enter the value. Selected / Clear = Item is used / not used as a filter.
Revision Status check box and field	Represents the number of revisions in the file's life cycle. Select value from list: <ul style="list-style-type: none"> • Done: Ready for release on release date. • Edit: File in workflow, waiting for contributors and/or reviewers. • GenWWW: Waiting for refinery process, or failed trying to index and release to web sites. • Released: Processing completed. File is viewable and searchable on the web site (if permitted by release date). • Pending: Waiting for completion of all files within a workflow. • Expired: File removed from web sites. • Deleted: Temporary state until next index cycle removes file completely. • Review: File in a workflow review step. Selected / Clear = Item is used / not used as a filter.

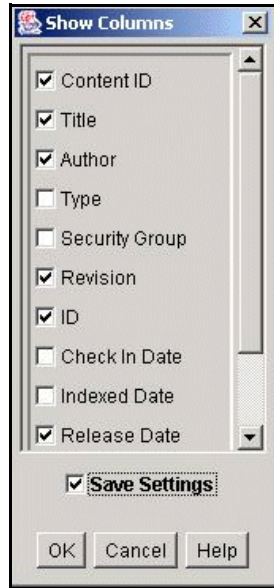
Feature	Description
Indexer Status check box and field	<p>The status of the file's Index Refinery. Select value from list:</p> <ul style="list-style-type: none"> • New: Files checked in and released to web site. • Current: Latest revision of file is released. (Only one version of file can be Current.) • Old: Prior revisions and files. • Workflow: After Done status and is in a workflow. • Processing: Preparing files for indexer. • Indexing: File being indexed. • Update: File is currently being updated by another user. <p>Selected / Clear = Item is used / not used as a filter.</p>
Conversion Status check box and field	<p>The status of all defined connections. Select value from list:</p> <ul style="list-style-type: none"> • Processing: File being converted by Document Refinery. • Converted: Conversion successful and web-viewable version of file is viewable. • MetaData Only: Full-text index bypassed and only metadata is indexed. • Refinery PassThru: Document Refinery failed to convert the file and passed the native file through to the web. • Failed: File deleted, locked, or corrupted; indexer error. • Incomplete Conversion: An error occurred in the conversion. A step after a valid web-viewable file was produced. This file is full-text indexed. <p>Selected / Clear = Item is used / not used as a filter.</p>

Feature	Description
Indexer Cycle check box and field	<p>The status of the index cycle. Select value from list:</p> <ul style="list-style-type: none"> • Idle: Indexer cycle is complete and the content is not in an Indexer cycle. • Loading for Active: The content is being loaded for an update cycle. • Indexed for Active: The content is being indexed during an update cycle. • Loading for Rebuild: The content is being loaded for a rebuild cycle. • Indexed for Rebuild: The content is being indexed during a rebuild cycle. • Rebuilt: The content has been processed by a rebuild cycle. • Updated: The content has been processed by an update cycle. <p>Selected / Clear = Item is used / not used as a filter.</p>
Workflow State check box and field	<p>Revision state of content in a given workflow.</p> <ul style="list-style-type: none"> • Reviewer/Contributor: Workflow step consisting of named users allowing check-in, approval, and rejection. • Contributor: Initial step in a basic workflow consisting of a predefined users list. • Reviewer: Subsequent step in a workflow consisting of named users allowing approval and rejection. • Pending: Workflow is waiting for next user's contribution. <p>Selected / Clear = Item is used / not used as a filter.</p>

Feature	Description
Publish Type check box and field	<p>The published project type. This field is only used for files that originate from Content Publisher. If the content item checked in is not a Content Publisher file, the Publish Type field will default to None and the Publish Status field will default to Content. Select value from list:</p> <ul style="list-style-type: none"> • None: The system default for content items that do not originate at Content Publisher files. • Contributor: A resource created by a Content Publisher user. • Gallery: A gallery file from the content publisher. • Home: A Home page file from the publisher project. • Page: Any other page from the publisher project. • Navigation: A navigation graphic file from the publisher project. • Other: Any other type of publisher project file. • Support: A publisher project support file. <p>Selected / Clear = Item is used / not used as a filter.</p>
Publish Status check box and field	<p>The status of the published project type. This field is only used for files that originate from Content Publisher. If the content item checked in is not a Content Publisher file, the Publish Type field will default to None and the Publish Status field will default to Content. Select value from list:</p> <ul style="list-style-type: none"> • Content: The source file. The system default for content items that do not originate as Content Publisher files. • Published: The file is released to the published Web site. • Staging: The file is released to the staging Web site. • Workflow: The file is in the staging workflow. <p>Selected / Clear = Item is used / not used as a filter.</p>
Latest Revision check box	<p>The most current revision of the file.</p> <p>Selected / Clear = Item is used / not used as a filter.</p>

Feature	Description
Inhibit Propagation check box and field	Propagation enables default metadata values to be copied from a higher-level content item and applied to other lower-level content items. Selected / Clear = Item is used / not used as a filter.
Comments check box and field	Words or phrases used to narrow the search and selected file results. Enter the word or phrase. Selected / Clear = Item is used / not used as a filter.
Hidden check box and field	Specifies whether the content is hidden from the folder view. Selected / Clear = Item is used / not used as a filter.
Read Only check box and field	Specifies whether the content should not be altered. Selected / Clear = Item is used / not used as a filter.
Trash Delete Old Name check box and field	Defines a file name metadata field for the Trash function which enables the original file name to be recorded as metadata for items that are moved to the Trash folder. Enter the value. Selected / Clear = Item is used / not used as a filter.
Trash Deleter check box and field	Defines a user metadata field for the Trash function which enables the user's login information to be recorded as metadata for items that are moved to the Trash folder. Enter the value. Selected / Clear = Item is used / not used as a filter.
OK button	Saves the filter settings and exits the Define Filter dialog.
Cancel button	Exits the Define Filter dialog without saving any changes or filter settings.
Help button	Opens the online Help system.

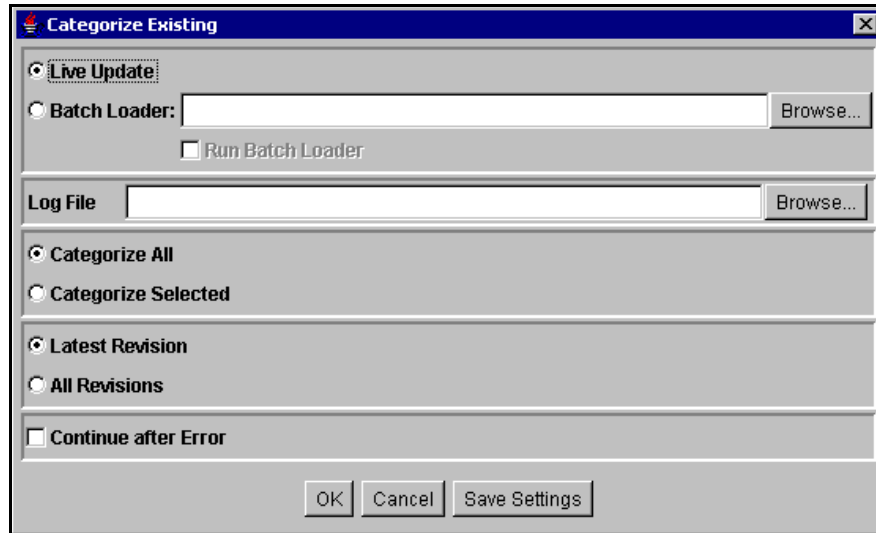
Show Columns Screen



The Show Columns screen is accessed from the Batch Categorizer utility.

Feature	Description
Properties columns check boxes	Selected = Displays the column on the Batch Categorizer Screen (page 4-2). Clear = Does not display the column on the Batch Categorizer Screen (page 4-2).
Save Settings check box	Selected = The column settings are applied every time the Batch Categorizer Screen (page 4-2) is opened. Clear = The column settings apply only until the Batch Categorizer Screen (page 4-2) is closed.
OK button	Saves the filter settings and exits the Define Filter dialog.
Cancel button	Exits the Define Filter dialog without saving any changes or filter settings.
Help button	Opens the online Help system.

Categorize Existing Screen



The Categorize Existing screen is accessed from the Batch Categorizer utility.

Feature	Description
Live Update option	When Live Update is selected, existing selected repository content is categorized and updated.
Batch Loader option	When Batch Loader is selected, existing selected repository content is categorized, and Update Records are written to the Batch Loader control file.
Batch Loader field	Defines the location and file name for the Batch Loader control file. A typical file name is <i>batchinsert.txt</i> . You can edit/filter this file before submitting it to Batch Loader, or submit it directly to Batch Loader by selecting the Run Batch Loader check box.
Run Batch Loader check box	Selected = Clicking OK runs Content Categorizer and then runs the Batch Loader utility. Clear = Clicking OK runs only Content Categorizer.
Log File field	Defines the location and file name for the Content Categorizer log file.

Feature	Description
Categorize options	<p>Categorize All = all content items in the filtered list of content will be categorized.</p> <p>Categorize Selected = only the selected content items in the filtered list of content will be categorized.</p>
Revision options	<p>Latest Revision = only the latest revision of content items that have been selected for categorization will be categorized.</p> <p>All Revisions = all revisions of content items that have been selected for categorization will be categorized.</p>
Continue after Error check box	<p>Selected = Batch Categorizer continues processing when it encounters an error and logs the error.</p> <p>Clear = Batch Categorizer stops processing when it encounters an error.</p>
OK button	Runs the categorization process, and if selected, the Batch Loader utility.
Cancel button	Exits without running the categorization process.
Save Settings button	Saves the settings on the dialog screen.

SEARCH RULES

OVERVIEW

This section includes the following topics:

- ❖ [Understanding Search Rules](#) (page 5-1)
- ❖ [Pattern Matching Search Rules](#) (page 5-3)
- ❖ [Abstract Search Rules](#) (page 5-8)
- ❖ [Option List Search Rule](#) (page 5-11)
- ❖ [Categorization Engine Search Rule](#) (page 5-15)
- ❖ [Filetype Search Rule](#) (page 5-16)
- ❖ [Defining Search Rules](#) (page 5-17)

UNDERSTANDING SEARCH RULES

Search rules define how Content Categorizer determines metadata values to return to the Content Check In Form or Info Update Form (for Interactive mode) or the batch file (for Batch mode).

Every search rule is defined by:

- ❖ A **rule type**, which determines the method that Content Categorizer uses to search the XML document.

- ❖ A **key**, which defines the XML element, phrase, or keyword that Content Categorizer looks for in the document, or the categorization engine/taxonomy that Content Categorizer uses to classify the document.
- ❖ A **count**, which is used to refine the search criteria.



Note: Keys and counts are explained in more detail in the Help topics for each search rule type.

Search Rule Types

Metadata values can be derived using these methods:

- ❖ A Pattern Matching search rule looks for specific text or a specific XML element and returns an associated value.
- ❖ An Abstract search rule looks for an XML element and returns a descriptive sentence or paragraph from that element.
- ❖ An Option List search rule looks for keywords within the source document, applies a score for each keyword found, and returns the option list value that has the highest keyword score.
- ❖ A Categorization Engine search rule uses a third-party categorization engine and a defined taxonomy to determine appropriate metadata values.
- ❖ A Filetype search rule examines the filename extension of the primary file and returns a term associated with that filename extension.

Search Rule Guidelines

- ❖ Search rules can be applied to any custom metadata field.
- ❖ Search rules can be applied to the Title, Comments, and Type standard metadata fields. Search rules cannot be defined for any other standard metadata fields (such as Author, Security Group, and Account).
- ❖ Multiple search rules can be defined for a metadata field. (For a single metadata field, however, multiple CATEGORY rules that refer to different taxonomies are not supported.)
- ❖ Multiple search rules are run in the order specified, so that if a search rule does not result in a suggested value, the next rule is run. The list should be arranged from most to least specific.

- ❖ Search rule types can be mixed within a metadata field. For example, you can define an Option List rule, a Pattern Matching rule, and an Abstract rule for the same metadata field.
- ❖ If none of the search rules specified for a metadata field can be satisfied, the field is left blank.

PATTERN MATCHING SEARCH RULES

Pattern Matching search rules look for specific text or a specific XML element and return an associated value. For example, the *Invoice #* metadata field can be filled by the value that follows an *Invoice:* or *Invoice Number:* label in the source document, or it can be filled by the value that is within the *<Invoice>* tag in the XML document.

Rule Types

There are two general types of Pattern Matching rules: Tag Search and Text Search.

- ❖ [Tag Search](#) searches for an XML element that exactly matches the key. If such an element is found, the text contained in the element is returned as the result.
- ❖ [Text Search](#) searches for text that matches the key. If such text is found, the text near or following the key is returned as the result.



Note: Tag Searches are case sensitive. Text Searches are not case sensitive.

Sub-Types

Within each of the two general types of Pattern Matching search rules, there are several sub-types. These sub-types are explained in more detail in the Examples section below.

Tag Search

- ❖ TAG_TEXT
- ❖ TAG_ALLTEXT

Text Search

- ❖ TEXT_REMAINDER
- ❖ TEXT_ALLREMAINDER

- ❖ TEXT_FULL
- ❖ TEXT_ALLFULL
- ❖ TEXT_NEXT
- ❖ TEXT_ALLNEXT

Key

The key for a Pattern Matching search rule is either an XML element (for a Tag Search) or a text phrase (for a Text Search).

Count

The count for a Pattern Matching search rule defines the number of tags or text phrases that must be matched before the rule returns results. For example, a count of 4 will look for the fourth occurrence of the key. If only three occurrences of the key are found in the document, the rule fails.

The default count of 1 returns the first occurrence of the key.

Examples

The following examples illustrate the use of the Pattern Matching search rules.

TAG_TEXT

TAG_TEXT searches for an XML element name that matches the key exactly (including case). If such an element is found, all text that belongs to the element is concatenated and returned as the result.

Content	<pre><TAG_A>Title: The Big <TAG_B>Bad</TAG_B> Wolf</TAG_A> <TAG_C>Subtitle: A <TAG_D>Morality</TAG_D> Play</TAG_C></pre>
Rule	TAG_TEXT

Key	TAG_A
Returns	Title: The Big Wolf

TAG_ALLTEXT

TAG_ALLTEXT searches for an XML element name that matches the key exactly (including case). If such an element is found, all text that belongs to the element, and to all children of the element, is concatenated and returned as the result.

Content	<TAG_A>Title: The Big <TAG_B>Bad</TAG_B> Wolf</TAG_A> <TAG_C>Subtitle: A <TAG_D>Morality</TAG_D> Play</TAG_C>
Rule	TAG_ALLTEXT
Key	TAG_A
Returns	Title: The Big Bad Wolf

TEXT_REMAINDER

TEXT_REMAINDER searches for text that matches the key exactly (except for case). If such text is found, any text following the key that belongs to the same XML element is returned as the result.

Content	<TAG_A>Title: The Big <TAG_B>Bad</TAG_B> Wolf</TAG_A> <TAG_C>Subtitle: A <TAG_D>Morality</TAG_D> Play</TAG_C>
Rule	TEXT_REMAINDER
Key	Title:
Returns	The Big Wolf

TEXT_ALLREMAINDER

TEXT_ALLREMAINDER searches for text that matches the key exactly (except for case). If such text is found, any text following the key that belongs to the same XML element, and to all children of the element, is returned as the result.

Content	<TAG_A>Title: The Big <TAG_B>Bad</TAG_B> Wolf</TAG_A> <TAG_C>Subtitle: A <TAG_D>Morality</TAG_D> Play</TAG_C>
Rule	TEXT_ALLREMAINDER
Key	Title:
Returns	The Big Bad Wolf

TEXT_FULL

TEXT_FULL searches for text that matches the key exactly (except for case). If such text is found, any text that belongs to the same XML element, including the key text, is returned as the result.

Content	<TAG_A>Title: The Big <TAG_B>Bad</TAG_B> Wolf</TAG_A> <TAG_C>Subtitle: A <TAG_D>Morality</TAG_D> Play</TAG_C>
Rule	TEXT_FULL
Key	Title:
Returns	Title: The Big Wolf

TEXT_ALLFULL

TEXT_ALLFULL searches for text that matches the key exactly (except for case). If such text is found, any text that belongs to the same XML element, including the key text and any text belonging to children of the element, is returned as the result.

Content	<TAG_A>Title: The Big <TAG_B>Bad</TAG_B> Wolf</TAG_A> <TAG_C>Subtitle: A <TAG_D>Morality</TAG_D> Play</TAG_C>
Rule	TEXT_ALLFULL
Key	Title:
Returns	Title: The Big Bad Wolf

TEXT_NEXT

TEXT_NEXT searches for text that matches the key exactly (except for case). If such text is found, any text that belongs to the next non-blank XML element is returned as the result. Blank elements and elements composed of non-printing characters will not be selected as the return value.

Content	<TAG_A>Title: The Big <TAG_B>Bad</TAG_B> Wolf</TAG_A> <TAG_C>Subtitle: A <TAG_D>Morality</TAG_D> Play</TAG_C>
Rule	TEXT_NEXT
Key	Title:
Returns	Subtitle: A Play

TEXT_ALLNEXT

TEXT_ALLNEXT searches for text that matches the key exactly (except for case). If such text is found, any text that belongs to the next non-blank XML element, and to all children of the element, is returned as the result. Blank elements and elements composed of non-printing characters will not be selected as the return value.

Content	<TAG_A>Title: The Big <TAG_B>Bad</TAG_B> Wolf</TAG_A> <TAG_C>Subtitle: A <TAG_D>Morality</TAG_D> Play</TAG_C>
Rule	TEXT_ALLNEXT
Key	Title:
Returns	Subtitle: A Morality Play

ABSTRACT SEARCH RULES

Abstract search rules look for an XML element and return a descriptive sentence or paragraph from that element. For example, the *Summary* metadata field could be filled by a returned value of "Germany is a large country in size, culture, and worldwide economics. One of Germany's largest industries includes the manufacturing of world class automobiles like BMW, Mercedes, and Audi."

The Abstract rule type is useful where there is no readily identifiable or explicitly tagged block of text in the content item. Typically, these rules are used to suggest summary or topic information about the document.

Rule Types

There are two Abstract search rules: First Paragraph and First Sentence.

- ❖ **First Paragraph** searches for an XML element that exactly matches the key. The entire paragraph of the first such element that meets the size criteria (specified by the count) is returned as the result.
- ❖ **First Sentence** searches for an XML element that exactly matches the key. If such an element is found, the first sentence of the element is returned as the result.

Key

The key for an Abstract search rule is an XML element.

Count

The count is interpreted differently for the First Paragraph and First Sentence search rules.

First Paragraph

For a First Paragraph search rule, the count is a size threshold measured in percent:

1. The rule searches the document for all paragraphs that match the key.
2. The rule calculates the average size (based on character count) of the paragraphs that match the key.
3. The rule multiplies the average size by the count percentage (0 = 0%, 100 = 100%).
4. The rule looks for the first paragraph larger than the resulting number.

For example, if the count is set to 75 and the average paragraph size is 100 characters, the rule returns the first paragraph larger than 75 characters that matches the key.

If the count is set to the default of 1, the rule is likely to return the first paragraph that matches the key.

First Sentence

For a First Sentence search rule, the count is the number of elements that have their first sentences returned.

For example, if the count is set to 3, the rule returns the first sentence from each of the first three elements that match the key.

Examples

The following examples illustrate the use of the Abstract search rules.

FIRST_PARAGRAPH

This example returns the first <Text> element that exceeds one-half the average <Text> element paragraph size. Note that the <Title> element does not match the key value, so it is ignored for both the search and for the average length calculation.

Content	<Title>Poem</Title> <Text>Mary had</Text> <Text>a little Lamb</Text> <Text>The fleece was white as snow</Text> <Text>And everywhere that Mary went the lamb was sure to go</Text>
Rule	FIRST_PARAGRAPH
Key	Text
Count	50
Returns	The fleece was white as snow.

FIRST_SENTENCE

This example returns the first sentence of the first two <Text> elements. Note that the <Title> element does not match the key value, so it is excluded from the search.

Content	<Title>Barefoot in the Park</Title> <Text>See Dick run. See Jane run. See Dick and Jane.</Text> <Text>See Spot run. See Puff chase Spot.</Text> <Text>See Dick chase Spot and Puff.</Text>
Rule	FIRST_SENTENCE
Key	Text
Count	2
Returns	See Dick run. See Spot run.

OPTION LIST SEARCH RULE

The Option List search rule, named `OPTION_LIST`, looks for keywords within the source document, applies a score for each keyword found, and returns the option list value that has the highest keyword score. For example, if the keywords “*margin*,” “*SEC filing*,” or “*invoice*” were found in a document, the suggested value for the *Department* field would be *Accounting*, while the keywords *tolerance*, *assembly*, or *inventory* would return *Manufacturing* as the suggested value.

- ❖ The Option List search rule will usually be applied to metadata fields that have an option list defined in the Configuration Manager. See the Content Server online help for information on creating option lists for custom metadata fields.
- ❖ Option list names and values (called **categories** in Content Categorizer) appear in Content Categorizer as specified in the Configuration Manager. If you create or change a custom option list field while the CC Admin Applet is open, you will need to close and reopen the applet to see the changes.
- ❖ The current version of Content Server automatically inserts a blank value as the default value in a custom option list field. In this case, the first value (by default, a blank value) will not be considered a user-entered value, and the Option List search rule will be applied. If you do not want the Option List search rule to override the first value in a custom option list field, you must provide a default value for that option list on the Configuration Manager Applet.

Rule Types

There is one type of Option List search rule, which searches for keywords (single words or phrases) that exactly match the keywords defined in the key.

- ❖ Keywords may be single words (for example, *dog*) or multiple-word phrases (for example, *black dog*).
- ❖ Keywords may use the following defined set of operators to further refine a search:
 - `$$AND$$`
 - `$$OR$$`
 - `$$AND_NOT$$`
 - `$$NEAR$$`
- ❖ Keywords are pre-assigned to each category (value) in the option list, and each keyword has a weight assigned to it. See [Defining Option List Keywords](#) (page 5-19).

- ❖ The number of occurrences of each keyword found in the document is multiplied by its weight, resulting in a keyword score.
- ❖ The keyword scores for each category are added together, resulting in a category score.
- ❖ The category with the highest score is returned as the suggested value.
- ❖ If there is a tie between categories, the category earliest in the option list is returned as the suggested value.
- ❖ The weights *Always* and *Never* can be used to override the scores and count threshold.
 - An occurrence of a keyword with the *Always* weight forces the category to be returned as the suggested value, regardless of score.
 - An occurrence of a keyword with the *Never* weight disqualifies the category from being returned as the suggested value, regardless of score.
 - If two categories have keywords that are assigned the *Always* weight, and both keywords occur in the document, the keyword first found in the document takes precedence.



Important: Option List searches are case sensitive and must match exactly. For example, *Invoice*, *Invoices*, *invoice*, and *invoices* must be defined to retrieve all instances of this keyword.

Key

The key for an Option List search rule is the Option List name, as shown on the Option Lists tab of the CC Admin Applet.

Count

The count for an Option List search rule sets a minimum threshold score for the rule to return results. For example, if the count is set to 50, and the highest accumulated keyword score is 45, the rule fails.

Examples

In this example, the score for *Dick* and *Spot* is 30 (3 occurrences x 10), and the score for *Jane* and *Puff* is 20 (2 occurrences x 10). *Dick* is returned as the suggested value because it is earlier in the option list than *Spot*:

Content	<Title>Barefoot in the Park</Title> <Text>See Dick run. See Jane run. See Dick and Jane.</Text> <Text>See Spot run. See Puff chase Spot.</Text> <Text>See Dick chase Spot and Puff.</Text>
Rule	OPTION_LIST
Key	MainCharacter
Count	10
Option List Categories, Keywords, and Weights	Dick: Dick=10, boy=5, Richard=2 Jane: Jane=10, girl=5, Janie=2 Spot: Spot=10, dog=5 Puff: Puff=10, cat=5
Returns	Dick

In this example, *Spot* is returned as the suggested value because its score of 60 (3 occurrences x 20) is higher than the other categories:

Content	<Title>Barefoot in the Park</Title> <Text>See Dick run. See Jane run. See Dick and Jane.</Text> <Text>See Spot run. See Puff chase Spot.</Text> <Text>See Dick chase Spot and Puff.</Text>
Rule	OPTION_LIST
Key	MainCharacter
Count	10

Option List Categories, Keywords, and Weights	Dick: Dick=10, boy=5, Richard=2 Jane: Jane=10, girl=5, Janie=2 Spot: Spot=20, dog=10 Puff: Puff=10, cat=5
Returns	Spot

In this example, the rule fails because none of the scores is above the Count threshold of 50:

Content	<Title>Barefoot in the Park</Title> <Text>See Dick run. See Jane run. See Dick and Jane.</Text> <Text>See Spot run. See Puff chase Spot.</Text> <Text>See Dick chase Spot and Puff.</Text>
Rule	OPTION_LIST
Key	MainCharacter
Count	50
Option List Categories, Keywords, and Weights	Dick: Dick=10, boy=5, Richard=2 Jane: Jane=10, girl=5, Janie=2 Spot: Spot=10, dog=5 Puff: Puff=10, cat=5
Returns	Fail

In this example, *Puff* is returned as the suggested value because the keyword "Puff" has a weight of Always:

Content	<Title>Barefoot in the Park</Title> <Text>See Dick run. See Jane run. See Dick and Jane.</Text> <Text>See Spot run. See Puff chase Spot.</Text> <Text>See Dick chase Spot and Puff.</Text>
----------------	---

Rule	OPTION_LIST
Key	MainCharacter
Count	10
Option List Categories, Keywords, and Weights	Dick: Dick=10, boy=5, Richard=2 Jane: Jane=10, girl=5, Janie=2 Spot: Spot=10, dog=5 Puff: Puff=Always, cat=5
Returns	Puff

CATEGORIZATION ENGINE SEARCH RULE

The Categorization Engine search rule, named CATEGORY, uses a 3rd-party categorizer engine and defined taxonomy to determine and return a value that represents a category within the specified taxonomy, for example, News/Technology/Computers.

Rule Types

There is one type of Categorization Engine search rule, which uses the categorizer engine and taxonomy specified in the Key to return a value for the field.

Key

The key for a Categorization Engine search rule is the name of the categorizer engine followed by the name of the taxonomy. For example, *EngineName/TaxonomyName*.



Note: If you do not specify an engine name in the Key field, Content Categorizer defaults to the first engine displayed in the Categorizer Engines list. Therefore, if you have defined only one engine, you would only need to enter the taxonomy name in the Key field.

Count

The count for a Categorization Engine search rule sets a minimum confidence level threshold for the returned results.

When a categorization engine returns a category (or set of categories) for a given query, a confidence level is also returned, which is often expressed as a percentage for each category. The Category rule always accepts the highest-confidence category, unless the confidence level is below the count value specified for the rule, in which case the rule fails. For example, if the count is set to 50, and the highest-confidence category returned is 45, the rule fails.

The default count of 1 would always accept the highest-confidence category returned by the categorizer engine.



Note: The actual range for the Count value depends on the categorizer engine that is being used.

FILETYPE SEARCH RULE

The Filetype search rule, named FILETYPE, looks at the filename extension of a document and returns a term, usually a file type description associated with the filename extension.

Rule Types

There is one type of Filetype search rule, which uses the filename extension of the primary (native) file to return a value for the field.

When the Filetype search rule is defined for a metadata field, the filename extension of the content item is matched against all values in the Content Server's DocFormatsWizard table. This table is found in the file *doc_config.htm*, which is located in the `<install_dir_path>/shared/config/resources/` directory.

If a match is found, the associated value in the Description column is extracted and translated. The resulting string is returned as the suggested metadata value for the field.



Note: If the primary file path has no extension, or if the extension does not match any of the "extensions" values in the DocFormatsWizard table, the rule fails and the next rule in the list for the metadata field is executed.

Key

The key for a FILETYPE search rule is not used when determining a metadata value. **The Key field should be left blank.**

Count

The count for a FILETYPE search rule is not used when determining a metadata value.

The Count field should be left blank.



Note: If a FILETYPE rule is created with non-blank Key or Count fields, a warning message is displayed indicating that these fields are not supported by the rule.

Examples

	Example 1	Example 2
Primary File	policies.doc	procedures.wpd
Rule	FILETYPE	FILETYPE
Key	blank	blank
Count	blank	blank
Returns	Microsoft Word Document	Corel WordPerfect Document

DEFINING SEARCH RULES

This topic includes the following tasks:

- ❖ [Defining Search Rules](#) (page 5-18)
- ❖ [Defining Option List Keywords](#) (page 5-19)
- ❖ [Applying Rules to the Type Field](#) (page 5-20)

Defining Search Rules



Important: During Content Server startup, Content Categorizer takes a snapshot of the current metadata field configuration including field names and lengths. If your metadata field configuration changes, you must restart Content Server before running the Content Categorizer Admin Applet to add or modify any search rules.

To define search rules for any metadata field:

1. Log into the Content Server as the system administrator.
2. Click the **Administration** link.
3. Click the **Content Categorizer Administration** link (under Administration Pages for *instance_name*).

The CC Admin Applet screen is displayed.

4. Click the Rule Sets tab.
5. Click on the Ruleset drop-down list and select the desired ruleset, or click Add to add a new ruleset.
6. Select a metadata field from the **Field** choice list.
7. Click **Add**.

The Rules for *field* screen is displayed.

8. Select the rule type from the **Rule** choice list.
9. Enter the search rule key in the **Key** field.



Note: For an OPTION_LIST search rule, keywords for the option list must be defined on the Option List tab. See [Defining Option List Keywords](#) (page 5-19).

10. Enter the count in the **Count** field.
11. Click **OK**.
12. Add search rules to each metadata field as desired.
 - ❖ To delete a rule, select the rule in the **Rules List** and click **Delete**.
 - ❖ To edit a rule, select the rule in the **Rules List** and click **Edit**.
 - ❖ To adjust the order of the rules, select the rule in the **Rules List** and click **Move Up** or **Move Down**. Rules are applied in the order listed. If the first rule succeeds, no other rules are applied. If the first rule fails, then the next rule is applied, and so forth.



Important: If you have added, edited, or deleted a CATEGORY rule, a dialog will prompt you to apply the changes and build, rebuild, or check for orphaned query trees for this rule on the Query Trees tab.

13. Click **Apply** to save the changes, or click **OK** to save the changes and close the CC Admin Applet screen.

Defining Option List Keywords

To define the keywords and weights for an option list:

1. Log into the Content Server as the system administrator.
2. Click the **Administration** link.
3. Click the **Content Categorizer Administration** link (under Administration Pages for *instance_name*).

The CC Admin Applet screen is displayed.

4. Click the Option Lists tab.
5. Select an option list from the **Option List** choice list.



Caution: When an option list metadata field is deleted from the Configuration Manager, the field is removed from the Rule Sets tab, but it still appears in the **Option List** choice list on the Option Lists tab. Be careful not to select an obsolete option list.

6. Select a value from the **Category** choice list.
7. Enter a keyword or phrase in the **Keyword** field.
 - ❖ Keywords may be single words or multiple-word phrases.
 - ❖ Keywords may include Boolean-type expressions, where the following set of binary operators are valid: `$$AND$$`, `$$OR$$`, `$$AND_NOT$$`, `$$NEAR$$`



Important: Option List searches are case sensitive and must match exactly.

8. Select a weight for the keyword.
 - ❖ **Always** = If the keyword is found, the selected category will be returned as the suggested value, regardless of the score.
 - ❖ **Weight** = This number multiplied by the number of occurrences of the keyword is the category's score. The category with the highest score is returned as the suggested value for the option list field.

- ❖ **Never** = If the keyword is found, the selected category will not be returned as the suggested value, regardless of the score.
9. Click **Add**.
 10. Enter keywords for each category in the selected option list.
 - ❖ To delete a keyword, select the keyword in the **Keywords** list and click **Delete**.
 - ❖ To edit a keyword, select the keyword in the **Keywords** list, click **Edit**, edit the keyword and/or the weight, and click **Update**.
 11. Click **Apply** to save the changes, or click **OK** to save the changes and close the CC Admin Applet screen.

Applying Rules to the Type Field

You can edit the content server configuration file so that Content Categorizer ignores the **Type** default value and applies search rules to the **Type** field.



Note: This procedure applies only to the **Type** (dDocType) field. Search rules cannot be applied to the other standard option list fields (Security Group, Author, and Account).

To apply search rules to the **Type** field:

1. Open the *config.cfg* file located in the `<cs_install_dir_path>/config/` directory in a text-only editor such as WordPad.
2. Add the following line to the file:
`ForceDocTypeChoice=true`
3. Save and close the file.
4. Stop and restart the Content Server:
 - a. Log into the Content Server as the system administrator.
 - b. Click the **Administration** link.
 - c. Click the **Admin Server** link.
 - d. On the Content Admin Server, click the *instance_name* button.
 - e. In the sidebar, click the **Component Manager** link.
 - f. Click the **Start/Stop Content Server** link in the sidebar.
 - g. Click the **Restart** icon.

USING CATEGORIZER ENGINES

OVERVIEW

This section includes the following topics:

- ❖ [Categorizer Engine Registration](#) (page 6-1)
- ❖ [Building Query Trees](#) (page 6-2)
- ❖ [Hierarchical Browsing](#) (page 6-4)

CATEGORIZER ENGINE REGISTRATION

If you want to define a CATEGORY rule to determine metadata field values, you must install and set up one or more categorization engines. Then, you must register each engine in Content Categorizer.



Note: Adaptor modules for Autonomy's *Categorizer* engine and APR Smartlogik's *Structure* engine have been developed and are available.

To register a categorizer engine in Content Categorizer:

1. Log into the Content Server as the system administrator.
2. Click the **Administration** link.

3. Click the **Content Categorizer Administration** link (under Administration Pages for *instance_name*).

The CC Admin Applet screen is displayed.

4. On the Categorizer Engines tab, click **Add**.

The Add Categorizer Engine screen is displayed.

5. Enter the following:

- ❖ Engine Name
- ❖ Indexer Class
- ❖ Indexer Configuration
- ❖ Extractor Class
- ❖ Extractor Configuration
- ❖ Description



Important: To register an engine, valid entries for all fields (except Description) are required.

6. Click **OK**.
7. Click **Apply** to save the changes, or click **OK** to save the changes and close the CC Admin Applet.

BUILDING QUERY TREES

If you use the CATEGORY rule to determine metadata values that conform to a taxonomy created by a categorization engine, the CC Admin Applet enables you to create a query tree to provide a browsable hierarchy of categories (and sub-categories, if they exist) based on the taxonomy structure.



Note: The CATEGORY rule associates the name of a metadata field with the name of a taxonomy that is owned by a categorization engine.

The basic steps to build a query tree include:

1. Registering a categorizer engine in Content Categorizer. See [Categorizer Engine Registration](#) (page 6-1).

2. Defining a CATEGORY search rule for one or more metadata fields. See [Categorization Engine Search Rule](#) (page 5-15).
3. Building a query tree based on the taxonomy selected for the CATEGORY rule. See the procedure that follows.

To build a query tree in Content Categorizer:

1. Log into the Content Server as the system administrator.
2. Click the **Administration** link.
3. Click the **Content Categorizer** link (under Administration Pages for *instance_name*). The CC Admin Applet screen is displayed.
4. On the Query Trees tab, select the specific query tree to be built.



Note: A new, yet-to-be-built query tree will not display a value in the Taxonomy Title column, and will display **YES** in the Needs Rebuild? column.

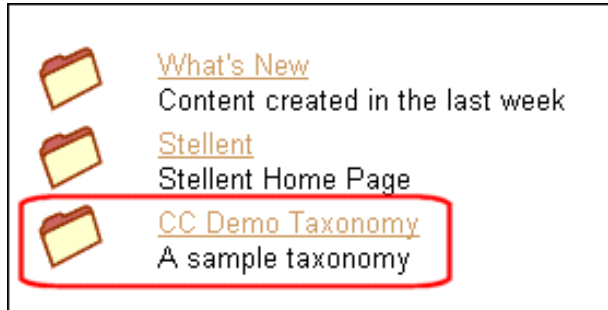


Note: The Query Trees list displays queries only if a CATEGORY rule has been defined for a metadata field (that is, the Query Trees list displays a row for each CATEGORY rule that has been defined).

5. Click **Build Query Tree**.
The Build Query Tree screen is displayed.
6. Enter the following:
 - ❖ Taxonomy Title [a required field; this label is displayed as the top-level link on the Library page]
 - ❖ Taxonomy Description [an optional field; this label is displayed under the top-level link on the Library page]
7. Select Build Taxonomy Cache check box.
8. Click **OK**.
9. To view Query Tree in Content Server, click the **Library** link.
The Query Tree link is displayed on the Library page.

HIERARCHICAL BROWSING

After building a query tree in the CC Admin Applet, a Taxonomy Title link is added to the Content Server Library page.



Browsing the Taxonomy

Clicking the Taxonomy Title link displays the categories defined in the taxonomy. (In this example, only one top-level category NYTimes is defined in the CC_DemoCat taxonomy.) Note that there are no content items in the root taxonomy.



Browsing a Category in the Taxonomy

Clicking the NYTimes category link enables you to browse its subcategories and content items. Note that in the following example there are 10 sub-categories but no content items in the main NYTimes category.

[Library](#) > [CC_DemoCat](#) > **NYTimes**

Sub-categories: (10)

- [Business](#)
- [Education](#)
- [Health](#)
- [International](#)
- [National](#)
- [NewYorkRegion](#)
- [Politics](#)
- [Science](#)
- [Sports](#)
- [Technology](#)

Content Items: (0)

Browsing a Sub-Category

Clicking a sub-category link enables you to browse its sub-categories and content items. Note that in the following example there are five sub-categories and 29 content items returned from the search query.

[Library](#) > [CC_DemoCat](#) > [NYTimes](#) > [International](#)






Sub-categories: (5)

- [Africa](#)
- [Americas](#)
- [AsiaPacific](#)
- [Europe](#)
- [MiddleEast](#)

Content Items: (29)

List Navigation: 1 2 [Next]

Displaying matches 1-25.

	Description	Rev.	Info
	CC_DB_737 Allied Special Forces Scouting Afghanistan, Official Says Category: (CC_DemoCat) > NYTimes > International	1	
	CC_DB_738 Afghan Plant Has Potential Worrying Bush Category: (CC_DemoCat) > NYTimes > International	1	
	CC_DB_739		



Important: To display the CATEGORY metadata field name and the taxonomy path for the content object, you must set your Search Results to “Classic View”. The Thumbnail and Headline views are too compact for this information to display properly.



ADAPTOR MODULES

ADAPTOR MODULES FOR AUTONOMY AND APR SMARTLOGIK

Adaptor modules for Autonomy's *Categorizer* engine and APR Smartlogik's *Structure* engine have been developed. To register either of these engines in Content Categorizer, refer to the information in the following tables and to the API documentation provided by the categorization engine vendor.



Note: The indexer class files and the extractor class files referenced in the tables are located in the `<install_dir_path>/classes/contentcategorizer/` directory.

<i>Autonomy Categorizer</i>	
Engine Name	A unique identifier, for example: Autonomy
Indexer Class	CC.SccRuleAutonomyCategorizer
Indexer Configuration	A comma-separated list of engine-specific initialization parameters, which is passed to the Indexer Class's <code>setup()</code> method. For example: [HOST IP],4000,4001,[HOST IP],4002,4003
Extractor Class	CC.SccTaxonomyExtractorAutonomy

Autonomy Categorizer	
Extractor Configuration	A comma-separated list of engine-specific initialization parameters, which is passed to the Extractor Class's doExtract() method. For example: [HOST IP],4000,4001,[HOST IP],4002,4003
Description	A plain text description of the categorization engine.



Note: For more information refer to your Autonomy documentation.

APR Smartlogik Structure	
Engine Name	A unique identifier, for example: Structure
Indexer Class	CC.SccRuleStructureCategorizer
Indexer Configuration	A comma-separated list of engine-specific initialization parameters, which is passed to the Indexer Class's setup() method. For example: [HOSTNAME],InfoSort,1050
Extractor Class	CC.SccTaxonomyExtractorStructure
Extractor Configuration	A comma-separated list of engine-specific initialization parameters, which is passed to the Extractor Class's doExtract() method. For example: [HOSTNAME],InfoSort,1050
Description	A plain text description of the categorization engine.



Note: For more information refer to your APR Smartlogik documentation.

B

SAMPLE *doc_config.htm* PAGE

<@table DocFormatsWizard@>

dFormat	extensions	dConversion	dDescription
application/corel-wordperfect, application/wordperfect	wpd	WordPerfect	apWordPerfectDesc
application/vnd.framemaker	fm	FrameMaker	apFramemakerDesc
application/vnd.framebook	bk, book	FrameMaker	apFrameMakerDesc
application/vnd.mif	mif	FrameMaker	apFrameMakerInterchangeDesc
application/lotus-1-2-3	123, wk3, wk4	123	apLotus123Desc
application/lotus-freelance	prz	Freelance	apLotusFreelanceDesc
application/lotus-wordpro	lwp	WordPro	apLotusWordProDesc
application/msword, application/ms-word	doc, dot	Word	apMicrosoftWordDesc
application/vnd.ms-excel, application/ms-excel	xls	Excel	apMicrosoftExcelDesc
application/vnd.ms-powerpoint, application/ms-powerpoint	ppt	PowerPoint	apMicrosoftPowerPointDesc

application/vnd.ms-project, application/ms-project	mpp	MSPProject	apMicrosoftProjectDesc
application/ms-publisher	pub	MSPub	apMicrosoftPublisherDesc
application/write	wri	Word	apMicrosoftWriteDesc
application/rtf	rtf	Word	apRtfDesc
application/vnd.visio	vsd	Visio	apVisioDesc
application/vnd.illustrator	ai	Illustrator	apIllustratorDesc
application/vnd.photoshop	psd	PhotoShop	apPhotoshopDesc
application/vnd.pagemaker	p65	PageMaker	apPageMakerDesc
image/gif	drw, igx, flo, abc, igt	iGrafx	apiGrafxDesc
text/postscript	ps	Distiller	apDistillerDesc
application/hangul	hwp	Hangul97	apHangul97Desc
application/ichitaro	jtd, jtt	Ichitaro	apIchitaroDesc
image/graphic	gif, jpeg, jpg, png, bmp, tiff, tif	ImageThumbnail	apThumbnailsDesc
image/application	txt, eml, msg	NativeThumbnail	apNativeThumbnailsDesc

<@end@>

<@table PdfConversions@>

dFormat	extensions	dConversion	dDescription
application/pdf	pdf	PDFOptimization	apPdfOptimization
application/pdf	pdf	ImageThumbnail	apPdfThumbnailsDesc

<@end@>



CONFIGURATION VARIABLES

OVERVIEW

This chapter covers the following topic:

- ❖ [MaxQueryRows](#) (page C-1)

MAXQUERYROWS

The MaxQueryRows is used in Content Categorizer to specify the maximum number of documents that can be included in a single batch load process. The default setting for this configuration variable is 200 but can be decreased or increased as necessary. Increasing the value will slow the response time for loading a large list of documents. Although the impact of setting MaxQueryRows to something in the range of 1000 to 2000 is minor, setting it in the area of 100,000 would probably produce an unacceptable performance level.

Format

MaxQueryRows=2000

XSLT TRANSFORMATION

OVERVIEW

Content Server uses a two-step process for categorizing content. The first step translates content into an XML format, the second step transforms the XML file into another XML file useful to Content Categorizer. The process is transparent in that the original content is not modified, and both the translated and transformed XML files are discarded after use.

The translation step uses the OutsideIn XML Export filters to output the XML in either SearchML or Flexiondoc XML format, depending on the type of content being translated and whether the format is available for the platform being used. This translation process enables Categorizer to support a large number of different source document formats.

The transformation step uses eXtensible Stylesheet Language Transformations (XSLT) to transform the initial XML output into an XML equivalent that can be easily searched and analyzed by Content Categorizer, based on search rules defined by the user.

An overview of the transformation process may be useful to anyone interested in the categorization process, and serve as a starting point for users who would like to define their own XSLT stylesheets to accommodate their specific document processing needs.

This section covers the following topics:

- ❖ [Translation Using OutsideIn XML Export Filters](#) (page D-2)
- ❖ [Transformation Using XSLT Stylesheets](#) (page D-2)
- ❖ [SearchML Transformation](#) (page D-3)
- ❖ [Flexiondoc Transformation](#) (page D-4)

Translation Using OutsideIn XML Export Filters

A runtime version of the OutsideIn XML Export product is integrated and installed with Content Categorizer, and it filters content checked in for categorization. The Export filters convert content to XML for transformation using Categorizer's XSLT stylesheets. The transformation is necessary because the Export XML schemas, Flexiondoc and SearchML, are not in a form easily searched by Content Categorizer rules.

Transformation Using XSLT Stylesheets

Two stylesheets are included with Content Categorizer and applied based on the initial translation format provided by the OutsideIn XML Export filter. The stylesheets are located in the following directory.

```
/<install_dir>/<instance_dir>/custom/ContentCategorizer/stylesheets/
```

For content items output in SearchML, *searchml_to_scc.xsl* is applied. For content items output in Flexiondoc, *flexiondoc_to_scc.xsl* is applied. SearchML and Flexiondoc both reproduce style designations found in the source content, but they do so differently, in ways not detectable by Content Categorizer rules. The appropriate stylesheet can recognize the necessary style information in each format and use that information as the basis for transforming the final output tags into an XML document useful to Content Categorizer.

The similarity between SearchML and Flexiondoc depends on the degree to which internal styles or metadata are used in the content. When working with content using named styles, such as Microsoft Word, the resultant output will be similar. When working with content in formats such a PDF or text, results come out with more generic tagging.



Important: There is a problem with the XSLT transformation used to post-process PDF content that is output in Flexiondoc format. When Flexiondoc is used, single words are assigned to individual XML elements, making the final XML unsuitable for most Categorizer search rules. It is therefore recommended that you use SearchML for categorizing PDF content.

SearchML Transformation

When the OutsideIn XML Export filter translates content into SearchML XML format, it identifies the properties of the content item, such as title, subject, and author, and tags them as a `<doc_property>` element. It distinguishes the properties by a “type” attribute. It also identifies document text and tags it as a `<p>` element. It distinguishes styles within text by an “s” attribute.

Document Properties and Text Style Examples

For example, using the *Wellington_WordStyle.doc* example found in the `/<install_dir>/<instance_dir>/custom/ContentCategorizer/CC_Sample/` directory, the file’s author property, “Duke of Wellington,” is tagged in the SearchML XML output as:

```
<doc_property type="author">Duke of Wellington</doc_property>
```

The first paragraph of the item, listing the date, would be tagged as:

```
<p>Date: August 24, 1812</p>
```

Note that no style attribute is defined.

Applying the *searchml_to_scc.xsl* stylesheet to the translated XML file searches the XML for all `<doc_property>` tags and uses the “type” attribute as the suffix for the transformed output tag used as a key in a Content Categorizer rule.

For example, the following code in the *searchml_to_scc.xsl* stylesheet would take the tag, `<doc_property type="author">Duke of Wellington</doc_property>` and output `<scc_author>Duke of Wellington</scc_author>`:

```
<xsl:template match="sml:doc_property[@type]">
  <xsl:variable name="typeValue">
    <xsl:value-of select="@type"/>
  </xsl:variable>
  <xsl:element name="scc_{translate($typeValue, $translateFrom,
    $translateTo)}">
    <xsl:value-of select="."/>
  </xsl:element>
</xsl:template>
```

Example D-1 *searchml_to_scc.xsl* stylesheet template element to transform property tags.

Similarly, the *searchml_to_scc.xsl* stylesheet also causes the XML file to be searched for all `<p>` tags and uses the “s” attribute as the suffix for the transformed output tag used as a

key in a Content Categorizer rule. Where no style attribute is defined, the transformation passes the <p> tag through.

Flexiondoc Transformation

When the OutsideIn XML Export filter translates content into Flexiondoc XML format, it identifies the properties of the content item, such as title, subject, and author, and tags them as a <doc_property> element, just like SearchML. However, it distinguishes the properties by a “name” attribute, instead of “type.”

Document Properties Example

For example, using the *Wellington_WordStyle.doc* example found in the `/<install_dir>/<instance_dir>/custom/ContentCategorizer/CC_Sample/` directory, the file’s author property, “Duke of Wellington,” is tagged in the Flexiondoc XML output as:

```
<doc_property name="author">Duke of Wellington</doc_property>
```

Applying the *flexiondoc_to_scc.xsl* stylesheet to the translated XML file searches the XML for all <doc_property> tags and uses the “name” attribute as the suffix for the transformed output tag used as a key in a Content Categorizer rule.

For example, the following code in the *flexiondoc_to_scc.xsl* stylesheet would take the tag,

```
<doc_property name="author">Duke of Wellington</doc_property>
and output <scc_author>Duke of Wellington</scc_author>:
```



```

<xsl:template match="fld:doc_property">
  <xsl:variable name="propName">
    <xsl:choose>
      <xsl:when test="@name">
        <xsl:value-of select="@name" />
      </xsl:when>
      <xsl:when test="@user_defined_name">
        <xsl:value-of select="@user_defined_name" />
      </xsl:when>
      <xsl:otherwise>NAMELESS_DOC_PROPERTY_WITH_ID_<xsl:value-of
        select="@id" /></xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:element name="scc_{translate($propName, $translateFrom, $translateTo)}">
    <xsl:value-of select="." />
  </xsl:element>
</xsl:template>

```

Example D-2 *flexiondoc_to_scc.xsl* stylesheet template element to transform property tags.

Text Style Example

Where Flexiondoc differs from SearchML is in how it identifies styles. Paragraph styles are tagged with <tx.p> tags, and character styles are tagged with <tx.r> tags, but each have an attribute based on a unique style “id,” in addition to a “name” attribute.

All styles are defined in child elements of the <style_tables> element of the Flexiondoc XML file, and given an “id” attribute, which is called when referencing the style, and which the template file uses to define a style key with a “name” attribute.

For example, in the Flexiondoc XML output of the *Wellington_WordStyle.doc* example, character styles are defined in the `<tx.char_style_table>` child of the `<style_tables>` parent element. Notice the “id” attribute:

```
<tx.char_style_table>
  <tx.char_style id="ID16d" auto_kern_above="0.1111in" auto_kerning="false"
    back_brush="ID168" font="ID16f" kerning="0in" text_brush="ID16e"
    text_effect="normal" text_hidden="false" text_position="normal"
    text_protected="false" text_strikethrough="none" underline="ID170"/>
  <tx.char_style id="ID178" back_brush="ID176" font="ID177"
    text_brush="ID176"/><tx.char_style id="ID187" font="ID186"/><tx.char_style
    id="ID1d6" font="ID1d5"/>
  <tx.char_style id="ID1e5" font="ID1e4"/>
  <tx.char_style id="ID1e8" name="Default Paragraph Font"
    predefined="default"/>
  <tx.char_style id="ID1ec" font="ID1eb"/>
</tx.char_style_table>
```

Example D-3 Character styles as tagged in the Flexiondoc XML translation of *Wellington_WordStyle.doc*.

When the *flexiondoc_to_scc.xsl* stylesheet is applied, it causes the output XML file to be searched for all character styles, <tx.char_style>. It uses the “id” attribute of the style to define unique <xsl:key> elements with a “name” attribute based on the “id” of each <tx.char_style> tag:

```
<xsl:key name="charStyleKey" match="fld:tx.char_style" use="@id" />
<xsl:template match="fld:tx.r[@style]">
  <xsl:variable name="charStyleName">
    <xsl:value-of select="key('charStyleKey', @style)/@name" />
  </xsl:variable>
  <xsl:choose>
    <xsl:when test="string-length($charStyleName) > 0">
      <xsl:element name="scc_{translate($charStyleName, $translateFrom,
        $translateTo)}">
        <xsl:apply-templates />
      </xsl:element>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="." />
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

Example D-4 Character style transformation code in the *flexiondoc_to_scc.xsl* file.

Similarly, when the stylesheet is applied, it causes the output XML file to be searched for all paragraph styles, <tx.para_style>. It then uses the “id” attribute of the style to define unique <xsl:key> elements with a “name” attribute based on the “id” of each <tx.para_style> tag:

```
<xsl:key name="paraStyleKey" match="fld:tx.para_style" use="@id" />
<xsl:template match="fld:tx.p[@style]">
  <xsl:variable name="styleValue">
    <xsl:value-of select="@style" />
  </xsl:variable>
  <xsl:variable name="paraStyleName">
    <xsl:value-of select="key('paraStyleKey', $styleValue)/@name" />
  </xsl:variable>
  <xsl:choose>
    <xsl:when test="string-length($paraStyleName) &gt; 0">
      <xsl:element name="scc_{translate($paraStyleName, $translateFrom,
        $translateTo)}">
        <xsl:apply-templates />
      </xsl:element>
    </xsl:when>
    <xsl:otherwise>
      <xsl:element name="p" >
        <xsl:value-of select="." />
      </xsl:element>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

Example D-5 Paragraph style transformation code in the *flexiondoc_to_scc.xsl* file.

Example Files

For more detailed study of examples, sample files are located here:

<install_dir>/<instance_dir>/custom/ContentCategorizer/CC_Sample

For more detailed study of the XSLT style sheets, they are located here:

<install_dir>/<instance_dir>/custom/ContentCategorizer/stylesheets



Caution: The stylesheets located in the directory listed above are used by Content Categorizer. Make duplicates for study.

THIRD PARTY LICENSES

OVERVIEW

This appendix includes a description of the Third Party Licenses for all the third party products included with this product.

- ❖ [Apache Software License](#) (page E-1)
- ❖ [W3C® Software Notice and License](#) (page E-2)
- ❖ [Zlib License](#) (page E-4)
- ❖ [General BSD License](#) (page E-5)
- ❖ [General MIT License](#) (page E-5)
- ❖ [Unicode License](#) (page E-6)
- ❖ [Miscellaneous Attributions](#) (page E-7)

APACHE SOFTWARE LICENSE

- * Copyright 1999-2004 The Apache Software Foundation.
- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *

Third Party Licenses

- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.

W3C® SOFTWARE NOTICE AND LICENSE

- * Copyright © 1994-2000 World Wide Web Consortium,
- * (Massachusetts Institute of Technology, Institut National de
- * Recherche en Informatique et en Automatique, Keio University).
- * All Rights Reserved. <http://www.w3.org/Consortium/Legal/>
- *
- * This W3C work (including software, documents, or other related items) is
- * being provided by the copyright holders under the following license. By
- * obtaining, using and/or copying this work, you (the licensee) agree that
- * you have read, understood, and will comply with the following terms and
- * conditions:
- *
- * Permission to use, copy, modify, and distribute this software and its
- * documentation, with or without modification, for any purpose and without
- * fee or royalty is hereby granted, provided that you include the following
- * on ALL copies of the software and documentation or portions thereof,
- * including modifications, that you make:
- *
- * 1. The full text of this NOTICE in a location viewable to users of the
- * redistributed or derivative work.
- *
- * 2. Any pre-existing intellectual property disclaimers, notices, or terms

* and conditions. If none exist, a short notice of the following form
* (hypertext is preferred, text is permitted) should be used within the
* body of any redistributed or derivative code: "Copyright ©
* [\$date-of-software] World Wide Web Consortium, (Massachusetts
* Institute of Technology, Institut National de Recherche en
* Informatique et en Automatique, Keio University). All Rights
* Reserved. <http://www.w3.org/Consortium/Legal/>"
*
* 3. Notice of any changes or modifications to the W3C files, including the
* date changes were made. (We recommend you provide URIs to the location
* from which the code is derived.)
*
* THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS
* MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT
* NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR
* PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE
* ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.
*
* COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR
* CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR
* DOCUMENTATION.
*
* The name and trademarks of copyright holders may NOT be used in advertising
* or publicity pertaining to the software without specific, written prior
* permission. Title to copyright in this software and any associated
* documentation will at all times remain with copyright holders.
*

ZLIB LICENSE

* zlib.h -- interface of the 'zlib' general purpose compression library
version 1.2.3, July 18th, 2005

Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly jloup@gzip.org

Mark Adler madler@alumni.caltech.edu

GENERAL BSD LICENSE

Copyright (c) 1998, Regents of the University of California

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

"Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

"Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

"Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GENERAL MIT LICENSE

Copyright (c) 1998, Regents of the Massachusetts Institute of Technology

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

UNICODE LICENSE

UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

Unicode Data Files include all data files under the directories <http://www.unicode.org/Public/>, <http://www.unicode.org/reports/>, and <http://www.unicode.org/cldr/data/> . Unicode Software includes any source code published in the Unicode Standard or under the directories <http://www.unicode.org/Public/>, <http://www.unicode.org/reports/>, and <http://www.unicode.org/cldr/data/>.

NOTICE TO USER: Carefully read the following legal agreement. BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"), YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA FILES OR SOFTWARE.

COPYRIGHT AND PERMISSION NOTICE

Copyright © 1991-2006 Unicode, Inc. All rights reserved. Distributed under the Terms of Use in <http://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that (a) the above copyright notice(s) and this permission notice appear with all copies of the Data Files or Software, (b) both the above copyright notice(s) and this permission notice appear in associated documentation, and (c) there is clear notice in each modified Data File or in the Software as well as in the documentation associated with the Data File(s) or Software that the data or software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

Unicode and the Unicode logo are trademarks of Unicode, Inc., and may be registered in some jurisdictions. All other trademarks and registered trademarks mentioned herein are the property of their respective owners

MISCELLANEOUS ATTRIBUTIONS

Adobe, Acrobat, and the Acrobat Logo are registered trademarks of Adobe Systems Incorporated.

FAST Instream is a trademark of Fast Search and Transfer ASA.

HP-UX is a registered trademark of Hewlett-Packard Company.

IBM, Informix, and DB2 are registered trademarks of IBM Corporation.

Jaws PDF Library is a registered trademark of Global Graphics Software Ltd.

Kofax is a registered trademark, and Ascent and Ascent Capture are trademarks of Kofax Image Products.

Linux is a registered trademark of Linus Torvalds.

Mac is a registered trademark, and Safari is a trademark of Apple Computer, Inc.

Microsoft, Windows, and Internet Explorer are registered trademarks of Microsoft Corporation.

MrSID is property of LizardTech, Inc. It is protected by U.S. Patent No. 5,710,835. Foreign Patents Pending.

Oracle is a registered trademark of Oracle Corporation.

Portions Copyright © 1994-1997 LEAD Technologies, Inc. All rights reserved.

Portions Copyright © 1990-1998 Handmade Software, Inc. All rights reserved.

Portions Copyright © 1988, 1997 Aladdin Enterprises. All rights reserved.

Third Party Licenses

Portions Copyright © 1997 Soft Horizons. All rights reserved.

Portions Copyright © 1995-1999 LizardTech, Inc. All rights reserved.

Red Hat is a registered trademark of Red Hat, Inc.

Sun is a registered trademark, and Sun ONE, Solaris, iPlanet and Java are trademarks of Sun Microsystems, Inc.

Sybase is a registered trademark of Sybase, Inc.

UNIX is a registered trademark of The Open Group.

Verity is a registered trademark of Autonomy Corporation plc



#

\$\$AND\$\$, 5-11
 \$\$AND_NOT\$\$, 5-11
 \$\$NEAR\$\$, 3-4, 5-11
 \$\$OR\$\$, 5-11

A

Abstract search rules, 5-1, 5-8, 5-17

- count, 5-8
- defining, 5-17
- examples, 5-8
- key, 5-8
- rule types, 5-8

 Account field, 5-1, 5-17

 Adaptor modules, A-1

 AddCCToArchiveCheckin (optional component), 1-8

 AddCCToNewCheckin (optional component), 1-6

 adding, 3-2

- keywords, 3-2
- search rules, 3-2

 Always weight, 5-11

 applet, 3-2

 APR Smartlogik, A-1

 Author field, 5-1, 5-17

 Autonomy, A-1

B

Batch Categorizer, 4-2

 Batch Loader file, 4-2

 Batch Loader utility, 4-2

 Batch mode, 1-2, 2-1, 4-2

- requirements, 2-1

 batch mode

- overview, 1-5
- process, 1-6

 binary operators, 5-11, 5-17

C

categories, 3-2, 5-11, 5-17

 Categorize button, 1-2

 Categorizer Engine, A-1

 CATEGORY, 5-1, 5-17

 CC Admin Applet, 3-2, 3-6, 5-17

 changing search rule order, 3-2

 Comments field, 5-1

 config.cfg, 5-17

 Configuration Manager, 3-2, 5-11

 Configuration tab, 3-2

 configuration variables

- MaxQueryRows, C-1

 configuring, 2-1, 4-2

- Content Categorizer, 2-1

 Content Admin Server, 5-17

 Content Categorizer, 1-2, 2-1

- installation, 2-1
- overview, 1-2
- setup, 2-1

 Content Check In Form, 1-2, 5-1

 Content Publisher, 1-2, 1-2, 2-1, 2-1, 3-2

 Content Server, 1-2, 2-1, 3-2, 5-17, 5-17

- restarting, 5-17
- starting, 5-17
- stopping, 5-17

 count, 3-2, 5-1, 5-3, 5-8, 5-11, 5-17

- Abstract search rules, 5-8
- Option List search rule, 5-11
- Pattern Matching search rules, 5-3
- setting, 3-2

 creating, 3-2

- keywords, 3-2
- search rules, 3-2

D

dDocType, 5-17

 default values, 3-2

- setting for batch, 3-2

Index

defining, 5-11, 5-17
 keywords, 5-11
 search rules, 5-17
 weights, 3-2
DocType, 3-2

E

elements, 5-3
 XML, 5-3
Entry in index, 3-1
ERROR log level, 3-6
examples, 5-3, 5-8, 5-11
 Abstract search rules, 5-8
 Option List search rule, 5-11
 Pattern Matching search rules, 5-3
existing content
 search rules for metadata values, 2-2
eXtensible Markup Language, 1-2, 3-2, 5-1, 5-3, 5-8

F

FIRST_PARAGRAPH, 5-8
FIRST_SENTENCE, 5-8
Flexiondoc, 1-2, 2-1
 interactive mode process, 1-5
 interactive mode requirements, 1-4
 overview, 1-3
 setting XML conversion method, 2-2
flexiondoc_to_scc.xml
 custom XSLT style sheet, 1-3
ForceDocTypeChoice, 5-17

I

Index entry
 second-level entry, 3-1
INFO log level, 3-6
installation, 2-1
 Content Categorizer, 2-1
Interactive mode, 1-2, 2-1, 5-1
 requirements, 2-1
interactive mode
 Flexiondoc process, 1-5
 Flexiondoc requirements, 1-4
 overview, 1-5
 process, 1-5
 requirements, 1-4
 SearchML process, 1-6
 SearchML requirements, 1-5

K

key, 3-2, 5-1, 5-3, 5-8, 5-11, 5-17
 Abstract search rules, 5-8
 Option List search rule, 5-11
 Pattern Matching search rules, 5-3
 setting, 3-2
keywords, 3-2, 5-11, 5-17
 adding, 3-2

L

Library main page, 6-4
log file, 4-2
 Batch Categorizer, 4-2
log levels, 3-2, 3-6

M

map file, 4-2
MapFile.xml, 4-2
MaxQueryRows
 setting batch load size limit, C-1
metadata, 1-2, 5-1, 5-11, 5-17
 assigning values for required and non-required
 fields, 2-2
 defining field properties, 2-2
 search rules for existing content, 2-2
 search rules for new content, 2-2
modes, 1-2

N

name, 3-2
 project, 3-2
 template, 3-2
Never weight, 5-11
new content
 search rules for metadata values, 2-2
non-required fields
 assigning metadata, 2-2
NONE log level, 3-6

O

operating modes, 1-2
 batch mode, 1-5
 batch mode process, 1-6
 Flexiondoc process in interactive mode, 1-5
 Flexiondoc requirements in interactive mode, 1-4
 interactive mode, 1-5

- interactive mode process, 1-5
- interactive mode requirements, 1-4
- overview, 1-5
- SearchML process in interactive mode, 1-6
- SearchML requirements in interactive mode, 1-5
- Option List search rule, 3-2, 5-1, 5-11, 5-17
 - categories, 5-11, 5-17
 - count, 5-11, 5-17
 - defining, 5-11, 5-17
 - examples, 5-11
 - key, 5-11, 5-17
 - keywords, 5-17
 - rule type, 5-11
 - values, 5-11
 - weights, 3-2, 5-11, 5-17
- Option Lists Tab, 3-2
- OPTION_LIST, 5-11, 5-17
- Optional Component
 - AddCCToArchiveCheckin, 1-8
 - AddCCToNewCheckin, 1-6
- order, 3-2, 5-1, 5-17
 - defining for search rules, 5-17
 - search rule, 3-2, 5-1
- output control file, 4-2
- overview, 1-2, 2-1
 - Content Categorizer, 1-2
 - Content Categorizer setup, 2-1

P

- Pattern Matching search rules, 5-1, 5-3, 5-17
 - count, 5-3
 - defining, 5-17
 - examples, 5-3
 - key, 5-3
 - rule types, 5-3
- properties, 3-2, 3-6
- Property Config screen, 3-2

R

- required fields
 - assigning metadata, 2-2
- requirements, 2-1
- restarting Content Server, 5-17
- Rule Sets tab, 3-2, 5-17
- rule types, 3-2, 5-1, 5-3, 5-8, 5-11, 5-17
 - Abstract, 5-8
 - Option List, 5-11
 - Pattern Matching, 5-3
 - setting, 3-2, 5-17

- Tag Search, 5-3
- Text Search, 5-3
- Rules List, 5-17
- runtime configuration parameter
 - setting to Flexiondoc, 2-2
 - setting to SearchML, 2-2
 - setting XML conversion method, 2-1

S

- sample XML conversion files
 - Well_WS_Flexiondoc.xml, 1-3
 - Well_WS_SearchML.xml, 1-3
- sccConsoleLogLevel, 3-4, 3-6
- sccConversionProject, 3-4
- sccConversionTemplate, 3-4
- sccServerLogLevel, 3-4, 3-6
- sccStrictXML, 3-4
- sccXMLConversion, 3-4
 - configuration setting values, 1-3
- score, 5-11
 - option list, 5-11
- search rule override
 - applied to fields during content check in, 1-2
- search rules, 3-2, 5-1, 5-3, 5-8, 5-11, 5-17
 - Abstract, 5-8
 - defining, 3-2, 5-17
 - guidelines, 5-1
 - Option List, 5-11
 - order, 3-2, 5-1, 5-17
 - overview of types, 1-2
 - Pattern Matching, 5-3
 - types, 5-1, 5-17
 - understanding, 5-1
- SearchML, 1-2, 2-1
 - interactive mode process, 1-6
 - interactive mode requirements, 1-5
 - overview, 1-4
 - setting XML conversion method, 2-2
- searchml_to_scc.xsl
 - custom XSLT style sheet, 1-4
- Security Group field, 5-1, 5-17
- server logs, 3-2
- setting, 3-2, 3-6, 5-17
 - batch default values, 3-2
 - count, 3-2
 - key, 3-2
 - keywords, 3-2
 - log levels, 3-6
 - properties, 3-6
 - rule types, 3-2, 5-17
 - weights, 3-2

Index

setup, 2-1
 Content Categorizer, 2-1
Site Builder, 1-2, 1-2
standard metadata fields, 5-1, 5-17
starting Content Server, 5-17
stopping Content Server, 5-17
sub-types, 5-3
 Pattern Matching, 5-3

T

Tag Search, 5-3
TAG_ALLTEXT, 5-3
TAG_TEXT, 5-3
Text Search, 5-3
TEXT_ALLFULL, 5-3
TEXT_ALLNEXT, 5-3
TEXT_ALLREMAINDER, 5-3
TEXT_FULL, 5-3
TEXT_NEXT, 5-3
TEXT_REMAINDER, 5-3
Title, 3-2
 project, 3-2
Title field, 5-1
troubleshooting, 3-6
Type field, 5-17

U

understanding search rules, 5-1

V

values, 5-11
 Option List, 5-11

W

WARNING log level, 3-6
weights, 3-2, 5-11, 5-17
 defining, 3-2, 5-11, 5-17
Well_WS_Flexiondoc.xml
 sample XML conversion file, 1-3
Well_WS_SearchML.xml
 sample XML conversion file, 1-3

X

XML, 1-2, 3-2, 5-1, 5-3, 5-8
 conversion, 1-2, 2-1
 elements, 5-3, 5-8
XML conversion
 sample files, 1-3
 sccXMLConversion configuration setting, 1-3
 setting conversion method, 2-1
 setting to Flexiondoc, 2-2
 setting to SearchML, 2-2
XML converters
 Flexiondoc, 1-3
 overview, 1-3
 SearchML, 1-4
XSLT custom style sheets
 flexiondoc_to_scc.xsl, 1-3
 searchml_to_scc.xsl, 1-4