



THE
POWER
TO KNOW.

SAS® Contextual Extraction Studio 5.2 User's Guide



The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2011.
SAS® Contextual Extraction Studio 5.2: User's Guide. Cary, NC: SAS Institute Inc.

SAS® Contextual Extraction Studio 5.2: User's Guide

Copyright © 2011, SAS Institute Inc., Cary, NC, USA

All rights reserved. Produced in the United States of America.

For a hard-copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a Web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government Restricted Rights Notice: Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19, Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st electronic book, July 2011

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at support.sas.com/publishing or call 1-800-727-3228.

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

Contents

About This Book	vii
Audience	vii
Prerequisites	vii
Conventions	viii
 What's New in SAS Contextual Extraction Studio 5.2	 ix
Overview	ix
Coreference Operators Added	ix
XML Field Specified for Matching	ix
Additional Operators and Case-Insensitive Matching	x
Case-Insensitive Matching and Comments	x
 1 About SAS Contextual Extraction Studio	 1
1.1 What is SAS Contextual Extraction Studio?	1
1.2 Benefits to Using SAS Contextual Extraction Studio	2
1.3 How Does SAS Contextual Extraction Studio Work with SAS Content Categorization Studio?	3
1.4 Architecture	4
 2 Interface Components	 5
2.1 Your First Look at the SAS Contextual Extraction Studio Interface Components	5
2.2 Start SAS Contextual Extraction Studio	6
2.3 The LITI Radio Button in the Definition Tab	6
2.4 The Priority Setting in the Data Window	7
2.5 The Compile Concepts Window	9
2.6 The Project Settings Interface	10
2.7 The Upload LITI Operation	12
2.8 Using the <language>.li File	13
 3 Writing Contextual Extraction Concept Definitions	 15
3.1 Overview of Definitions	15
3.2 Create a Project	17
3.3 Before You Write Your Contextual Extraction Definitions	18
3.4 The Rule Types	19

3.5 The Building Blocks	21
3.5.1 Overview of the Building Blocks	21
3.5.2 Case-Insensitive Matching	21
3.5.3 Entering Comments into Rules	21
3.5.4 The Tokens	21
3.5.5 The _c Marker	22
3.5.6 The _w Term	22
3.5.7 The _cap Term	22
3.5.8 The > Symbol	23
3.5.9 The Quotation Marks	23
3.5.10 The Parentheses, Square Braces, and Curly Braces	23
3.5.11 The Commas	24
3.5.12 The Colons	24
3.5.13 The Spaces	25
3.5.14 The Part-of-Speech Tags	25
3.5.15 The Export Feature	25
3.5.16 The Regular Expressions	26
3.5.17 The Priorities and Project Settings	27
3.5.17.A Overview of Priorities	27
3.5.17.B Choose Project Settings	27
3.5.17.C Choosing Priorities and Project Settings	28
3.6 The Operators	29
3.6.1 The Boolean Operators	29
3.6.1.A The ALIGNED Operator	30
3.6.1.B The AND Operator	30
3.6.1.C The OR Operator	30
3.6.1.D The DIST_n Operator	30
3.6.1.E The ORDDIST_n Operator	31
3.6.1.F The SENT Operator	31
3.6.1.G The SENT_n Operator	31
3.6.1.H The SENTSTART_n Operator	31
3.6.1.I The SENTEND_n Operator	32
3.6.2 The Stemming Operator	32
3.6.3 The PARA Operator	32
3.6.4 The Operators for Coreference Resolution	33
3.7 Contextual Extraction Concept Definition Examples	33
3.7.1 The Classifiers	33
3.7.2 Specifying a Sequence of Classifier Entries	35
3.7.3 Context Matching	36

3.7.4 Matching within Context	37
3.7.5 Eliminating Partial Matches	40
3.7.6 Disambiguating Matches	42
3.7.7 Exporting Classifiers	44
3.7.8 Setting Priorities for Overlapping Matches	47
3.7.9 Specifying Part-of-Speech Tags	50
3.7.10 Specifying Regular Expressions	51
3.7.11 Specifying a Sentence Operator	53
3.7.12 Specifying a Paragraph Operator	55
3.7.13 Specifying a DIST Operator	58
3.7.14 Specifying an ORDDIST Operator	60
3.8 Locating Facts	63
3.8.1 Overview of Facts	63
3.8.2 A Predicate Sequence Example	63
3.8.3 Predicate Examples	66
3.9 The Coreference Operators	72
3.9.1 Overview of Coreference	72
3.9.2 How to Use the Coreference Operator	72
3.9.3 How to Use the _ref Operator with the > Symbol	74
3.9.4 How to Use the _ref Operator with the Forward or Backward Symbols	74
3.9.4.A Limiting Matches to Those That Follow or Precede a Coreference Match	74
3.9.4.B Matching with the Forward Symbol	74
3.9.4.C Matching with the Preceding Symbol	76
3.9.5 Coreference in a Classifier Definition Example	77
3.9.6 Assigning New Concept Names to Coreference Matches	78
3.9.7 Rank Coreference Definitions and Eliminate False Positives	79
3.10 XML Fields	81
3.10.1 Overview of XML Field Matching	81
3.10.2 SEQUENCE Rules with an XML Field	81
3.10.3 Matching More than One XML Field	83
3.11 Writing Multiple Rules for One Definition	84
3.12 Troubleshooting Your Rules	85
Appendixes	87
A Using the Directive and Regex Syntax	89
A.1 Using the Directive in the Configuration File	89
A.2 Regular Expressions	90

A.2.1 Rules and Restrictions	90
A.2.2 Special Characters	91
A.2.3 Special Cases	92
B Part-of-Speech Tags	93
C Recommended Reading	99
D Glossary	101
Index	105

About This Book

Audience

SAS Contextual Extraction Studio is designed for subject matter experts who write the complex rules that identify the context-sensitive metadata in your organization's input documents. The metadata returned as a match on a concept identifies the data existing in your information. The contextual extraction concepts defined by you are an extension of the classifier and grammar concepts that are available in SAS Content Categorization Studio.


Prerequisites

Here are the prerequisites for using SAS Contextual Extraction Studio:

- SAS Contextual Extraction Studio loaded onto your machine, *after* you install SAS Content Categorization Studio
- Access to representative documents where you want to locate metadata
- Appropriate server permission for users who upload the output .lii binary file to SAS Content Categorization Collaborative Server to be applied to input documents

Conventions

This manual uses the following typographical conventions:

Convention	Description
TGM_ROOT	The root directory where SAS Contextual Extraction Studio is installed, typically the following: Windows: C:/Program Files/SAS/SAS Content Categorization Studio UNIX: /opt/SAS Content Categorization Studio
.li	The code examples for the .li file are shown in a fixed-width font.
LITI button	The labels for the user interface controls are shown in a bold, sans-serif font.
Top	The names of taxonomy nodes appear in a fixed-width font.
www.sas.com	The hypertext links are shown in a light blue, fixed-width font, and are underlined.
	The Question Mark button accesses <i>SAS Content Categorization Studio: User's Guide</i> in PDF format. Use this book to obtain user interface information for SAS Contextual Extraction Studio.

What's New in SAS Contextual Extraction Studio 5.2

Overview

New and enhanced features in SAS Contextual Extraction Studio include the following:

- Added coreference operators facilitate rule-writing precision.
- XML fields can be specified for matches.
- Case-insensitive matching and comments in rules are now enabled.

Coreference Operators Added

Coreference refers to pronoun resolution. A pronoun is matched to the antecedent that it refers to when you use these operators in your contextual extraction concept rules:

- Use the coreference operator (`_ref`) to link a matched string with its canonical form.
- Use `_coref` with `CLASSIFIER` definitions.
- Use the forward (`_F`) and the preceding (`_P`) symbols to restrict coreference matches.
- Assign a new concept name for a match on a term specified by the `_ref` operator.

XML Field Specified for Matching

Limit matches to specific XML fields when you write these fields into rules and apply them to input XML documents.

Additional Operators and Case-Insensitive Matching

Additional operators enable greater rule matching precision. These operators include:

- Specify a stemming symbol to enable SAS Contextual Extraction Studio to match all word forms, or only all noun or verb forms.
- Specify the paragraph symbol (`¶`) to enable SAS Contextual Extraction Studio to match all word forms, or only all noun or verb forms.
- Write a `SENT_n` operator into a rule to specify the maximum number of sentences where a match can occur.
- Use a `SENTSTART_n` operator to specify the number of words at the beginning of a sentence where a match can occur.
- Use a `SENTEND_n` operator to specify the number of words at the end of a sentence where a match can occur.

Case-Insensitive Matching and Comments

Case-insensitive matching occurs when you select the **Case Insensitive Matching** check box in the **Data** tab for a contextual extraction concept. (By default, all matching is case sensitive.)

You can also add comments to your rules using the pound character (#).

1

About SAS Contextual Extraction Studio

- *What is SAS Contextual Extraction Studio?*
- *Benefits to Using SAS Contextual Extraction Studio*
- *How Does SAS Contextual Extraction Studio Work with SAS Content Categorization Studio?*
- *Architecture*

1.1 What is SAS Contextual Extraction Studio?

In most organizations it is necessary to identify metadata, or data on information. This metadata is located in your documents, created internally and externally, and stored in your company's repositories.

SAS Contextual Extraction Studio, or LITI, uses advanced linguistic technologies to extend the concepts rule-writing features available in SAS Content Categorization Studio. SAS Contextual Extraction Studio enables you to write complex definitions that can include several types of rules in a single concept in order to identify the metadata in input documents. (The terms *definitions* and *rules* are used interchangeably in this book. Properly speaking, *definitions* apply to concepts.)

Using the intuitive, Windows based interface in the SAS Content Categorization Studio application, subject matter experts write complex rules to define each contextual extraction concept in the taxonomy.

1.2 Benefits to Using SAS Contextual Extraction Studio

SAS Contextual Extraction Studio expands the benefits available in SAS Content Categorization Studio:

Context sensitive matching

Match concepts within a specified context, only. For example, match New York but not New York City.

Syntax building blocks

Write your definitions to locate concept matches using parts of speech, logical operators, regular expressions, and separator characters.

Concept disambiguation

Return only the specific concept that you are seeking. For example, differentiate between Giants football and Giants baseball.

Relational concepts

Return related concepts. For example, locate the string *Drew Faust is president of Harvard University*, where *Drew Faust* and *Harvard University* are concepts.

Fact extraction

Extract facts from seemingly unrelated pieces of data, similar to relational concepts. Specify operators between the concepts that together form a fact to return the entire string. For example, Tide is produced by Procter & Gamble.

Write multiple rules for one definition

Match on any rule, within a concept definition, in an input document and return a match on this concept.

Write different types of rules for one definition

Specify different types of rules for one definition.

Prioritize contextual extraction concepts over matches on classifier and grammar concepts

Leave the default **Priority** setting in the **Data** tab, or increase this specification.

Determine how SAS Content Categorization Studio treats overlapping, identical, or duplicate matches

Specify the appropriate settings using the Project Settings - LITI dialog box to determine the matching process in these cases.

Easy Uploading

After you develop and test the taxonomy, upload the .li file to SAS Content Categorization Server where the contextual extraction concept definitions are applied to incoming documents.

Sample Project

A sample SAS Contextual Extraction Studio project is included.

1.3 How Does SAS Contextual Extraction Studio Work with SAS Content Categorization Studio?

Use SAS Contextual Extraction Studio to write multiple contextual extraction rules that together define each contextual extraction concept that you develop in SAS Content Categorization Studio.

The functionalities of SAS Contextual Extraction Studio are fully integrated into the SAS Content Categorization Studio user interface. Anyone can use this interface to develop taxonomies, define concepts, and upload the output .li file to SAS Content Categorization Collaborative Server.

1.4 Architecture

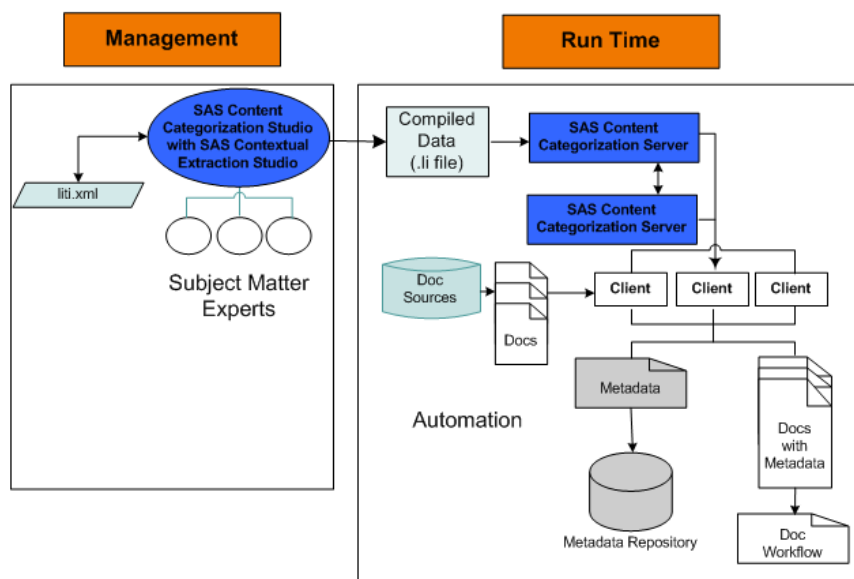
Use the architecture diagram below to gain an overview of the project development processes.

During the management phase, subject matter experts specify concepts that are based on the information in the documents where these terms are located.

During the second part of the management phase, developers write definitions. These definitions ensure that all of the documents that should match a concept are located and those that should not match a concept are not returned.

At run time, the compiled SAS Contextual Extraction Studio data, in the form of the .li binary file, is sent to SAS Content Categorization Server. The SAS Content Categorization Server returns metadata about input documents, based on the matching concepts.

Figure 1-1 SAS Contextual Extraction Studio Architecture



2

Interface Components

- *Your First Look at the SAS Contextual Extraction Studio Interface Components*
- *Start SAS Contextual Extraction Studio*
- *The LITI Radio Button in the Definition Tab*
- *The Priority Setting in the Data Window*
- *The Compile Concepts Window*
- *The Project Settings Interface*
- *The Upload LITI Operation*
- *Using the <language>.li File*

2.1 Your First Look at the SAS Contextual Extraction Studio Interface Components

SAS Contextual Extraction Studio, also known as LITI, works with SAS Content Categorization Studio. The interface components specific to SAS Contextual Extraction Studio appear in the SAS Content Categorization Studio interface, after you install both programs. (SAS Content Categorization Studio is installed first.)

You have the benefits of all of the interface components featured in the SAS Content Categorization Studio interface, in addition to the LITI-specific rule selections, when you use SAS Content Categorization Studio. For information about the SAS Content Categorization Studio Windows interface, see the *SAS Content Categorization Studio: User's Guide*.

This chapter presumes a working knowledge of SAS Content Categorization Studio and, for this reason, provides information that is specific only to SAS Contextual Extraction Studio.

2.2 Start SAS Contextual Extraction Studio

To open SAS Contextual Extraction Studio in SAS Content Categorization Studio, select **Start —> Programs —> SAS Content Categorization Studio —> SAS Content Categorization Studio**. The SAS Content Categorization Studio user interface appears.

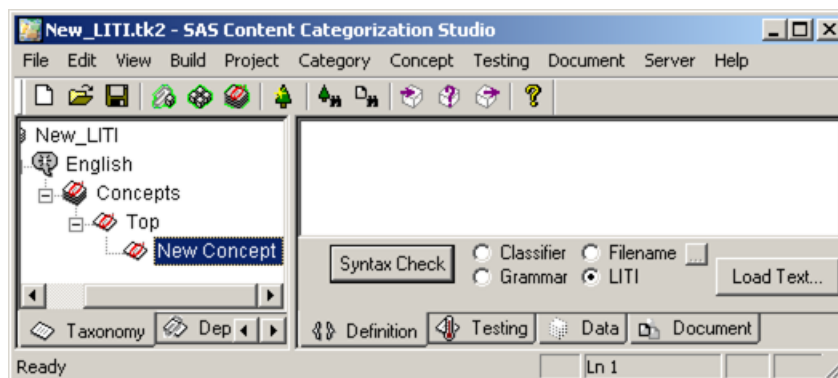


2.3 The LITI Radio Button in the Definition Tab

The **LITI** button appears in the Definition window after you install SAS Contextual Extraction Studio and add one or more concepts to your taxonomy. This component enables you to write, check, and compile contextual extraction concept definitions.

To specify a contextual extraction concept, complete these steps:

1. Right-click on the **Top** node and select **Add Concept** from the drop-down menu that appears.
2. Name the concept.



3. Select **LITI**. This radio button only becomes available after you name the concept.

Note: You must select **LITI** *before* you write the concept definition.

4. Write the contextual extraction concept definition. For more information, see Chapter 3: *Writing Contextual Extraction Concept Definitions*.

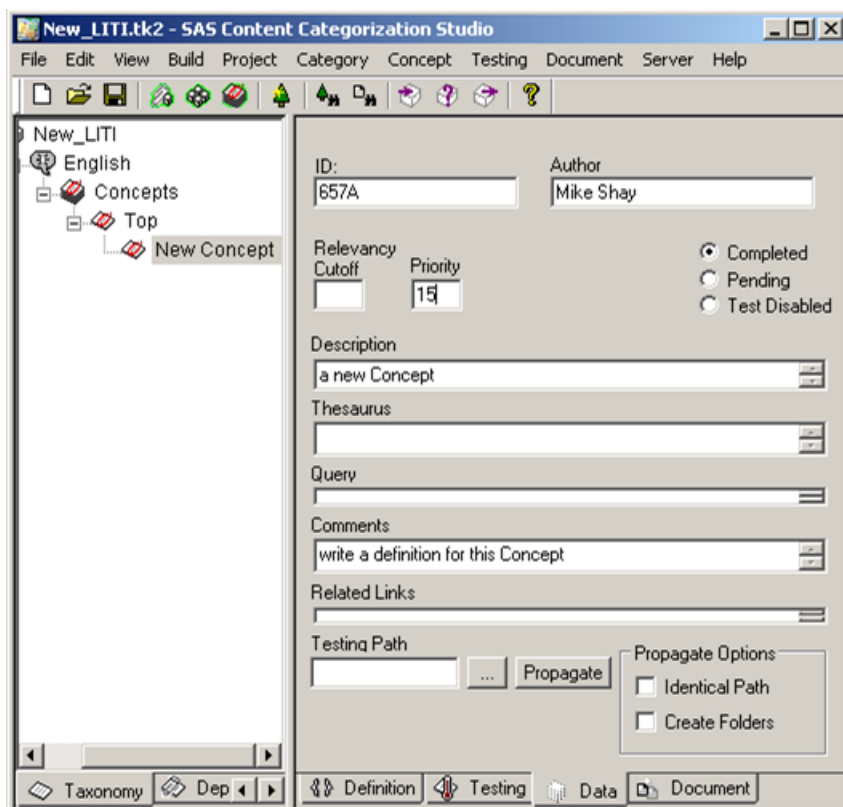
2.4 The Priority Setting in the Data Window

The **Priority** field in the **Data** tab enables you to set the ranking order of contextual extraction concepts higher, or lower, than classifier or grammar concepts. The default setting, 10, ranks all of the contextual extraction concepts higher than the classifier or grammar concepts developed in SAS Content Categorization Studio.

You can change the default setting, one concept at a time. If you do not want to prioritize the selected contextual extraction concept, specify 0. To increase the priority of the selected contextual extraction concept enter a number that is higher than 10.

To reset the **Priority** setting for one contextual extraction concept, complete these steps:

1. Select the **Data** tab.
2. Type a new number into the **Priority** field.



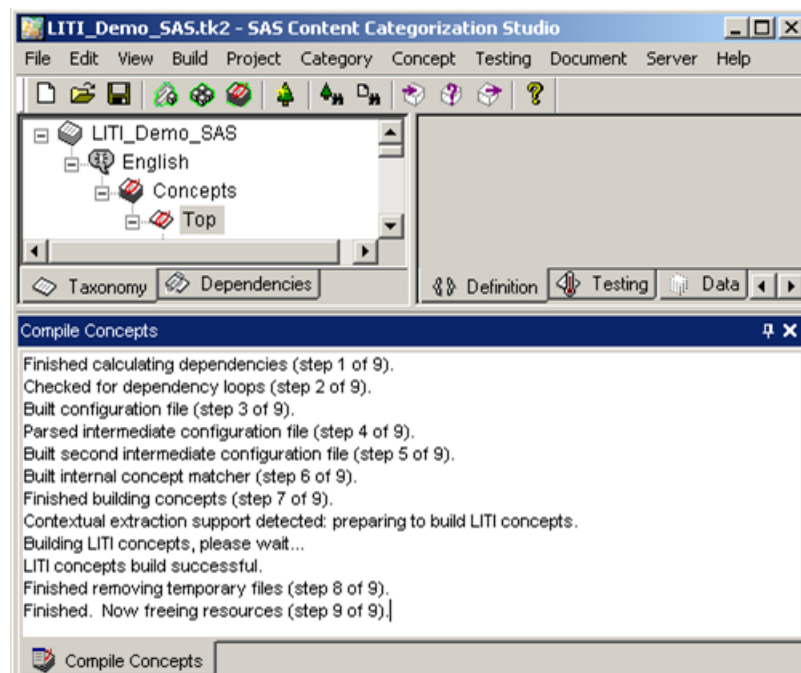
3. Select **Build** —> **Compile Concepts**.
4. Select **File** —> **Save**.

2.5 The Compile Concepts Window

Compile your concepts to ensure accuracy and to integrate all of the changes into the SAS Contextual Extraction Studio project. The Compile Concepts window that appears at the bottom of the SAS Content Categorization Studio interface displays information about the contextual extraction concepts. This data confirms a successful build, or it points to errors in the definitions.

To compile your concepts, complete these steps:

1. Select **Build —> Compile Concepts** and the Compile Concepts window appears at the bottom of the SAS Content Categorization Studio interface.



2. Locate the lines explaining the LITI concepts.

These self-explanatory messages include the following:

- Contextual extraction support detected: preparing to build LITI concepts

-
- Building LITI concepts, please wait...
 - LITI concepts build successful. If unsuccessful, the build fails and an explanation appears.
3. Click **X** in the upper right-hand corner of the Compile Concepts window.

2.6 The Project Settings Interface

Set project-wide settings for your contextual extraction concepts by using the Project Settings - LITI dialog box. These settings determine how matches in input documents are returned. For this reason, the settings that you specify in the Project Settings - LITI window affect the testing results that you see in the Document window and those returned by SAS Content Categorization Server.

Use the Project Settings Concordance window to specify the surrounding text that is returned with a match, if you choose to use the Concordance selection in the Document window. For more information, see the *SAS Content Categorization Studio: User's Guide*.

These radio buttons and check boxes enable you to make decisions concerning overlapping and identical matches and to remove all duplicate facts. The term *fact* is used to refer to two or more concepts or tokens. These concepts, or terms, are specified in one definition in order to define a relationship between otherwise isolated instances of information. For more information, see Section 3.8 *Locating Facts* on page 63.

To specify settings in the Project Settings - LITI dialog box, complete these steps:

1. Select **Project —> Settings** and the Project Settings - Concept dialog box appears.

-
2. Select LITI and the Project Settings - LITI tab appears.



3. Select one radio button under the **Overlapping Concept Matches** heading that determines how SAS Contextual Extraction Studio treats overlapping matches. Overlapping matches are strings where part, or all, of the string matches more than one concept.
 - Leave the default selection, **All matches**, selected and SAS Content Categorization Studio returns all of the terms that match any of the contextual extraction concept definitions in this project
 - Select **Longest** to return the longest match for the concept definition.
 - Select **Best** to return the match with the highest priority setting, only.

Note: If all of the tested concepts have the same priority setting, only the longest matches are returned. For more information, see Section 3.5.17 *The Priorities and Project Settings* on page 27.

4. If you select either the **Longest** or **Best Matches** radio button, **Return all identical matches** becomes available. Select this check box and SAS Content Categorization Studio returns all of the identical longest or best matches.

-
5. Select **Remove duplicate facts** when you want to delete matches on multiple arguments that comprise a single fact. This selection applies only to a predicate sequence, or to predicate, definitions. For more information, see Section 3.8 *Locating Facts* on page 63.

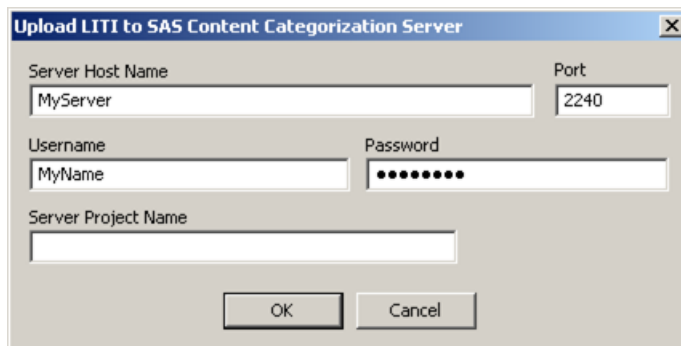
Note: These settings do not affect the returns specified by the `REMOVE_ITEM` rule that excludes matches on a concept for disambiguation purposes. For more information, see Section 3.7.6 *Disambiguating Matches* on page 42.

2.7 The Upload LITI Operation

After you build your taxonomy of contextual extraction concepts, you can upload these concepts to SAS Content Categorization Server.

To upload your LITI concepts, complete these steps:

1. Select **Build —> Compile Concepts**.
2. (If the compilation process was successful) Select **Build —> Upload LITI** and the Upload LITI to SAS Content Categorization Collaborative Server dialog box appears:



3. Type the name of the server where SAS Content Categorization Collaborative Server is located into the **Server Host Name** field.

-
4. (Optional) The **Port** number for this server is entered by default. Type in a new port number, if this number is incorrect.
 5. Type the user name that you use to access SAS Content Categorization Collaborative Server on this server into the **Username** field.
 6. Type the password that you use to access SAS Content Categorization Collaborative Server on this server into the **Password** field.
 7. Type the name of the project as you want it to appear on the server into the **Server Project Name** field.
 8. Click **OK**.

2.8 Using the <language>.li File

When you upload your contextual extraction concepts, SAS Content Categorization Studio creates a <language>.li file. This binary file includes the <language>.concepts file that is created for the classifier and grammar concepts in SAS Content Categorization Studio.

This feature enables you to reference SAS Content Categorization Studio classifier and grammar concepts when you write your definitions for contextual extraction concepts. However, when SAS Content Categorization Studio grammar and classifier concepts are built, the resulting <language>.concepts file does not include a <language>.li file.

3

Writing Contextual Extraction Concept Definitions

- *Overview of Definitions*
- *Create a Project*
- *Before You Write Your Contextual Extraction Definitions*
- *The Rule Types*
- *The Building Blocks*
- *The Operators*
- *Contextual Extraction Concept Definition Examples*
- *Locating Facts*
- *The Coreference Operators*
- *XML Fields*
- *Writing Multiple Rules for One Definition*
- *Troubleshooting Your Rules*

3.1 Overview of Definitions

SAS Contextual Extraction Studio uses Teragram Advanced Linguistic technologies to expand the concept development capabilities in SAS Content Categorization Studio:

- Write a simple rule that matches one term specified in a list of entries.
- Locate a match for a unique concept where each individually specified term in the concept appears in one of the rules that together define this concept.
- Match a concept if it appears within the specified context, only.

-
- Locate multiple partial matches and return them as full concept matches. These matches can occur only if there is a match on the fully defined concept within the input document.
 - Write restrictive rules to prevent matches from occurring within specified contexts.
 - Disambiguate matches. Avoid possible matches on concepts that are specified using identical terms with different meanings.
 - Specify part-of-speech tags to locate concepts.
 - Use Boolean operators and various types of operators to increase the matching precision of your rule.
 - Specify case-insensitive rule matches in the Data window in order to change the case-sensitive default setting.
 - Use the stemming operator to return all of the forms of a word. Alternatively, choose to return only the noun or verb forms of the word.
 - Specify coreference operators for pronoun resolution. In other words, when a pronoun or another word refers to the canonical form for a term, return the canonical form.
 - Specify the canonical form that is used for pronoun resolution.
 - Use the `PRIORITY` setting to specify that one rule is weighted more than another and to prevent the return of false positives for coreference matches. (In other words, you can rank one rule higher than another rule within the same definition.)

Note: By default the **Priority** setting is set to 10 in the Data window for concepts, only. (However, this setting is applied to any SAS Content Categorization Studio concepts with a **Priority** setting of less than 10. This change is made when the project is uploaded as a binary file.)

- Match predicates by specifying multiple arguments to extract a fact.
- Identify the semantic relations between concepts by using predicate rules with logical operators.
- Specify XML fields to limit matches to these fields.

-
- You can write comments into your rules.

This chapter describes how to write the definitions that specify your concepts and provides examples of these rules. Before you write your rules, see the following sections. These sections provide information about prerequisite knowledge.

3.2 Create a Project

To create your concepts project, complete these steps:

1. Select **Start —> Programs —> SAS Contextual Extraction Studio —> SAS Contextual Extraction Studio**.
2. Name the new project, specify a language, and enable concepts.
3. Right-click on the `Top` node under the `Concepts` node in the Taxonomy window and select **Add Concept** from the drop-down list that appears.
4. After the new concept is added, type the name of this concept into the box that appears to the right of the new node. Click the **LITI** radio button in order to specify its type.
5. Write the rule using the Definition window.
6. Click **Syntax Check** to check the rule.
7. Select **Build --> Compile Concepts** before you apply the rule to any input documents.

3.3 Before You Write Your Contextual Extraction Definitions

Consider the following information before you write your contextual extraction concept definitions:

- The terms *rule* and *definition* are used interchangeably. Properly speaking, definitions apply to all of the rules for one contextual extraction concept.
- Rule types, for example `CLASSIFIER` and `C_CONCEPT`, are written using uppercase letters.
- By default, SAS Contextual Extraction Studio performs case-sensitive matching.
- Matches are returned for contextual extraction concepts when matches also occur on classifier or grammar SAS Content Categorization Studio concepts. By default, the **Priority** setting in the Data window is set to 10 for all contextual extraction concepts. You can also specify a `PRIORITY` setting that overrides this setting within some contextual extraction rules.

Note: When you define classifier and grammar SAS Content Categorization Studio concepts as well as LITI concepts, the priority for the classifier and grammar concepts changes to 10. This behavior occurs when both types of concepts are extracted to the binary file.

- By default, matches can occur in any part of an input document. When the `PARA` or various `SENT` operators are specified, a match is returned if the matches occur in one paragraph, sentence, or the specified number of sentences.
- The settings in the Project Settings - LITI dialog box can affect match returns.

3.4 The Rule Types

There are many types of contextual extraction concept rules. Unlike the classifier and grammar concepts in SAS Content Categorization Studio, you can specify more than one rule for each of your contextual extraction concept definitions. A match on the concept occurs if there is a match on any one of these rules.

CLASSIFIER

Specify lists of terms, like you do for classifier concepts in SAS Content Categorization Studio. However in SAS Contextual Extraction Studio, each Classifier rule consists of the word `CLASSIFIER` followed by a string. For more information, see Section 3.7.1 *The Classifiers* on page 33.

CONCEPT

Reference one or more concepts and use the `_cap` term to specify that a match only occurs on a word that begins with an uppercase letter. When more than one concept is referenced, a relationship is specified between the matching terms. You can also use `CONCEPT` rules to locate, or to discover, related information. For more information, see Section 3.7.2 *Specifying a Sequence of Classifier Entries* on page 35.

C_CONCEPT

Specify the order for the match components in an input document using these contextual extraction concept rules. For more information, see Section 3.7.3 *Context Matching* on page 36.

NO_BREAK

Prevent partial matches on a term that is specified within this definition. Use this rule to determine that an entire phrase is treated as a single word. For more information, see Section 3.7.5 *Eliminating Partial Matches* on page 40.

REMOVE_ITEM

Eliminate a false match in input documents where one word is a unique identifier for two concepts. This rule ensures that the correct context for the match is considered. For more information, see Section 3.7.6 *Disambiguating Matches* on page 42.

REGEX

Match information that follows a preset pattern. This rule uses the same syntax as the regular expression classifier concept definitions in SAS Content Categorization Studio. For more information, see Appendix A: *Using the Directive and Regex Syntax* and Section 3.7.10 *Specifying Regular Expressions* on page 51.

CONCEPT_RULE

Specify Boolean operators to increase precision (relevancy of the matches) and recall (return all matching texts). For more information, see Section 3.7.11 *Specifying a Sentence Operator* on page 53.

SEQUENCE

Extract facts from input documents if the facts appear in the order specified. For more information, see Section 3.8.2 *A Predicate Sequence Example* on page 63.

PREDICATE_RULE

Specify the arguments that define your facts. Facts are related pieces of information in a text that are often located and matched as phrases. For more information, see Section 3.8.3 *Predicate Examples* on page 66.

3.5 The Building Blocks

3.5.1 Overview of the Building Blocks

SAS provides *n*-gram sequence features that are often used in natural Language Processing (NLP). These sequences specify the context that is necessary for the specified concept to match. Before you write your contextual extraction concept rules, consider the building blocks that are explained in this section.

3.5.2 Case-Insensitive Matching

By default, SAS Contextual Extraction Studio applies rules to input documents in a case-sensitive manner. You can specify case-insensitive matching when you click **Case Insensitive Matching** in the **Data** tab. This setting applies to the entire definition of the selected concept, only.

3.5.3 Entering Comments into Rules

Any character, or characters, following the pound sign (#) are considered to be comments. For a literal # to match, it should be escaped as \#.

3.5.4 The Tokens

Add tokens to your definitions:

- words, including noise words such as *and*, *the*, and *a*
- numbers including date and time
- newline mark
- URLs

Specify an undetermined token using the `_w` term. When you specify this term, SAS Contextual Extraction Studio returns a match on any word that occurs in this position in the document. If, on the other hand, there is an exact token that you want this contextual extraction concept to match, you can specify this word in any concept rule. When tokens are specified in `CONCEPT_RULES` and

PREDICATE_RULES, these tokens are set off with quotation marks (" "). For more information, see Section 3.5.6 *The _w Term* below.

3.5.5 The _c Marker

Use the context marker (_c) to specify that a match is returned if the keyword is located within the specified context. For example, you can any COMPANY concept that is immediately followed by the term *New York*:

```
C_CONCEPT:_c{COMPANY} New York
```

You can also use this marker to locate and return known and unknown words. See the following examples:

```
C_CONCEPT:COMPANY _c{New York}  
C_CONCEPT:COMPANY _c{_cap}
```

3.5.6 The _w Term

Use the word term (_w) to specify that a match can occur on a word. For example, you can match any type of business. This is true if _w immediately follows a reference to the COMPANYTYPE concept:

```
C_CONCEPT:_c{COMPANYTYPE} _w
```

This example could also return a match on law *firm*.

Hint: The _w term matches any single term. A term can consist of alphabetic or non-alphabetic characters. For example, *today*, *<*, *Web*, *1.0*, and so on.

3.5.7 The _cap Term

Use the _cap term in ways that are similar to the _w term. However, _cap only returns matches on words that begin with an uppercase letter. Use _cap to locate an unknown term that begins with an uppercase letter, or to match a single upper case letter. Alternatively, specify this term multiple times. When you repeatedly specify _cap, you can locate all of the unknown, consecutive occurrences of words that begin with an uppercase letter. This term can be

used with all of the contextual extraction rule types except for the `CLASSIFIER` and `REGEX` rules. You can also replace `_w` with `_cap` in the example provided for Section 3.5.6 *The _w Term* above. In this case, the word *Firm*, or another word beginning with an uppercase letter, is a match.

3.5.8 The > Symbol

Documents often reference a unique, full string only once. After this match these references might be made by one word from the original string. Use the greater than (>) symbol with either the `C_CONCEPT`, or `CONCEPT_RULE`, or a coreference operator (`_ref`). For more information about coreference, see Section 3.9.3 *How to Use the _ref Operator with the > Symbol* on page 74. Every occurrence of the bracketed term is a match if the entire rule is matched at least once in the input text.

Specify the greater than symbol within the `C_CONCEPT` rule using the `_c{ }>` syntax. For example, use this symbol to specify that every instance of the last name *Pelosi* should be returned as a match after the entire term *Ms. Nancy Pelosi* is located. See the following example where `TITLE` and `FIRST` refer to classifier concepts with a list of titles and first names, respectively:

```
C_CONCEPT:TITLE FIRST _c{_cap}>
```

3.5.9 The Quotation Marks

Use quotation marks (") to enclose tokens and concepts when writing a `CONCEPT_RULE`, `REMOVE_ITEM`, or `PREDICATE_RULE`. This example returns a match on *Mount Washington* if the term *Mount*, and a match on the concept `NAME`, appear within seven words of a match on the `STATE` concept:

```
CONCEPT_RULE:(DIST_7, "_c{Mount NAME}", "STATE")
```

3.5.10 The Parentheses, Square Braces, and Curly Braces

Use parentheses (`()`), square braces (`[]`), and curly braces (`{}`) as appropriate. These symbols qualify the matches for all of the contextual extraction definitions except the `CLASSIFIER` and `CONCEPT` types.

Use parentheses (`()`) to group the elements that comprise `CONCEPT_RULE`, `REMOVE_ITEM`, `SEQUENCE`, and `PREDICATE_RULE` definitions. For example, use parentheses with arguments and logical operators. Parentheses are also used with the `AND`, `OR`, `SENT`, `DIST_n`, `ORDDIST_n`, and `ALIGNED` Boolean operators. These operators are followed by a comma (`,`) and a space.

Use square braces (`[]`) to group `REGEX` rule elements with the Export operation. For more information, see Section 3.5.15 *The Export Feature* on page 25.

Use curly braces to delimit the information that is returned as a match. Curly braces (`{}`) are used with or without parentheses (`()`), according to the type of definition that is specified. For more information, see Section 3.8.2 *A Predicate Sequence Example* on page 63 and the following example:

```
CONCEPT_RULE: ( SENT,   "_c{FIRST, _cap}" ,
                  "TITLE" , "COMPANY" )
```

3.5.11 The Commas

Commas (`,`) always follow definition elements:

- Boolean operators are enclosed in parentheses (`()`) and a space follows the comma (`,`) after this string.
- Quotation marks (`"`) enclose concept names and a comma follows the second quotation mark.
- Separate the arguments used to construct facts with commas.
- Commas follow logical operators in a `PREDICATE_RULE`.

3.5.12 The Colons

Use a colon (`:`) in the following cases:

- Type a colon after specifying the concept rule type. For example, use a colon with these rules `CONCEPT`, `CLASSIFIER`, and `CONCEPT_RULE`.
- Use a colon when specifying terms to export to `CLASSIFIER` rules. For more information, see Section 3.7.7 *Exporting Classifiers* on page 44.

-
- Use colons between arguments for a `SEQUENCE` or `PREDICATE_RULE` concept. For more information, see Section 3.8.2 *A Predicate Sequence Example* on page 63 and Section 3.8.3 *Predicate Examples* on page 66.
 - Type a colon before a part-of-speech tag. For example, type `:Prep` and `:sep`. For more information, see Section 3.7.9 *Specifying Part-of-Speech Tags* on page 50.

3.5.13 The Spaces

When you write `CONCEPT`, `CONCEPT_RULE`, or `C_CONCEPT` definitions, type at least one space before each of the following items, tokens, concepts, part-of-speech tags, `_w` terms, and `_cap` terms. Also type a space before the `_c` marker if it is preceded by a token, comma (,), or the name of a concept. See the following example:

```
CONCEPT_RULE:(ORDDIST_9, "_c{_cap} :sep _cap :sep and  
_cap", "ORGTYP")
```

3.5.14 The Part-of-Speech Tags

Specify part-of-speech tags when you don't know the exact word that you are seeking. For example, `:Prep` to represent preposition and `:sep` to specify a separator character. A separator character is any punctuation mark. These part-of-speech tags are preceded by a colon (:) and a space. See the following example:

```
CONCEPT_RULE:(SENT, "_c{VACATION :Prep _cap :sep  
LOCATION}", "vacation")
```

For a complete list of part-of-speech tags, see Appendix B: *Part-of-Speech Tags*.

3.5.15 The Export Feature

Export a matched term to one or more concepts. Use the `Export=` operation to define a term that matches a classifier concept. Also use the coreference operator (`_ref`) with the export symbol to eliminate false positives. You can specify this operation within the definition. Alternatively, declare an acronym

as part of the definition for the concept where the selected term is exported. See the following example:

```
FULLNAME: CLASSIFIER: [export=eLN:Clinton]: Bill Clinton
LASTNAME: eLN
```

The term `Clinton` is exported to the `LASTNAME` concept. When you write the export operation into a Classifier rule, all instances of partial matches such as *Clinton*, are returned. For this reason, the export feature functions in ways that are similar to the effects of placing the greater than symbol (`>`) at the end of a rule. For more information, see Section 3.7.7 *Exporting Classifiers* on page 44.

3.5.16 The Regular Expressions

Match known patterns by using regular expressions to specify a range of letters or numbers inside square braces (`[]`). For example, place `a-z` or `0-9` inside square braces. This specification matches any word beginning with an ASCII character whose value is between `a` and `z`, or numbers between `0` and `9` inclusive. You can also add a plus (`+`) sign after the last square brace. See the following example:

```
REGEX: [a-z] +
```

When you add the plus sign, all instances of terms beginning with a lowercase letter match any and all occurrences of a word that appears in the input document. You can continue to build this definition by specifying a context for the word occurrence.

You can also add either the `%` symbol or write out `percent`, after these bracketed numbers. This feature enables you to locate percentage matches in your documents. See the following example:

```
REGEX: [0-9] +%
REGEX: [0-9] + percent
```

This regular expression specifies that only numbers followed by the percentage sign match. For example, `99%`, or `50 percent`, are both matches.

For more information, see Appendix A: *Using the Directive and Regex Syntax*.

3.5.17 The Priorities and Project Settings

3.5.17.A Overview of Priorities

Priorities determine the concepts that are matched when priorities are applied to input documents by SAS Content Categorization Server. Matching is displayed in the Document pane and is applied after the concepts are uploaded to SAS Content Categorization Server as binary files.

For example, you might have a document that contains matches for both concept A and concept B. To prioritize a match on concept A, set the **Priority** setting in the **Data** tab for concept A to a higher number than that of concept B. Alternatively, you could specify `PRIORITY=n` in one or more rules in your definitions.

The `PRIORITY` rule specification that is set higher than 10, overrides the **Priority** setting in the **Data** tab. By default, the **Priority** setting in the **Data** tab is set to 10. For this reason, a `PRIORITY` setting in a rule ranks overlapping rule matches in one concept definition as well as matches on different concept definitions. For more information, see Section 3.7.8 *Setting Priorities for Overlapping Matches* on page 47 and Section 3.9.7 *Rank Coreference Definitions and Eliminate False Positives* on page 79.

See the following example where 35 overrides the default **Priority** setting of 10 in the Data window:

```
C_CONCEPT:PRIORITY=35: _c{CITY COUNTRY}
```

Note: When you upload SAS Content Categorization Studio concepts to SAS Content Categorization Server, the default priority settings are also set to 10 for the SAS Content Categorization Studio concepts.

3.5.17.B Choose Project Settings

Use the **LITI** tab in the Project Settings window to choose the types of matches that you want to return. These settings are particularly important when you specify priorities and when multiple matches occur within one input document.

To specify Project Settings, complete these steps:

-
1. Select **Project --> Settings** and the Project Settings window appears.



2. Select **All matches** to return matches on all of the matching rules in an input document.
3. Select **Longest** to return only the match with the most characters.
4. Select **Best** to return only the best match.
5. Select **Return all identical matches** when you want to locate each instance of a rule match.
6. If you specified either a `PREDICATE` or a `SEQUENCE` rule, you can select **Remove duplicate facts** to return the first instance of a match, only.

3.5.17.C Choosing Priorities and Project Settings

Selecting project settings is an important consideration when you specify priorities. For more information and an example, see Section 3.7.8 *Setting Priorities for Overlapping Matches* on page 47.

3.6 The Operators

3.6.1 The Boolean Operators

To locate related information with greater precision, specify Boolean, or logical operators, with some types of contextual extraction rules.

Table 3-1: Boolean Operators

Operator	Description
<u>ALIGNED</u>	Disambiguate between matches on two contextual extraction concept rules. Disambiguation enables SAS Contextual Extraction Studio to determine the correct match based on context. When terms are disambiguated, only the match is returned.
<u>AND</u>	Specify that a match can occur only when both arguments are present, somewhere within the entire document.
<u>OR</u>	Specify that a match is returned if one, but not both, of the concepts or tokens is located.
<u>DIST_n</u>	Specify the number of words between matches on rule terms. The first match takes the starting position 1, while the last match falls at or before the specified number of words.
<u>ORDDIST_n</u>	Specify the maximum word count between arguments. Otherwise this operator functions like the DIST operator above.
<u>SENT</u>	Specify a sentence delimiter. For example, ., ?, or !. A match is returned when all of the specified components are located in the sentence where the first match occurs.
<u>SENT_n</u>	Specify a sentence delimiter that returns matches on multiple sentences.
<u>SENTSTART_n</u>	Specify that matches are returned within n words from the start of the sentence.
<u>SENTEND_n</u>	Specify that matches are returned within n words from the end of the sentence.

Specify a comma (,) and a space after a Boolean operator and enclose it in parentheses (()). For example, write (SENT, "NAME").

3.6.1.A The ALIGNED Operator

Specify the `ALIGNED` operator to refer to a term that matches two concepts within one rule. The presence of this operator enables SAS Contextual Extraction Studio to determine what concept is an exact match for this term.

For example, the following rule specifies that if a term matches both the `LOC` and `PERSON` concepts, only a match for the `PERSON` concept is returned. Matches for the `LOC` concept, such as *Washington*, are returned as a match on the `PERSON` concept:

```
REMOVE_ITEM:ALIGNED, ("_c{LOC}", "PERSON")
```

3.6.1.B The AND Operator

Specify the `AND` operator for two or more arguments. A match occurs only if both arguments are present. For example, the following rule limits matches to *Bills* in documents where the word *football* also occurs:

```
CONCEPT_RULE:(AND, "_c({Bills})", "football")
```

3.6.1.C The OR Operator

Specify the `OR` operator for two or more matched rule components. A match occurs for an input document if at least one of these components is present. For example, the following rule matches if either the token *Barack* or *Obama* is present in the text:

```
CONCEPT_RULE:(OR, "_c{Barack}", "_c{Obama}")
```

3.6.1.D The DIST_n Operator

Specify the maximum distance, in words, between located terms in order for a match to be returned for the selected concept. For example, if you want to specify that a match on the `FULLNAME` concept that appears within eight words of *Harvard University* is a match for the concept, write:

```
CONCEPT_RULE:(DIST_8, "_c{FULLNAME}",  
                "Harvard University")
```

3.6.1.E The ORDDIST_n Operator

Specify the order and distance between the terms or concepts that you want the selected concept to match. This operation locates and returns a match even when the usual contextual clues provided by adjacent matches are missing. For example, a match can be located when a name and position do not follow one another. The following example returns a match on the POSITION concept when it is followed by the word *Obama*. This is true only if the term *Obama* is located within 12 words from a match on the POSITION concept.

```
CONCEPT_RULE: (ORDIST_12, "_c{POSITION}", "Obama")
```

3.6.1.F The SENT Operator

Locate matches in the same sentence. For example, write a concept definition that locates a match for the term *Amazon* when the token *river* also occurs within the same sentence:

```
CONCEPT_RULE: (SENT, "_c{Amazon}", "river")
```

3.6.1.G The SENT_n Operator

Locate matches that occur in the specified number of sentences. For example, write a concept definition that locates matches for the PER concept and the term *he* within two sentences:

```
PER concept: CLASSIFIER:Obama  
CONCEPT_RULE: (SENT_2, "_c{PER}", "he")
```

3.6.1.H The SENTSTART_n Operator

Locate matches that occur within the specified number of words from the beginning of the sentence. For example, write a concept definition that locates matches for the term *Democratic* that occur within five words from the start of the sentence:

```
CONCEPT_RULE: (SENTSTART_5, "Democratic")
```

3.6.1.1 The SENTEND_n Operator

Locate matches that occur within the specified number of words from the end of the sentence. For example, write a definition that locates matches on a term in the PER concept if these matches occur within five words from the end of a sentence. The following example shows how the SENT_n, SENTSTART_n, and SENTEND_n qualifiers work together with a contextual operator and a classifier concept:

```
PER concept:  CLASSIFIER:Obama

CONCEPT_RULE:(SENT_2, (SENTSTART_5, "Democratic"),
                  (SENTEND_5, "_c{PER}"))
```

3.6.2 The Stemming Operator

When you add an @ symbol as a suffix to a word, you enable the expansion of the word into all of its forms. For example, if you append an @ sign to the word *book*, matches on books, booking, bookings, and so on, could be returned:

```
CONCEPT:book@
```

You can also append the @ sign followed by the letter N or the letter V to stem the word into all of its noun or verb forms, respectively. See the example below:

```
CONCEPT_RULE:(SENT, "_c{book@N}", "train@V")
```

Note: The @ symbol cannot be used in CLASSIFIER and REGEX definitions.

3.6.3 The PARA Operator

When you add the paragraph (PARA) operator, you specify that matches are located only within one paragraph. Determine the paragraph boundaries by typing one or more separator characters into the **Paragraph Separator** field in the **Project Settings - Misc** tab. When you specify more than one type of paragraph separator, use a comma (,) to identify each string as a paragraph separator. For example, you can enter the following strings to specify the paragraph separators \n\n, \t\t, <P>.

You can then write one of the following rules that specify that matches can be located only in the text bounded by one or more of these separators:

```
CONCEPT_RULE: (PARA, "_c{SAS}", (OR, "statistics", "TM"))
CONCEPT_RULE: (PARA, "_c{TM}", (OR, "Enterprise Miner"))
```

3.6.4 The Operators for Coreference Resolution

Coreference resolution enables you to match pronouns and other words to the canonical forms that these terms reference. (This is also known as *anaphora resolution*.) When you use coreference resolution, you can specify the canonical form of the referencing word. For example, specify *Barack*, *Obama*, and *President* as referring terms for the canonical form *Barack Obama*. Alternatively, choose to make *President Barack Obama* the canonical form for these terms.

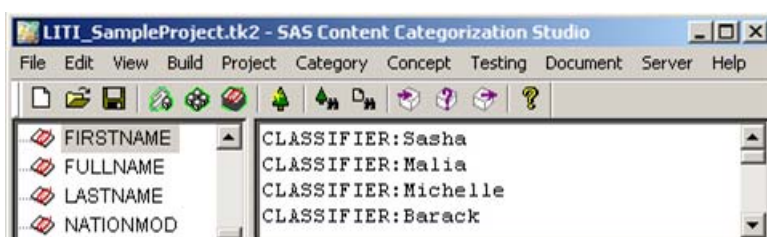
For more information about coreference operators, see Section 3.9 *The Coreference Operators* on page 72.

3.7 Contextual Extraction Concept Definition Examples

3.7.1 The Classifiers

Specify a `CLASSIFIER` rule to match one string, or dictionary entry. Like classifier definitions in SAS Content Categorization Studio, these rules specify a string to match in an incoming document. Unlike classifier concepts, each `CLASSIFIER` line is one `CLASSIFIER` rule.

Display 3-1 Classifier Rules



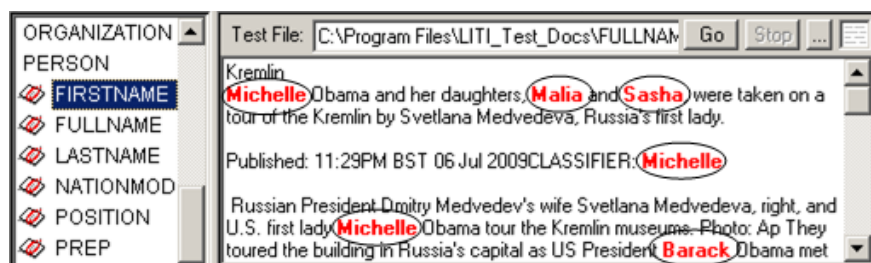
This FIRSTNAME concept consists of several CLASSIFIER rules.

Example 3-1: Matching a Sequence of Dictionary Entries

Concept Name	Entry
FIRSTNAME	CLASSIFIER:Sasha
	CLASSIFIER:Malia
	CLASSIFIER:Michelle
	CLASSIFIER:Barack

The FIRSTNAME concept matches any of the names to the right of the CLASSIFIER specifications in incoming texts. For example, any occurrence of Sasha, Malia, Michelle, or Barack, is a match.

Figure 3-1 FIRSTNAME matches in an Input Document

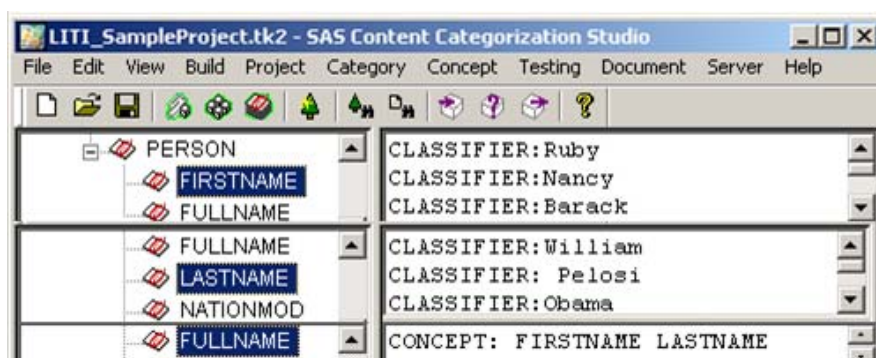


Note: You can also specify a returned information string after a comma (,). In this case the returned information is the value for the matched concept. For more information, see *SAS Content Categorization Studio: User's Guide*.

3.7.2 Specifying a Sequence of Classifier Entries

Write a **CONCEPT** rule to identify related information, whether these relationships are known beforehand. For example, you might want to identify all of the lakes in the state of Michigan, but not know the names of these lakes when you write the rule. The **CONCEPT** definition specifies the ordering of **CLASSIFIER** concepts. A match occurs when matching **CLASSIFIER** strings are located in the specified order in an input document.

Figure 3-2 *CONCEPT* Rule



This **FULLNAME** concept defines a relationship between the **FIRSTNAME** and **LASTNAME** concepts.

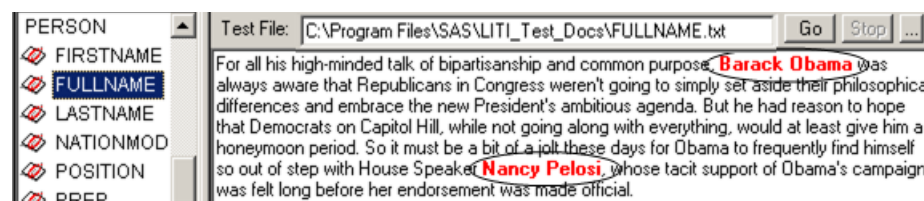
Example 3-2: *Matching a Sequence of Dictionary Entries*

Concept Name	Entry
FIRSTNAME	CLASSIFIER:Ruby
	CLASSIFIER:Nancy
	CLASSIFIER:Barack
LASTNAME	CLASSIFIER:William
	CLASSIFIER: Pelosi
	CLASSIFIER:Obama
FULLNAME	CONCEPT: FIRSTNAME LASTNAME

The **FULLNAME** concept uses the lists of terms that are specified by the **CLASSIFIER** definitions in the **FIRSTNAME** and **LASTNAME** concepts. A relationship between matches on the **FIRSTNAME** and **LASTNAME** concepts

is specified by the FULLNAME concept. For example, the terms *Nancy Pelosi* and *Barack Obama* match in an input document for both the FIRSTNAME and the LASTNAME concepts. These matches are also a match for the FULLNAME concept rule.

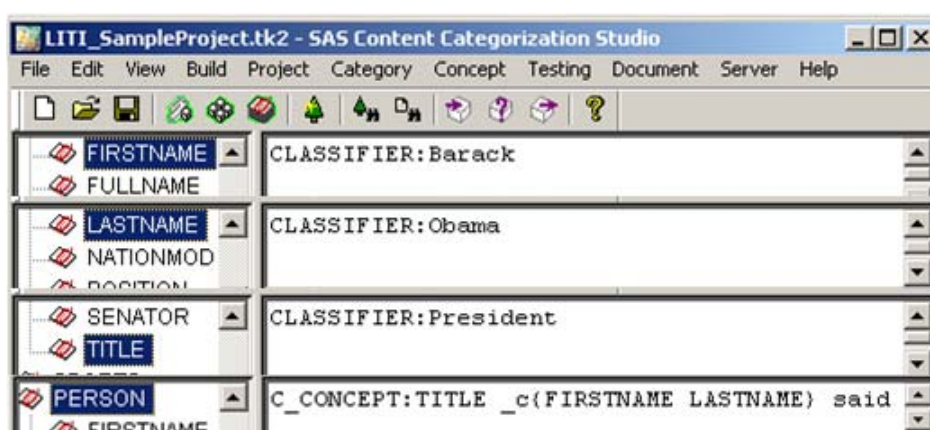
Figure 3-3 FULLNAME Concept Matches in an Input Document



3.7.3 Context Matching

Write a C_CONCEPT rule to match text in an input document based on the context of the matches. You can also use tokens with C_CONCEPT rules.

Figure 3-4 C_CONCEPT Rule



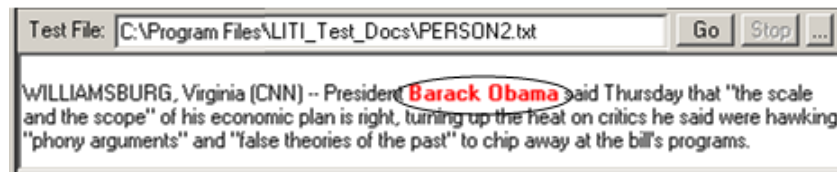
This C_CONCEPT rule specifies a relationship between the CLASSIFIER concept rules and the token *said*.

Example 3-3: Matching within Context

Concept Name	Entry
FIRSTNAME	CLASSIFIER:Barack
LASTNAME	CLASSIFIER:Obama
TITLE	CLASSIFIER:President
PERSON	C_CONCEPT:TITLE _c{FIRSTNAME LASTNAME} said

The PERSON concept locates matches for the FIRSTNAME and LASTNAME concepts. These matches occur in the context (_c) specified by the curly braces ({}) preceded by a match on the TITLE concept and followed by the token *said*. In this example, *Barack Obama* matches on the PERSON concept.

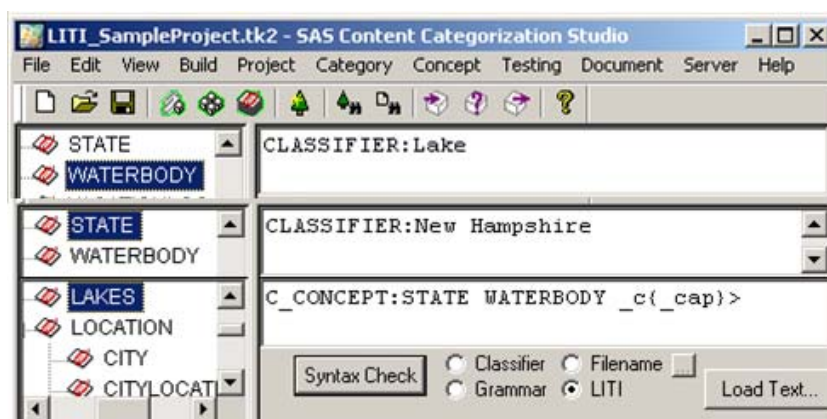
Figure 3-5 A C_CONCEPT Match in an Input Document



3.7.4 Matching within Context

Write a C_CONCEPT definition to locate and match a word that you do not know until a match on this definition is located. However, you should know the context for this match. For example, you might want to locate, and return each duplicate instance of *New Hampshire lake* in an input text.

Figure 3-6 C_CONCEPT Rule



This C_CONCEPT definition specifies a relationship between matching concepts and a word beginning with an uppercase letter.

Example 3-4: Using a Reference for a Match

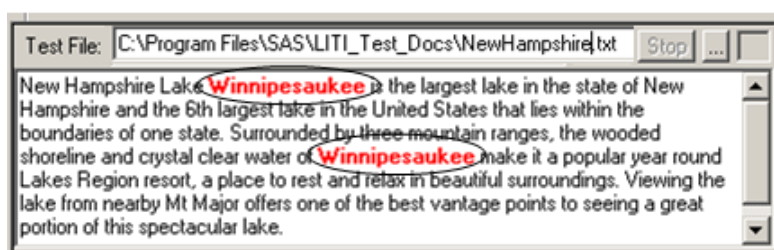
Concept Name	Entry
WATERBODY	CLASSIFIER:Lake
STATE	CLASSIFIER:New Hampshire
LAKES	C_CONCEPT:STATE WATERBODY _c{_cap}>

The LAKES concept specifies the context for the matched terms:

- When a match on the STATE concept is followed by a match on the WATERBODY concept, a partial match is located. For example, *New Hampshire Lake* is a partial match for this rule.
- `_c(_cap)` specifies that the matches above also appear in the context of a word that begins with an uppercase letter. In this example, a match occurs on the word *Winnepesaukee*.
- By default, all of the matches in an input document are returned. When the greater than (>) symbol is specified, every instance of the matched term in the document is returned as a match regardless of the context.

In this example, all instances of *Winnepesaukee* are matched. The second match occurs because the greater than (>) symbol is specified.

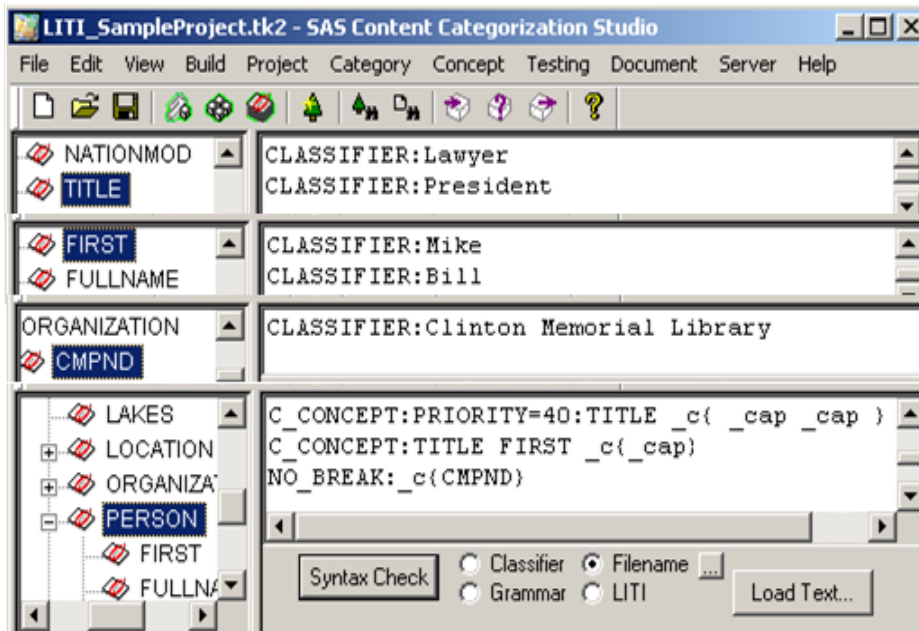
Figure 3-7 C_CONCEPT Matches in an Input Document



3.7.5 Eliminating Partial Matches

Specify a `NO_BREAK` rule to prevent partial matches on terms. This rule stipulates that a match can occur only if the entire string is located in an input document. This statement is true for any rules that might otherwise locate a partial match.

Figure 3-8 `NO_BREAK` Rule



The PERSON concept specifies the `NO_BREAK` rule.

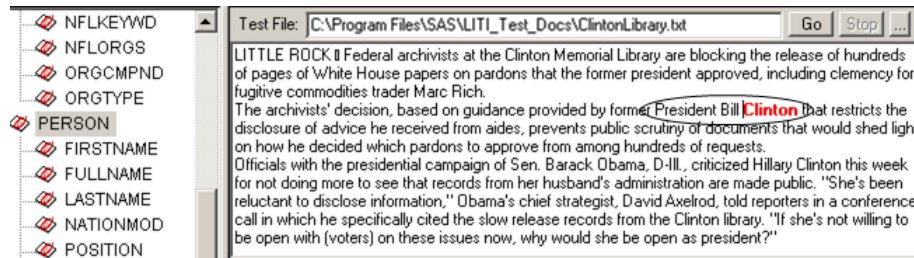
Example 3-5: Excluding Spaces

Concept Name	Entry
TITLE	CLASSIFIER:President
FIRST	CLASSIFIER:Bill
CMPND	CLASSIFIER:Clinton Memorial Library
PERSON	C_CONCEPT:TITLE FIRST _c{ _cap }

NO_BREAK : _c {CMPND}

When you add the NO_BREAK rule to the PERSON concept definition, the token Clinton is not matched when it occurs in the phrase *Clinton Memorial Library*. For this reason, matches are not returned for any definition that matches part, but not all, of this term.

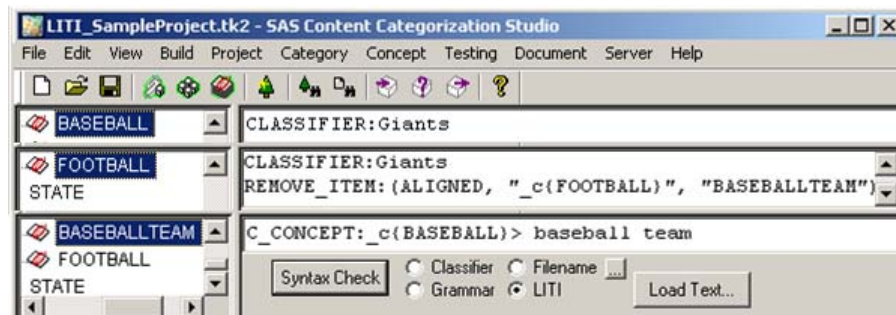
Figure 3-9 NO_BREAK Rule Match in an Input Document



3.7.6 Disambiguating Matches

REMOVE_ITEM definitions differentiate between matches according to their context. This process of differentiation is called *disambiguation*. In SAS Content Categorization Studio disambiguation is specified in a Boolean definition using the __TGIF or __TGUNLESS operator. SAS Contextual Extraction Studio enables you to specify this rule type when you refer to other concepts by writing a REMOVE_ITEM rule. Use this operation to eliminate a match on one rule, while returning a match on another rule.

Figure 3-10 REMOVE ITEM Rule



The FOOTBALL concept definition includes the REMOVE_ITEM rule to prevent Giants football documents from matching the Giants baseball concept.

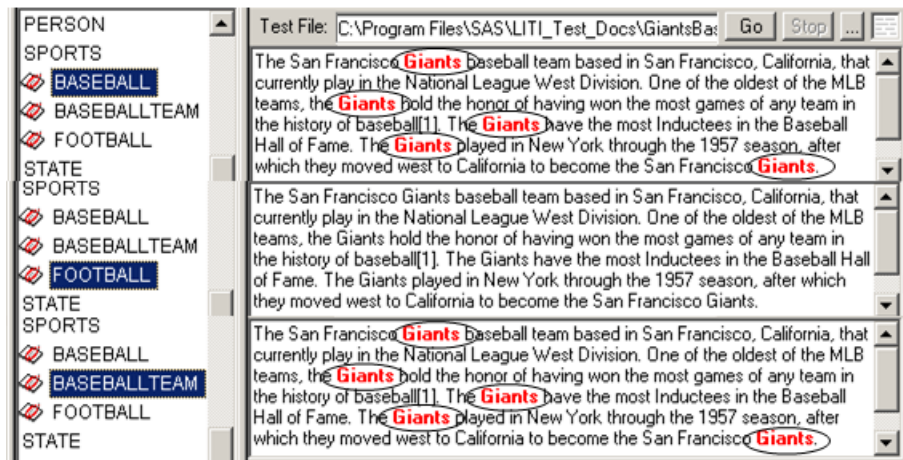
Example 3-6: Excluding Phrases

Concept Name	Entry
BASEBALL	CLASSIFIER:Giants
FOOTBALL	CLASSIFIER:Giants REMOVE_ITEM: (ALIGNED, "_c{FOOTBALL}", "BASEBALLTEAM")
BASEBALLTEAM	C_CONCEPT:_c{BASEBALL} baseball team

Matches on the word *Giants* are returned for the BASEBALLTEAM concept when the token *Giants* is located in the specified context, Giants baseball team. In this case, this match is not a match for the FOOTBALL concept. The REMOVE_ITEM rule specifies that any match on both the BASEBALLTEAM and

the FOOTBALL concepts return only matches for the BASEBALLTEAM concept.

Figure 3-11 Disambiguated Matches in Input Documents



3.7.7 Exporting Classifiers

The `CONCEPT` rule enables you to export previously unspecified classifier terms to another concept using an acronym that is specified in a concept rule. For example, specify `eLN` for last name. Alternatively, you can type the full name of the concept, `LASTNAME`.

To write a rule using an acronym, specify this acronym in the destination rule. After an acronym is specified in a `CONCEPT` rule, other rules can specify this acronym to list the exported term.

The `CLASSIFIER` rule that specifies the export feature enables you to match incomplete terms in ways that are similar to that of the greater than symbol. For more information, see Section 3.5.8 *The > Symbol* on page 23. However, you can use only the export operation with `CLASSIFIER` rules.

Figure 3-12 *CLASSIFIER Rule with Export Feature*



The `FULLNAME` concept specifies a `CLASSIFIER` rule that exports matches on `Sarkozy` to the `PERSON` concept that has a `CONCEPT` rule specifying `eLN`. This rule also specifies its own matching string and the context for matches.

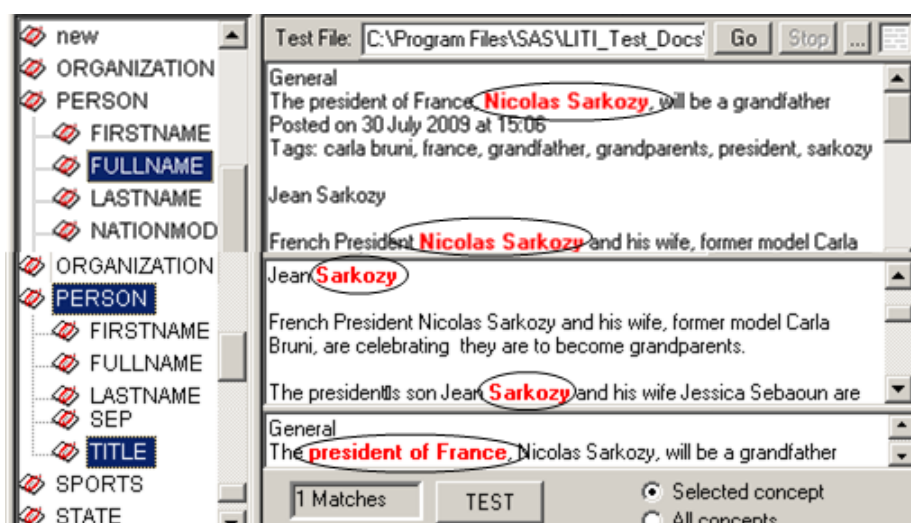
Example 3-7: *Exporting Classifiers Example 1*

Concept Name	Entry
FULLNAME	CLASSIFIER:[export=TITLE:president of France; eLN:Sarkozy]:Nicolas Sarkozy
PERSON	CONCEPT:eLN

The following matches occur in an input text that has the words *Nicolas Sarkozy* and *President of France* present somewhere in the same document:

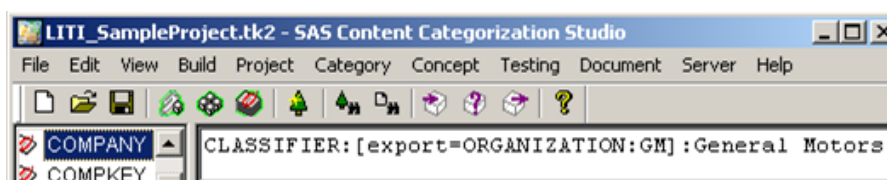
- *President of France* is exported to, and matches, the TITLE concept.
- *Sarkozy* matches the PERSON concept. This match occurs because the acronym *eLN* is specified in the PERSON concept.
- *Nicolas Sarkozy* is returned as the match for the FULLNAME concept.

Figure 3-13 Classifier and Export Matches in Input Documents



The export feature works on an internal, per-document basis. In this example, the terms *President of France* and *Sarkozy* only match the TITLE and PERSON concepts if *Nicolas Sarkozy* is present in the input document. The exported terms do not appear in the concept definitions when these terms are exported. The concepts do not have to exist in the taxonomy in order for the export rule to work.

Figure 3-14 CLASSIFIER Rule with Export Feature



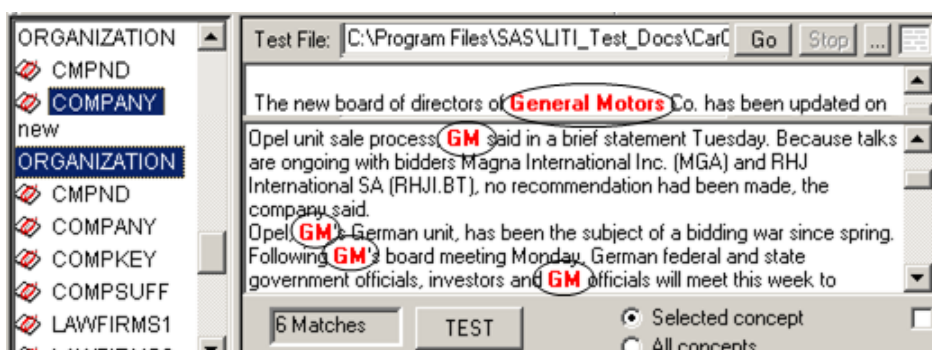
The COMPANY concept specifies that a match on GM is exported to the ORGANIZATION concept.

Example 3-8: Exporting Classifiers Example 2

```
COMPANY                                CLASSIFIER:[export=ORGANIZATION: GM]:  
General Motors
```

If an input text contains the string *General Motors*, the document matches the COMPANY concept. If this document also contains the word *GM*, the token *GM* is recognized as a match on the ORGANIZATION concept. However, if the word *GM* appears in a document without the term *General Motors*, *GM* is not returned as a match to the ORGANIZATION concept.

Figure 3-15 Export Rule Matches in Input Documents



3.7.8 Setting Priorities for Overlapping Matches

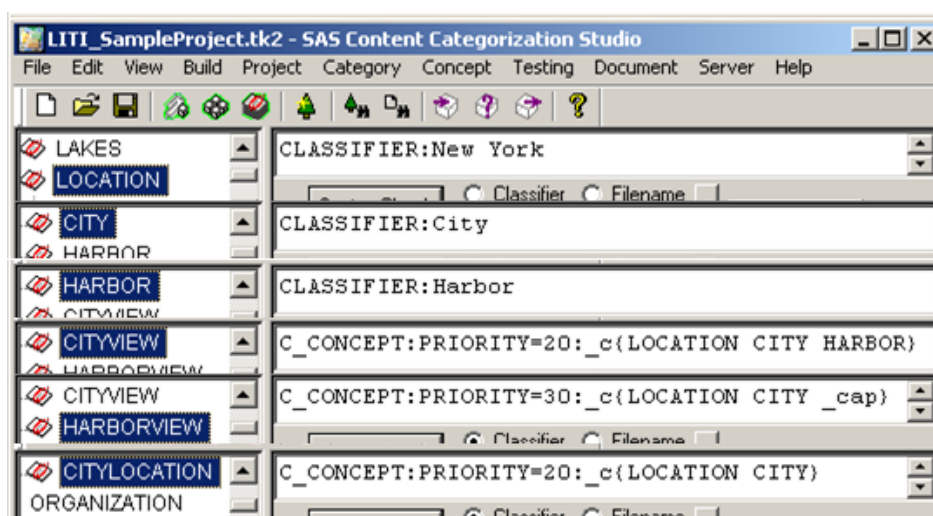
SAS Contextual Extraction Studio enables you to override the **Priority** setting in the Data window for the selected contextual extraction concept. This feature works with `CONCEPT_RULE` definitions and coreference rules when you write a `PRIORITY` specification into a rule. For more information about coreference, see Section 3.9.7 *Rank Coreference Definitions and Eliminate False Positives* on page 79.

To use this feature, select **Best Matches** in the **LITI** tab of the Project Settings window. By default, the **Priority** is set to 10 in the Data window for contextual extraction concepts. (However, this setting is applied to any SAS Content Categorization Studio concepts that you write when you upload a contextual extraction project as a binary file.)

You can also increase the **Priority** setting in the Data window for all of the rules in one definition, or specify a `PRIORITY` in a contextual extraction concept rule. When you specify a `PRIORITY` in a rule, this setting overrides the **Priority** setting in the Data window—for this rule only. The `PRIORITY` specification in a rule applies to the rule, and not to the entire definition. For this reason, any matches on this rule are prioritized over matches on any other rules in this definition, or in any other definitions

These specifications are used to increase the relative rankings between contextual extraction concepts. Priorities are also used to prevent matches on more than one concept. You can also use this setting to prevent matches on terms that are used in different contexts. For example, if `Roche` is specified in the `PERSON` concept and also in the `CORPORATE` concept, priorities can be used to determine the appropriate match.

Figure 3-16 C_CONCEPT Rule Specifying a Priority Setting



The HARBORVIEW concept has the highest PRIORITY setting. Documents that match this concept, and any of the other concepts shown, are matched to the HARBORVIEW concept.

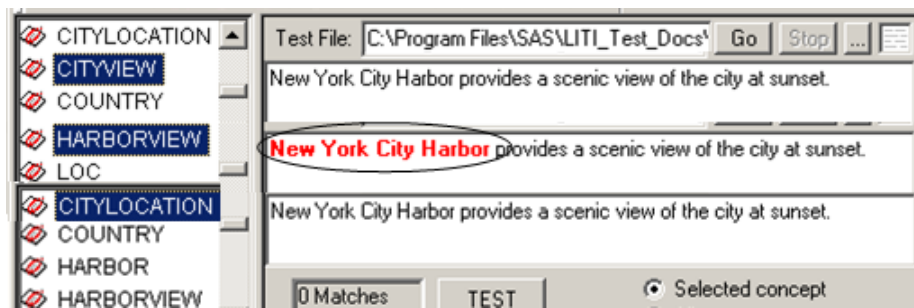
Example 3-9: Setting Priorities

Concept Name	Entry
LOCATION	CLASSIFIER:New York
CITY	CLASSIFIER:City
HARBOR	CLASSIFIER:Harbor
CITYVIEW	C_CONCEPT:PRIORITY=20:_c{LOCATION CITY HARBOR}
HARBORVIEW	C_CONCEPT:PRIORITY=30:_c{LOCATION CITY _cap}
CITYLOCATION	C_CONCEPT:PRIORITY=25:_c{LOCATION CITY}

The following document is returned as a match to the HARBORVIEW concept. This is true even though *New York City Harbor* also matches the CITYVIEW concept and part of this term matches CITYLOCATION.

New York City Harbor provides a scenic view of the city at sunset.

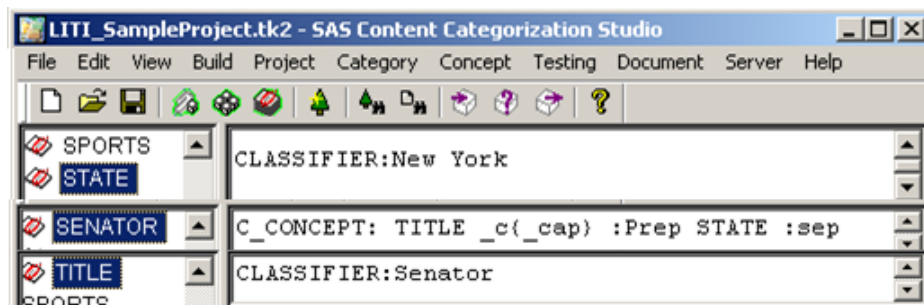
Figure 3-17 A Prioritized Match in an Input Document



3.7.9 Specifying Part-of-Speech Tags

Like SAS Content Categorization Studio, SAS Contextual Extraction Studio enables you to use part-of-speech tags to locate matches. These tags are useful when you want to locate a wide range of matches without specifying a list of dictionary entries. Part-of-speech tags are particularly useful when you know the syntax, but not the wording of, the exact matches that you are seeking.

Figure 3-18 C-CONCEPT with Part-of-Speech Tags



A space is required before the colon (:) that precedes the part-of-speech tag. Specify a lowercase *s* in the *sep* part-of-speech tag.

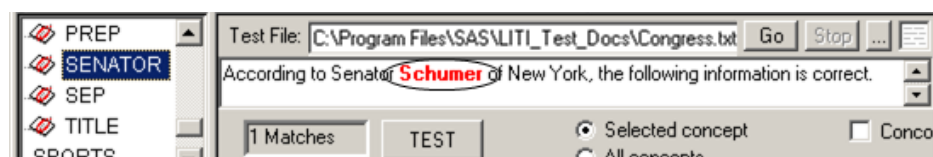
Example 3-10: Using Part-of-Speech Tags

Concept Name	Entry
STATE	CLASSIFIER:New York
TITLE	CLASSIFIER:Senator
SENATOR	C_CONCEPT: TITLE _c{_cap} :Prep STATE :sep

Schumer is returned as a match for the SENATOR concept when a preposition (*Prep*) precedes a match on the CITY CLASSIFIER concept and a separator (*sep*) character follows this concept. See the following example:

According to Senator *Schumer* of New York, the following information is correct.

Figure 3-19 A C_CONCEPT Rule with Part-of-Speech Tag Match in an Input Document



3.7.10 Specifying Regular Expressions

Specify regular expressions to locate matches based on known patterns. For example, telephone numbers, street, and e-mail addresses are all defined using recognizable patterns. When you write regular expressions, you specify a range of letters or numbers inside square braces ([]) to form a regular expression rule. For example, type `a-z` or `0-9`. This syntax matches any ASCII character whose value is between a and z or between 0 and 9 inclusive.

If you add a plus (+) sign after the last brace, all lowercase letters are matched. For example, you could write `REGEX: [a-z]+`.

You can also add either the % symbol or write out the word `percent`. If you do this after you add the plus (+) symbol all of the instances of percentages in the input document are returned.

Figure 3-20 Regular Expression Rules



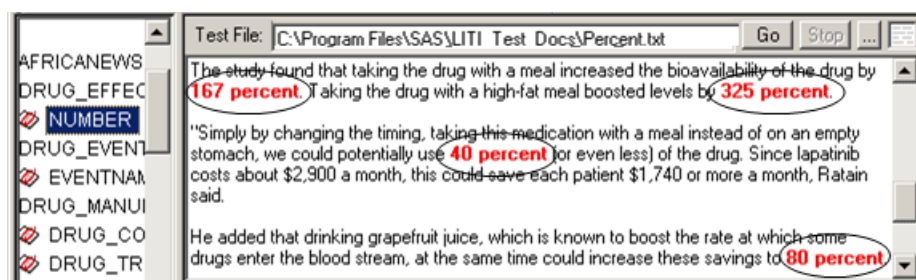
The NUMBER concept has REGEX rules. The different specifications for percent ensure wider definition coverage.

Example 3-11: Specifying Regular Expressions

Concept Name	Entry
NUMBER	REGEX: [0-9]+%
	REGEX: [0-9]+ percent

This regular expression definition specifies that numbers followed by either percentage sign match. For example, matches on both 99%, and 50 percent are both returned.

Figure 3-21 REGEX Rule Matches in an Input Document

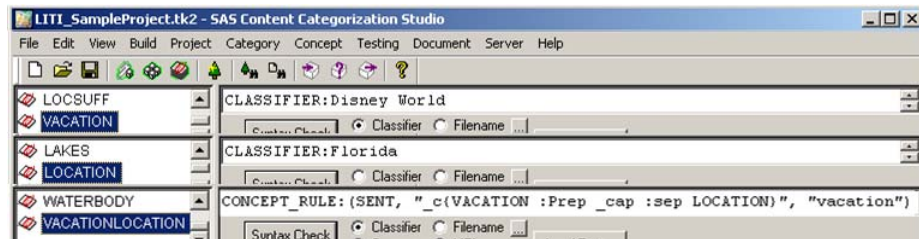


Notes: For more information, see Appendix A: *Using the Directive and Regex Syntax on page 89*.
You can also specify a returned information string after a comma (,). In this case, the returned information is the value for the matched concept. For more information, see *SAS Content Categorization Studio: User's Guide*.

3.7.11 Specifying a Sentence Operator

By default, SAS Content Categorization Studio looks for matches within the entire text of an input document. Limit matches to one sentence by writing the SENT operator into a CONCEPT_RULE.

Figure 3-22 CONCEPT_RULE with a Sentence Operator



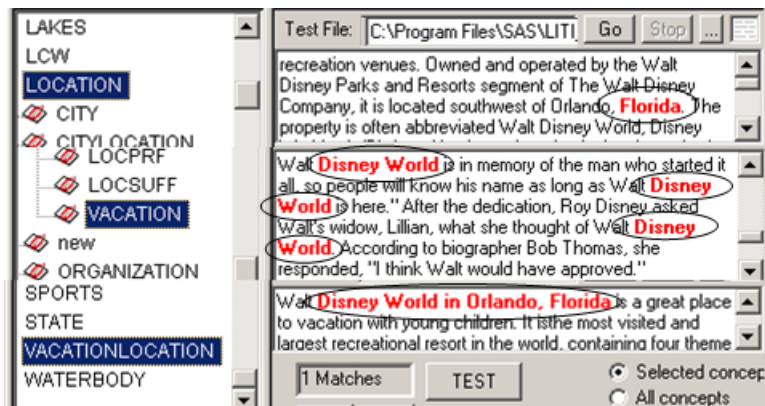
The VACATIONLOCATION concept specifies that a match is returned only when all of the specified elements are located in the context of a sentence.

Example 3-12: Specifying a Sentence Operator

Concept Name	Entry
VACATION	CLASSIFIER:Disney World
LOCATION	CLASSIFIER:Florida
VACATIONLOCATION	CONCEPT_RULE:(SENT, \"_c{VACATION :Prep_cap :sep LOCATION}\", \"vacation\")

The VACATIONLOCATION definition uses the CONCEPT_RULE to identify a match, when all of the specified components occur within one sentence. These matches occur when a preposition follows a VACATION concept match, a word that begins with an uppercase letter, a separator character, and a match on the LOCATION concept. If this match is followed by a match on the token vacation, a match is returned for the VACATIONLOCATION concept.

Figure 3-23 CLASSIFIER and CONCEPT_RULE Matches



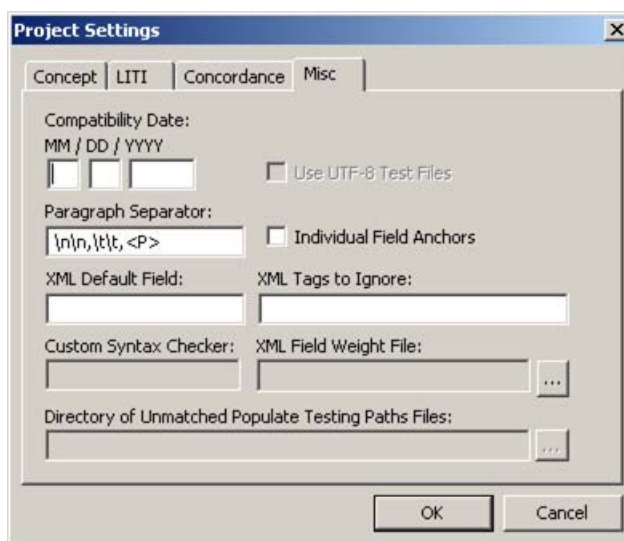
3.7.12 Specifying a Paragraph Operator

By default, SAS Content Categorization Studio looks for matches within the entire text of an input document. Limit matches to one paragraph by writing the `PARA` operator into the `CONCEPT_RULE`.

Before you specify your concept definitions, specify the paragraph separator that is used in your documents. For example, specify `<p>` for `.html` documents. If you are using multiple types of documents, list the paragraph separator for each type.

To specify the paragraph separator, complete these steps:

1. Select **Project --> Settings**. The Project Settings window appears.



2. Type the paragraph separators for your input documents into the **Paragraph Separator** field. For example, enter `\n\n,\t\t,<P>`.
3. Click **OK**.

After you specify your paragraph operator, or operators, you can specify the rules for each concept.

Display 3-2 CONCEPT_RULEs with a Paragraph Operator



The `PARA` operator specifies that a match is returned only when all of the specified elements are located in the context of a paragraph. Each paragraph is delineated by one of these paragraph markers.

Example 3-13: Specifying Paragraph Operators

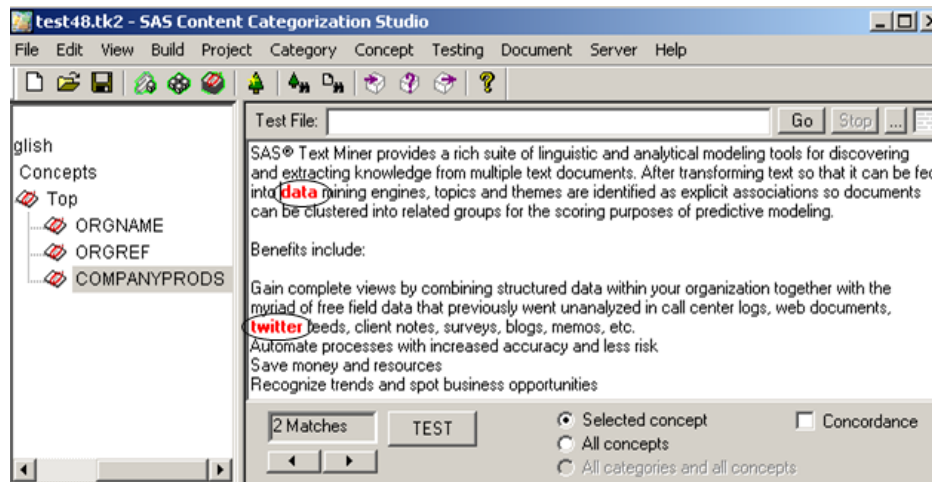
Concept Name	Entry
COMPANYPRODS	<code>CONCEPT_RULE:(PARA, "_c{data}", {OR, "date", "engines"})</code> <code>CONCEPT_RULE:(PARA, "_c{twitter}", {OR, "feeds"})</code>

The `COMPANYPRODS` definition uses `CONCEPT_RULE` definitions to identify matches within different paragraphs:

In the first case, a match occurs when *data* and either *date* or *engines* appear in the same paragraph.

In the second case, a match occurs when either *twitter* or *feeds* occur within the same paragraph.

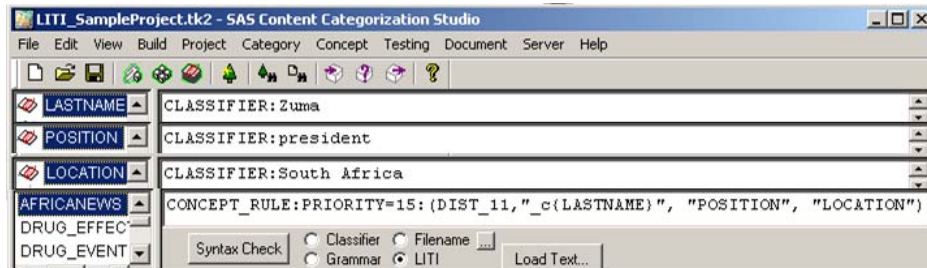
Figure 3-24 CONCEPT_RULE and Paragraph Matches



3.7.13 Specifying a DIST Operator

Specify the maximum number of words in which matches can be located, instead of using the default behavior to search the entire document. The distance (`DIST_n`) operator for `CONCEPT_RULE` enables you to specify the maximum number of words that can occur between matches on the first and the last term. However, this operator does not specify the ordering of the matches.

Figure 3-25 *CONCEPT RULE with a Distance Specification*



The AFRICANEWS definition specifies that a match is returned if there are no more than 11 words between a match on the LASTNAME and LOCATION concepts.

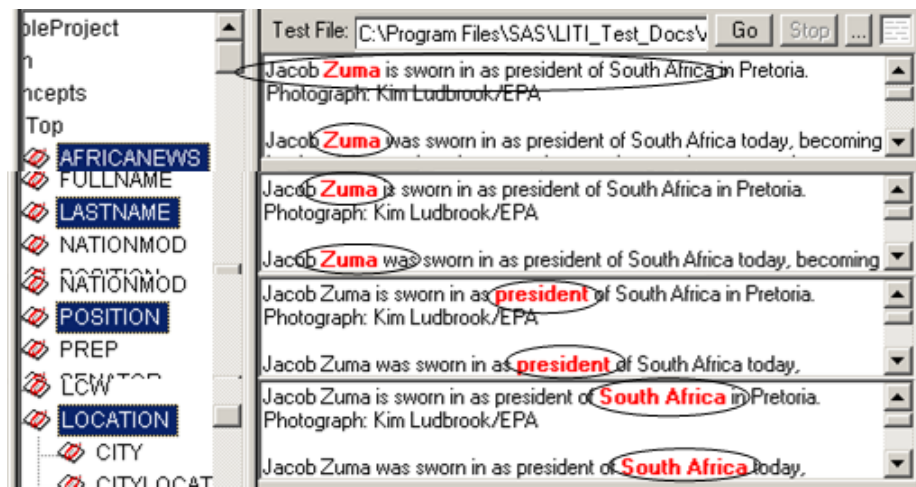
Example 3-14: *Specifying the DIST Operator*

Concept Name	Entry
LASTNAME	CLASSIFIER: Zuma
POSITION	CLASSIFIER: president
LOCATION	CLASSIFIER: South Africa
AFRICANEWS	CONCEPT_RULE: PRIORITY=15: (DIST_11, "_c{LASTNAME}", "POSITION", "LOCATION")

The AFRICANEWS concept uses the `DIST` operator to specify a distance of 11 words between the location of a match on the LASTNAME concept and the LOCATION concept. This match is returned if there is also a match on the POSITION concept within these 11 words. In addition, this `CONCEPT_RULE` overrides the default **Priority** setting in the Data window. If there were other

rules in this definition, these rules would keep the same priority setting specified in the Data window.

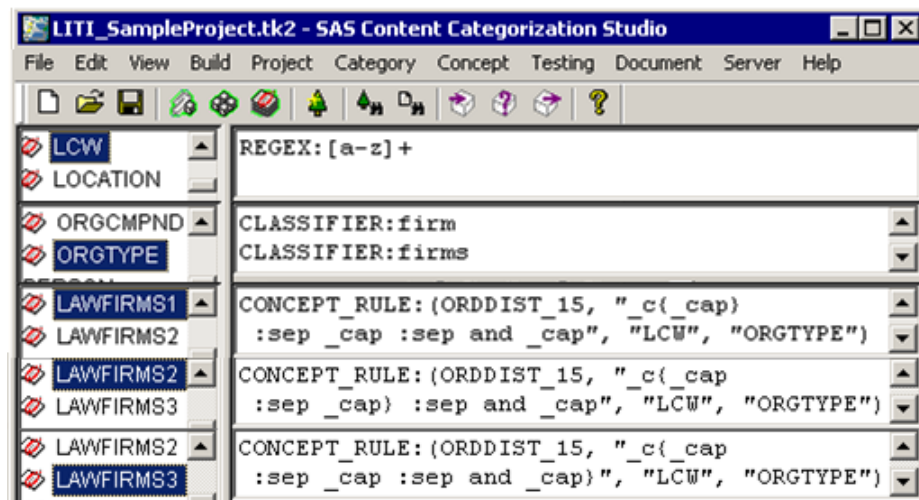
Figure 3-26 CONCEPT_RULE and CLASSIFIER Matches in Input Documents



3.7.14 Specifying an ORDDIST Operator

The ORDDIST_n operator is similar to the DIST operator. However, the ORDDIST operator specifies the order and distance requirements that are necessary to return a match on the CONCEPT_RULE definition.

Figure 3-27 CONCEPT_RULE with ORDIST Operator:



The CONCEPT_RULE for each LAWFIRMS concept places the ending curly brace (}) in a different location to return different results from the same input document.

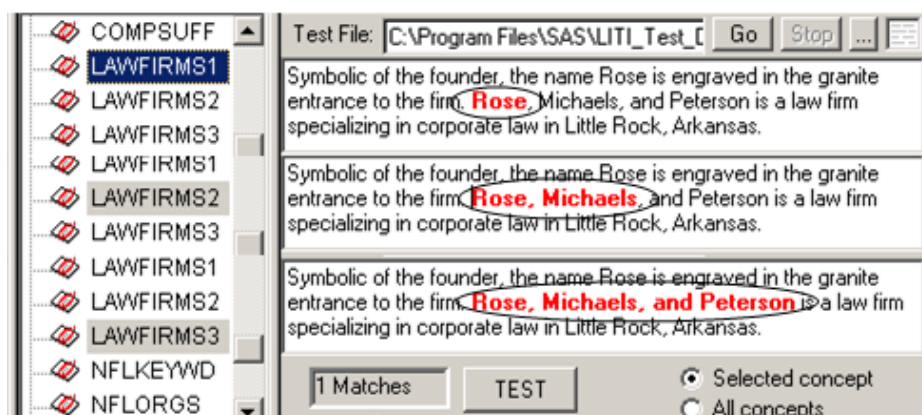
Example 3-15: Exporting Classifiers

Concept Name	Entry
LCW	REGEX:[a-z] +
ORGTYPE	CLASSIFIER:firm CLASSIFIER:firms
LAWFIRMS1	CONCEPT_RULE:(ORDDIST_15, "_c{ _cap} :sep _cap :sep and _cap", "LCW", "ORGTYPE")

This `CONCEPT_RULE` states that the following instances return a match if the matches occur in the specified order and within a distance of 15 words. A word begins with an uppercase letter and is followed by a separator character and an uppercase letter. This match is followed by a separator character, the token `and`, and another word beginning with an uppercase letter. The match is not returned unless the `LCW_REGEX` rule is also matched and a match on the `ORGTTYPE` concept also occurs within 15 words.

When the closing curly brace (`}`) is moved for the `LAWFIRMS2` and `LAWFIRMS3` concepts, the following matches are returned.

Figure 3-28 *CONCEPT_RULE Matches in Input Documents*



You can also change the default **Priority** setting of 10 in the Data window for any of the concept definitions shown above.

Display 3-3 Project Settings - LITI Settings

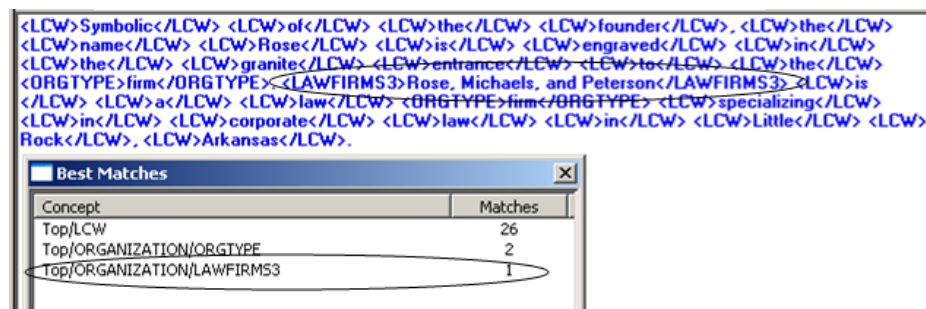


Use the Project Settings to affect how matches are returned:

- Select **All matches** and all of the matches for LAWFIRMS1, LAWFIRMS2, and LAWFIRMS3 above are returned. In this case, because the greater than (>) symbol does not end any of the CONCEPT_RULE definitions, only one match is returned for each concept.
- Select **Longest** and a match on LAWFIRMS3, only, is returned.
- Select **Best** and a match LAWFIRMS3 is returned. This is true unless you specify a higher priority in either the Data window or within a concept definition.

See the example shown below that applies to both the **Longest** and **Best** selections:

Figure 3-29 Best Matches Window



-
- Select **Return all identical matches**, if either **Longest** or **Best** is matched, and all of the instances with the same priority or length are returned.
 - The **Remove duplicate facts** operation does not apply. No facts can be specified for `CONCEPT_RULE` definitions.

3.8 Locating Facts

3.8.1 Overview of Facts

Facts, or predicates, refer to terms that match at least two concepts. Facts consist of at least two arguments. For example, *Harry Truman was president of the United States* is a fact based on several arguments. These arguments are defined by the following concepts `NAME`, `TITLE`, and `COUNTRY`. The following matches *Harry Truman*, *president*, and *United States* are returned to these concepts. By specifying this type of rule, you also locate similar matches in input documents without rewriting your rules.

Both `SEQUENCE` and `PREDICATE_RULES` extract facts. `SEQUENCE` rules specify the order of the matches. `PREDICATE_RULES` use Boolean operators, but do not specify the ordering of any matches. For more information, see Section 3.8.2 *A Predicate Sequence Example* on page 63 and Section 3.8.3 *Predicate Examples* on page 66.

3.8.2 A Predicate Sequence Example

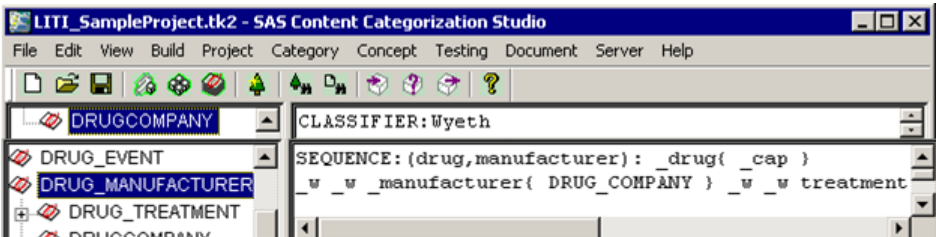
Identify previously unknown relationships, otherwise known as facts or events, in input documents. Predicate sequence, or `SEQUENCE`, rules extract the meaningful relationships between matched concepts and tokens. For example, identify the names and positions that various managers hold within a company. Locate this information even when these relationships are unknown to you, or when the concepts do not directly follow one another.

Predicates are also defined as facts or events. The terms are interchangeable. Facts are always defined by at least two concepts or tokens and one or more

parts of speech. The term *sequence* is used to specify the necessary ordering of the concepts and semantic terms that define these facts.

When you specify a predicate sequence definition, you define not only the concepts, but also the arguments that are used with these concepts. Use this rule to also specify the sequence of these entities and any appropriate parts of speech.

Figure 3-30 SEQUENCE Rule



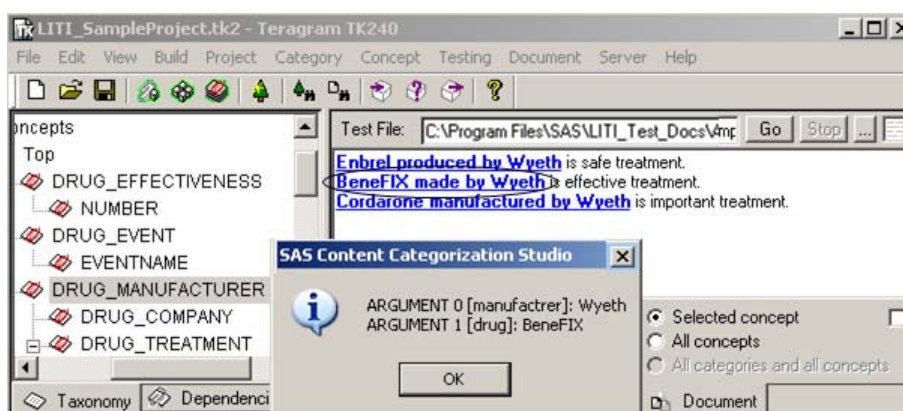
Example 3-16: Writing a Predicate Sequence Definition

Concept Name	Entry
DRUG_COMPANY	CLASSIFIER:Wyeth
DRUG_MANUFACTURER	SEQUENCE: (drug,manufacturer): _drug{ _cap } _w _w _manufacturer { DRUG_COMPANY } _w _w treatment

This SEQUENCE rule takes the arguments `drug` and `manufacturer`. To locate the `_drug` predicate, locate a word that begins with an uppercase letter that is followed by two tokens. To match the `_drug` predicate, locate the `DRUG_COMPANY` concept followed by two tokens and the word *treatment*. However, only the matches within and between the beginning and ending curly braces (`{}`) are returned as a match for this concept.

For example, the fact *BeneFIX produced by Wyeth* is returned as a match to the `DRUG_MANUFACTURER SEQUENCE` concept along with the matches on the arguments for this fact. You can see the fact matches in the Document window for this testing document. You can also click on one of the returned facts to open a SAS Contextual Extraction Studio status screen. This screen lists the matching arguments for the selected fact.

Figure 3-31 Argument Matches in an Input Document



SAS Content Categorization Server locates this fact and its arguments when you provide the .li file. In other words, you can decide to have SAS Content Categorization Server return all of the information shown in the example below. Alternatively, specify different returns. For example, choose to return only the fact, or only its arguments.

Example 3-17: SAS Content Categorization Server Output

```
FACT 0:[0(0)_4(24)]/DRUG_MANUFACTURER/: BeneFIX produced by  
Wyeth
```

```
ARG 0 [manufacturer]: Wyeth
```

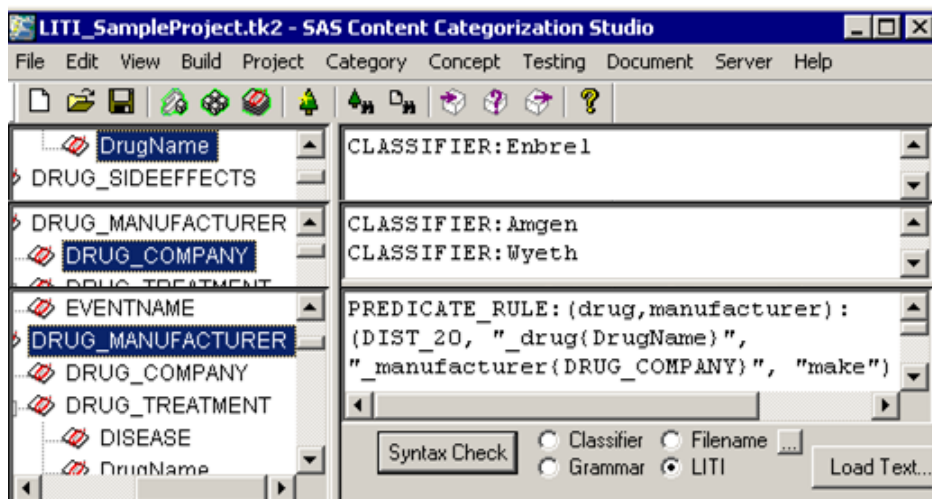
```
ARG [drug]: BeneFIX
```

Use the SAS Content Categorization Server Client API to specify fact matching strings, arguments, and the offsets returned by SAS Content Categorization Server.

3.8.3 Predicate Examples

Like `SEQUENCE` rules, `PREDICATE_RULES` locate facts and their supporting arguments. Unlike `SEQUENCE` rules, `PREDICATE_RULES` do not specify the matching order. Instead, `PREDICATE_RULES` use Boolean operators to increase the matching precision within the document. For more information, see Section 3.6 *The Operators* on page 29.

Figure 3-32 *PREDICATE_RULE with Logical Operators*



Like the preceding `SEQUENCE` rule, this `PREDICATE_RULE` defines the arguments `drug` and `manufacturer`. However, the `DRUG_MANUFACTURER` `PREDICATE_RULE` uses the `DIST` operator. This operator specifies that a match is returned when the `DrugName` concept is located within 20 words of a match on the `DRUG_COMPANY` concept.

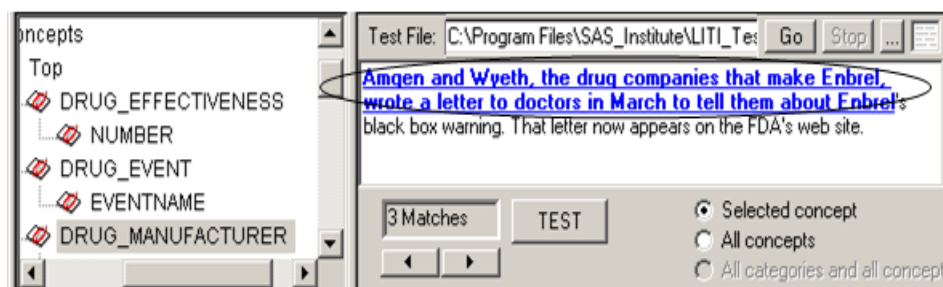
Example 3-18: Viewing a PREDICATE_RULE

Concept Name	Entry
DrugName	CLASSIFIER:Enbrel
DRUG_COMPANY	CLASSIFIER:Amgen CLASSIFIER:Wyeth
DRUG_MANUFACTURER	PREDICATE_RULE: (drug,manufacturer): (DIST_20, "_drug{ DrugName }", "_manufacturer{ DRUG_COMPANY }", "make")



This PREDICATE_RULE defines the arguments `drug` and `manufacturer`. Inside the parentheses that follow each argument is the concept that identifies a match. The `DIST` operator specifies that matches on the `DrugName` concept can occur within 20 words of a match on the `DRUG_COMPANY` concept. In addition, a match on the `DRUG_MANUFACTURER` concept only occurs when the token `make` is located. Although no other tokens are specified for this PREDICATE_RULE, all of the words located between matches on the concepts `DrugName` and `DRUG_COMPANY` are returned as a matching phrase. However, because a PREDICATE_RULE is specified and not a SEQUENCE rule, these matches can occur in any order.

For PREDICATE_RULES, like other definitions, multiple matches can occur in one document, and multiple facts can be returned.

Figure 3-33 PREDICATE_RULE Match in an Input Document



The results shown above are returned when the default setting, **All matches**, is selected under the **Overlapping Concept Matches** heading in the Project Settings - LITI dialog box.

Click  and  in the Document window to see each of the following matches:

Amgen and Wyeth, the drug companies that make Enbrel

This fact matches the word *Wyeth* as a token. It is not a match on the *DrugName* concept.

Wyeth, the drug companies that make Enbrel

This is the shortest of the matches that begin with a match on `Wyeth` in the `DRUG_COMPANY` concept and end with `Enbrel` as a match on the `DrugName` concept. Also see the following bulleted point.

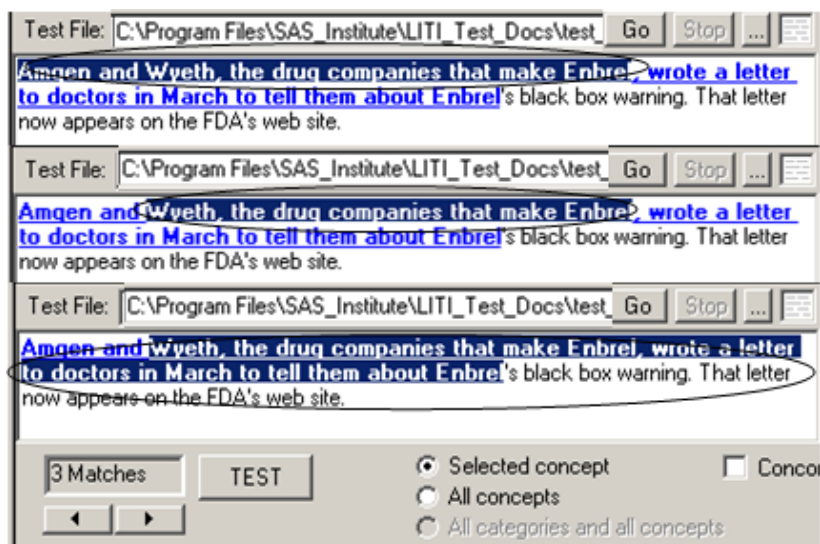
Wyeth, the drug companies that make Enbrel, wrote a letter to doctors in March to tell them about Enbrel

This is the longest of the matches that begin with a match on `Wyeth` in the `DRUG_COMPANY` concept and end with `Enbrel` as a match on the `DrugName` concept. In this case, the first instance of `Enbrel` is matched as a token and not as a match on the `DrugName` concept. Also see the bulleted point above.

This match is returned when you select **Longest** under the **Overlapping Concept Matches** heading in the Project Settings - LITI dialog box.

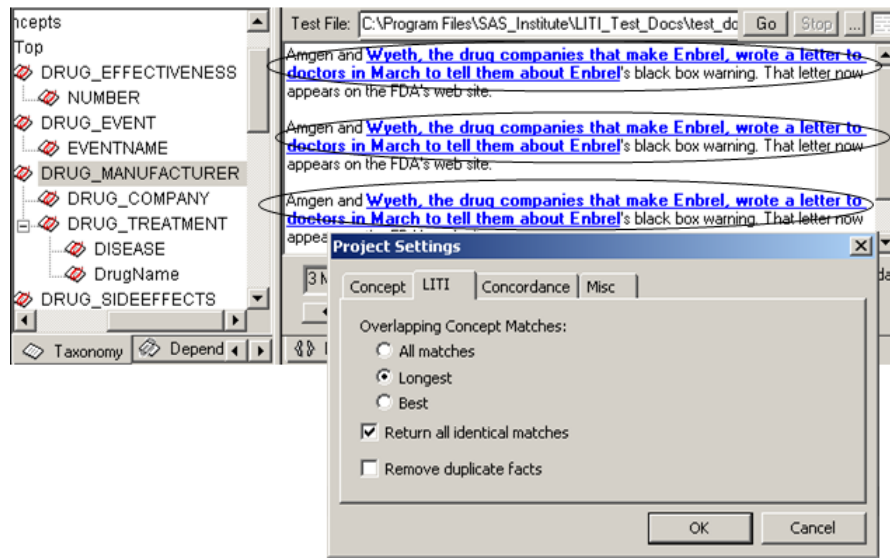
These results are also returned when you select **Best**. This statement is true unless you set a **Priority** specification in the **Definition** tab or overwrite the default setting of 10 in the Data window for this concept.

Figure 3-34 PREDICATE_RULE Matches in Input Documents



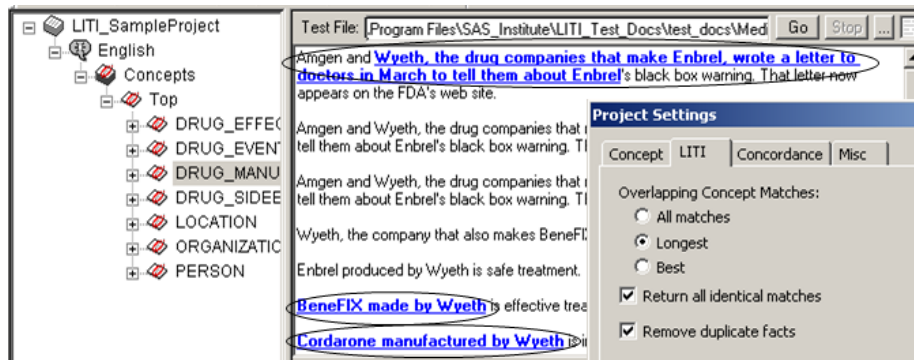
To return all of the instances of the longest fact matches, select **Return all identical matches** in the Project Settings - LITI dialog box. This operation can only be selected if you have also selected either **Longest** or **Best** under the **Overlapping Concept Matches** heading.

Figure 3-35 Several Instances of a Match in an Input Document



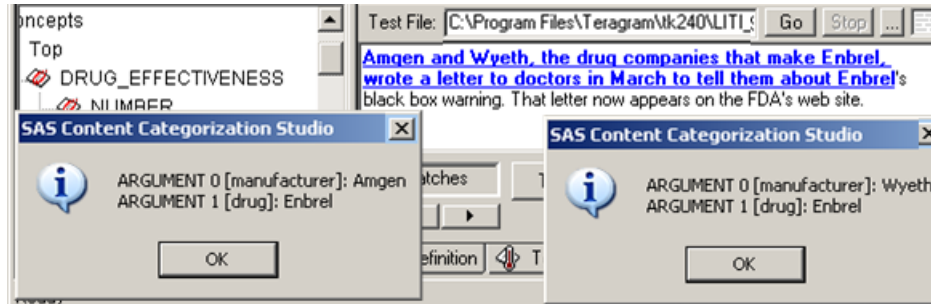
- In the figure below, **Remove duplicate facts** is added to the selections in the figure above. New text is added to the testing document to illustrate the functionality of these interrelated settings. Each instance of a match that is the longest for any of the overlapping matches, but not a duplicate fact, is returned as a match to the selected concept.

Figure 3-36 Longest Unique Matches in an Input Document



All of these facts are highlighted, and initially appear as a single match to the PREDICATE_RULE definition for the DRUG_MANUFACTURER concept. However, there are two sets of arguments, because there are two matches on the DRUG_COMPANY concept and one match on the DrugName concept. It is these matches that define the beginning and end of each fact.

Figure 3-37 Facts and Arguments in an Input Document



3.9 The Coreference Operators

3.9.1 Overview of Coreference

Use coreference operators to write rules that return the canonical form of a word along with the referring term. Coreference operators are often used with pronouns, or other words that are called *referring terms*. (This is also known as *anaphora resolution*.) The canonical form of a word can be any term that you choose. For example, return either *Barack Obama* or *President Barack Obama* as a match for each instance of the referring term *Barack* in an input document. Another alternative is to choose to return *President Barack Obama* as the canonical form for each match on the pronoun *he*.

When the tested document is displayed in the **Document** tab, both the canonical word form and the matching term are highlighted. This is because these matches are linked in SAS Content Categorization Studio.

Use the coreference operator (`_ref`) with a `CONCEPT`, `C_CONCEPT`, or a `CONCEPT_RULE` rule. If you want to use a coreference qualifier in a `CLASSIFIER` rule, use `_coref` instead of `_ref`.

Note: The **Overlapping Concept Matches** selections in the **LITI** tab of the Project Settings window do not affect matches made by the export, forward, and preceding operators.

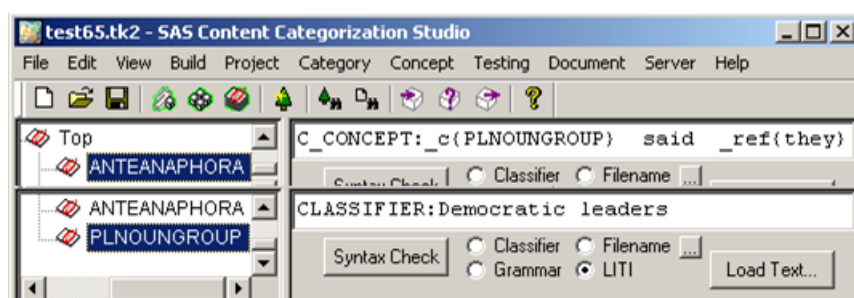
3.9.2 How to Use the Coreference Operator

Use the coreference operator (`_ref`) to link a matched string with its canonical form in an input document.

```
C_CONCEPT:{Jim Goodnight} said _ref{he}
```

In the example above, the canonical form *Jim Goodnight* is returned each time the matching term, *he* is located. This is true when the phrase *Jim Goodnight said he* is located in the text.

Figure 3-38 C_CONCEPT with _ref Operator.



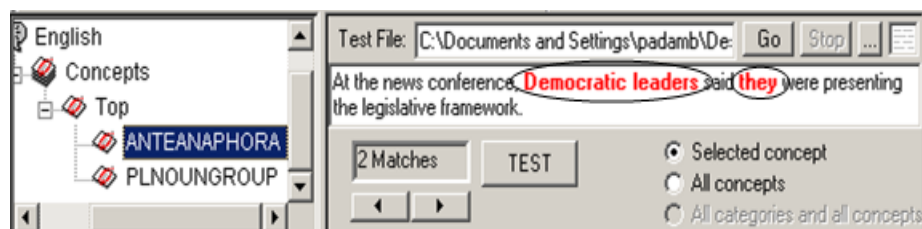
The `_c` operator is used in a `C_CONCEPT` rule that specifies the canonical form for the coreference specified by the `_ref` operator.

Example 3-19: C_CONCEPT Rule with the _ref Operator

Concept Name	Entry
PLNOUNGROUP	CLASSIFIER:Democratic leaders
PERSON	C_CONCEPT:_c{PLNOUNGROUP} said _ref{they}

When this definition is matched in an input document, a match on the referring term that follows the `_ref` operator returns the canonical form. The canonical form is specified in the bracketed term that follows the context operator (`_c`). This form is identified in the PLNOUNGROUP concept. In this example, the word that *they* references its specified canonical form *Democratic leaders*.

Figure 3-39 _ref Match in an Input Document



In this example, *Democratic leaders* and *they* are returned as matches in this input document. However, if the document contained other instances of the word *they*, these instances are not matched. You can see these matches in the Document window for this testing document.

3.9.3 How to Use the `_ref` Operator with the `>` Symbol

The greater than symbol (`>`) locates multiple instances of a match specified by the bracketed (`{}`) coreference operator (`_ref`) in an input document. For example, you might want to return the canonical form for each matched instance of a first name. In this case, you could specify a rule that identifies any references to *Jim* as a reference to *Jim Goodnight CEO of SAS Institute*. For more information, see Section 3.5.8 *The `>` Symbol* on page 23.

3.9.4 How to Use the `_ref` Operator with the Forward or Backward Symbols

3.9.4.A Limiting Matches to Those That Follow or Precede a Coreference Match

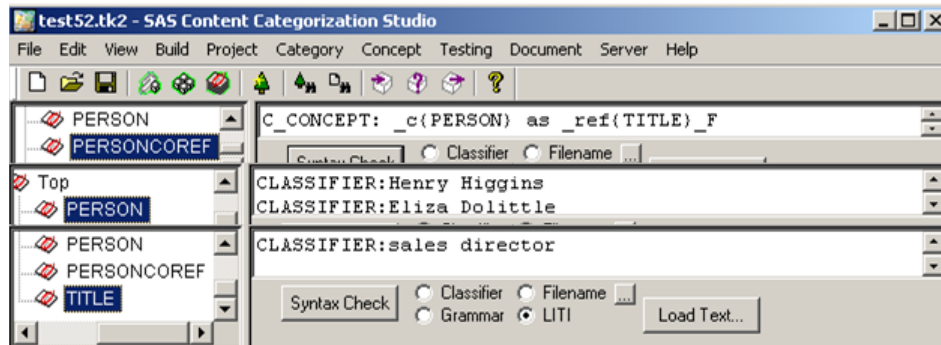
Use the forward (`_F`) and the preceding (`_P`) symbols to restrict coreference matches in an input document. When you specify these operators, only the matches that follow or precede the match for the rule, respectively, are returned.

Use these symbols when you want to return all of the matches instead of the one match that follows the rule (`coref` operator alone). Unlike the greater than (`>`) symbol, all of the returned matches can occur only before or after the coreference rule match.

3.9.4.B Matching with the Forward Symbol

Use the forward symbol (`_F`) to return all of the matches that follow a coreference rule match.

Figure 3-40 CONCEPT with _ref and Forward Symbol



The example above shows a concept with a concept rule with a forward symbol. The rule specifies that all of the instances of matches on the coreference term that follow the coreference match are returned as matches. (Any matches that precede the match on the coreference term are not returned.)

Example 3-20: C_CONCEPT Rules with the _F Symbol

Concept Name	Entry
PERSON	CLASSIFIER:Eliza Dolittle
TITLE	CLASSIFIER:sales director
PERSONCOREF	C_CONCEPT:_c{PERSON} as _ref{TITLE}_F

In this example, a match on the term *Eliza Dolittle* as *sales director* matches. Instances of the term *sales director* that follow are also returned as matches.

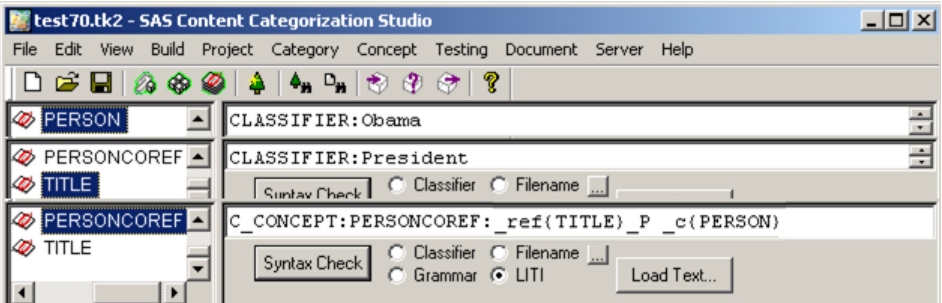
Figure 3-41 _ref and Forward Symbol Matches



3.9.4.C Matching with the Preceding Symbol

Use the preceding symbol (`_P`) to return matches on all instances of a coreference match that occur before the coreference rule match.

Figure 3-42 CONCEPT with `_ref` and Preceding Symbol



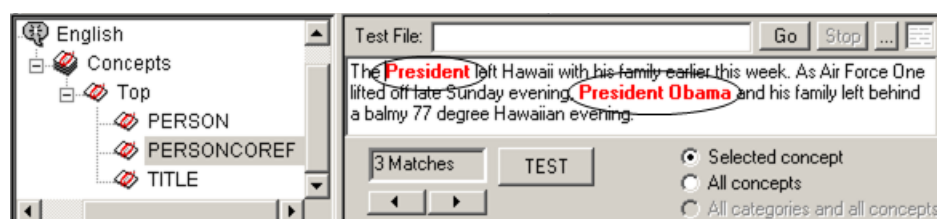
The example above shows a concept with a rule that specifies a preceding symbol. All instances of matches on the TITLE concept that are immediately followed by a match on the PERSON concept are returned as matches. (Any matches that follow the match on the coreference term are not returned.)

Example 3-21: C_CONCEPT Rules with the `_P` Symbol

Concept Name	Entry
PERSON	CLASSIFIER:Obama
TITLE	CLASSIFIER:President
PERSONCOREF	C_CONCEPT:_ref{TITLE}_P _c{PERSON}

In the example above, all instances of a match on the TITLE concept that precede a match on the TITLE and PERSON concepts are matched in an input document.

Figure 3-43 Matches on a Rule with the Preceding Operator



3.9.5 Coreference in a Classifier Definition Example

You can use the coreference operator (`coref`) to link a match in a coreference definition to its canonical form. For example, you might want to return *Barack Obama* for a match on any instance of the word *president* in an input document. The `coref` qualifier is used with classifier definitions, only.

Figure 3-44 Coreference Used to Link to Classifier Concept



The example above shows a classifier definition that links matches on the `coref` qualifier to its canonical form.

Example 3-22: A Classifier Concept with a Coreference Qualifier

Concept Name	Entry
FULLNAME	CLASSIFIER:[coref=Clinton,William Clinton;TITLE:President]:Bill Clinton

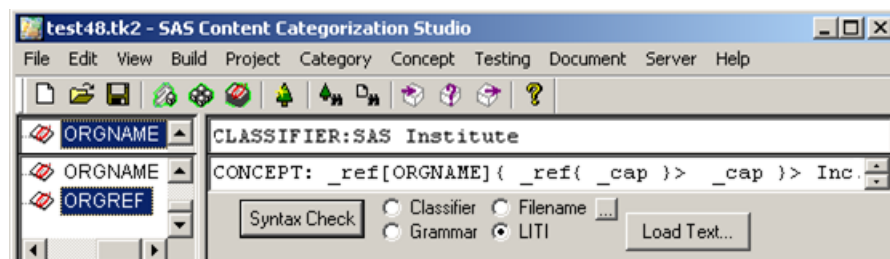
In the example above, if the canonical term *Bill Clinton* is matched once in an input document, all instances of matches on the `coref` qualifier terms also return matches. In this example, *Clinton*, *William Clinton*, and *President* all return matches. The canonical form for each matched term is *Bill Clinton*.

3.9.6 Assigning New Concept Names to Coreference Matches

You can assign a new concept name for a match on a term specified by the `_ref` operator. In this case, any instances of this match are output in SAS Content Categorization Server as a match on this new concept. You can also write a rule that specifies that a match is assigned to an existing concept. For example, you could assign matches on the names of an organization to an existing `CLASSIFIER` definition. In both cases, any matches on the complete definition are returned in the specified canonical form.

Specify a new, or an existing, concept name in square brackets (`[]`) that are preceded by the `_ref` operator. For example, specify `_ref [COMPANY]`.

Figure 3-45 Reassigning a Match



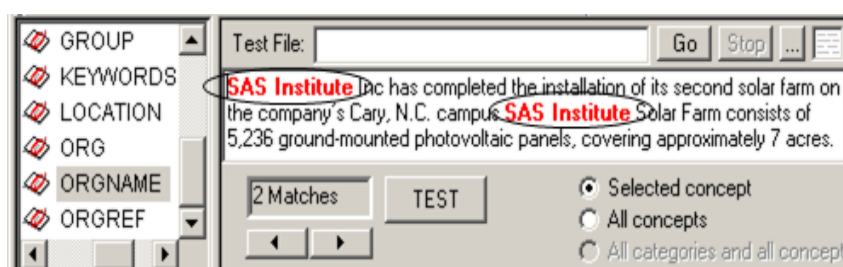
In the example above, if a sequence of two or more words that begins with an uppercase letter is followed by *Inc.*, a match is returned for the `ORGREF` concept. A sequence of two words that begin with uppercase characters is returned as a match for the concept `ORGNAME`. The canonical form is returned as a match for the `ORGREF` concept.

Example 3-23: Assigning a New Concept Name to a Coreference Match

Concept Name	Entry
ORGNAME	CLASSIFIER:SAS Institute Inc.
ORGREF	CONCEPT:_ref[ORGNAME]} {_ref (_cap)> _cap}> Inc.

In the example above, a match on the `ORGNAME` concept is returned when there is a match on the remainder of the `ORGREF` rule. For example,

Figure 3-46 Match Returned to Another Concept

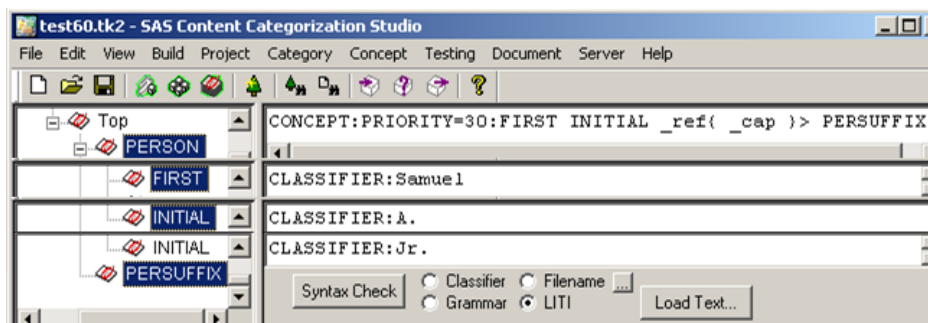


3.9.7 Rank Coreference Definitions and Eliminate False Positives

You can use the `PRIORITY` specification to make matches on one coreference rule rank higher than other rules. Specify a priority to rank matches on the concept that uses coreference higher than other matched concepts. (When you specify a `PRIORITY` in a rule, this setting overrides the **Priority** setting in the Data window—for this rule only.)

You can choose to specify a priority for a concept match that uses the `_ref` operator with the export symbol. You can also use the `PRIORITY` specification to eliminate false positives. For more information about priorities, see Section 3.7.8 *Setting Priorities for Overlapping Matches* on page 47.

Figure 3-47 `CONCEPT` with `_ref` and Export Symbol



In this example, if *Samuel A. Alito Jr.* is present once in the document, then every match on *Alito* returns his full name. The canonical form is *Samuel A. Alito Jr.* and the referring term is *Alito*.

Example 3-24: C_CONCEPT Rule with the Export Symbol

Concept Name	Entry
FIRST	CLASSIFIER:Samuel
INITIAL	CLASSIFIER:A.
PERSUFFIX	CLASSIFIER:Jr.


```

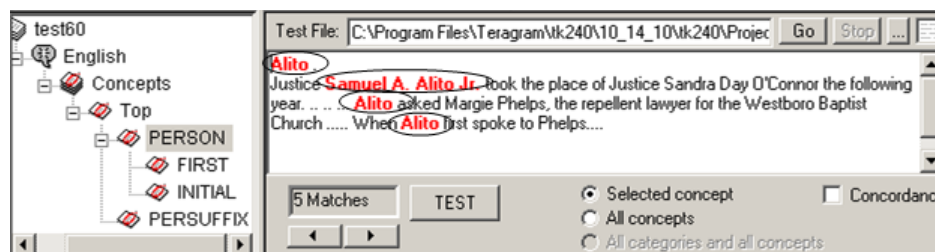
PERSON concept:          CONCEPT:PRIORITY=30:FIRST INITIAL
                           _ref{ _cap }> PERSUFFIX

```

In the example above, all instances of *Alito* are matched in an input document when all of the following conditions are met. A match on a first name listed in the FIRST classifier concept is located. This match is followed by a match on an initial specified in the INITIAL concept. When a word beginning with an uppercase letter follows this match, it is the coreference that is matched by all instances that occur in the document. Finally, a match on the PERSUFFIX concept is located.

In the example shown below, all instances of *Alito* are returned as a match. The PERSON concept also has a priority setting of 30. This means that matches on the PERSON concept rank higher than the matches that are also returned to the FIRST and INITIAL definitions.

Figure 3-48 _ref and Export Symbol Matches



3.10 XML Fields

3.10.1 Overview of XML Field Matching

If the input is a valid XML document, SAS Contextual Extraction Studio enables you to write rules that restrict matches to the fields that you specify. These are the ways to process XML documents:

1. Specify default fields in the rules.
2. Specify field names in the rules.
3. Combine both operations.

By default, text is extracted from all of the fields before matching takes place. If you want to restrict matching to specific fields, you can specify these fields in the **XML Default Field** of the **Misc** tab in the Project Settings interface.

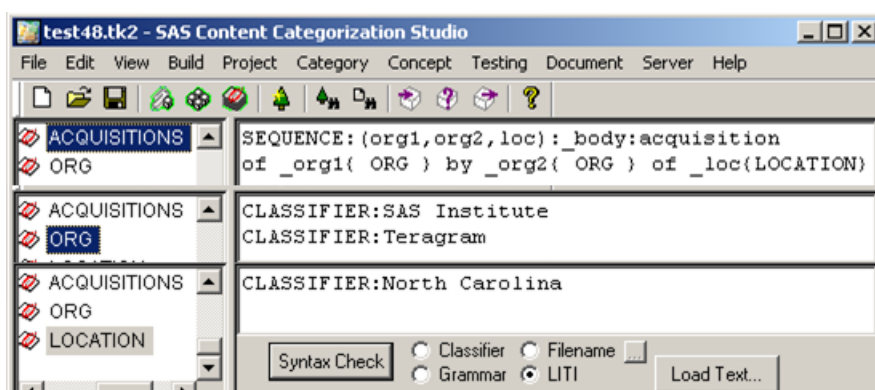
You can specify one XML field with the `CLASSIFIER`, `CONCEPT`, `C_CONCEPT`, `SEQUENCE`, `NO_BREAK`, and `REGEX` rules. Specify the field name at the beginning of the pattern to be matched. For example, specify the `body` field as the location where all matches occur.

Note: Matches are returned only if the matches are located within, and not across, fields.

3.10.2 SEQUENCE Rules with an XML Field

When you write a `SEQUENCE` rule, all of the individual tokens or concepts are matched. These matches occur if all of the tokens and concepts are present within the specified field. `SEQUENCE` rules do not enable matching across fields.

Figure 3-49 Body XML Field Specified in a Rule



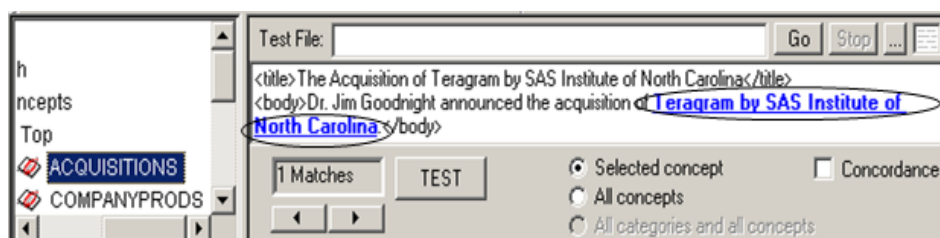
The XML field is preceded by an underscore (_) and the concepts to be matched follow. In the `SEQUENCE` rule example above, there are several arguments. A match occurs when each of these arguments is matched in the `body` field of an input XML document.

Example 3-25: Assigning a New Concept Name to a Coreference Match

Concept Name	Entry
ORG	CLASSIFIER:SAS Institute CLASSIFIER:Teragram
LOCATION	CLASSIFIER:North Carolina
ACQUISITIONS	SEQUENCE:(org1,org2,loc):_body: acquisition of _org1{ ORG } by _org2{ ORG } of _loc{LOCATION}

A match for the `ACQUISITIONS` concept occurs when the term *acquisition of* occurs followed by two matches on the `ORG` concept separated by the word *by*. This match is complete when it is followed by a match on the `LOCATION` concept and all of these matches occur in the `body` field.

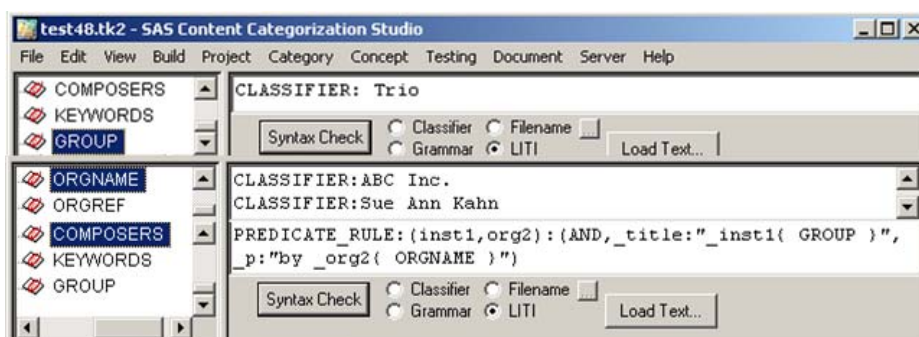
Figure 3-50 Match Located in an XML Field



3.10.3 Matching More than One XML Field

If you choose to use a PREDICATE_RULE, CONCEPT_RULE, or a REMOVE_ITEM definition, you can specify a separate field for each argument.

Figure 3-51 A Predicate Rule Specifying XML Fields



Each XML field is preceded by an underscore (_). For example, _title and _p. The specified matches are enclosed in quotation marks (""). See the following example:

Example 3-26: Matching XML Fields

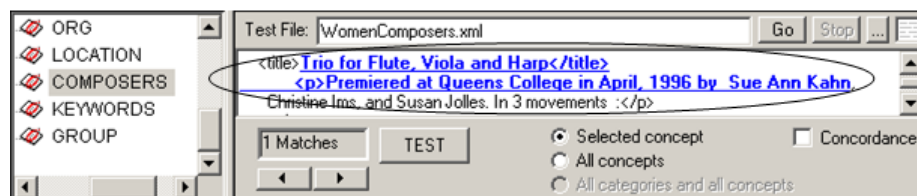
Concept Name	Entry
GROUP	CLASSIFIER:Trio
ORGNAME	CLASSIFIER:ABC Inc. CLASSIFIER:Sue Ann Kahn

COMPOSERS

```
PREDICATE_RULE:(inst1,org2):(AND,  
  _title:"_inst1{ GROUP }",_p:"by  
  _org2{ ORGNAME }")
```

A match for the COMPOSERS concept occurs when there is a match in the title field on the GROUP concept. The match is complete when there is also match on the p (paragraph) field on the word *by* followed by a match on the ORGNAME concept.

Figure 3-52 Predicate Rule Match in XML Document

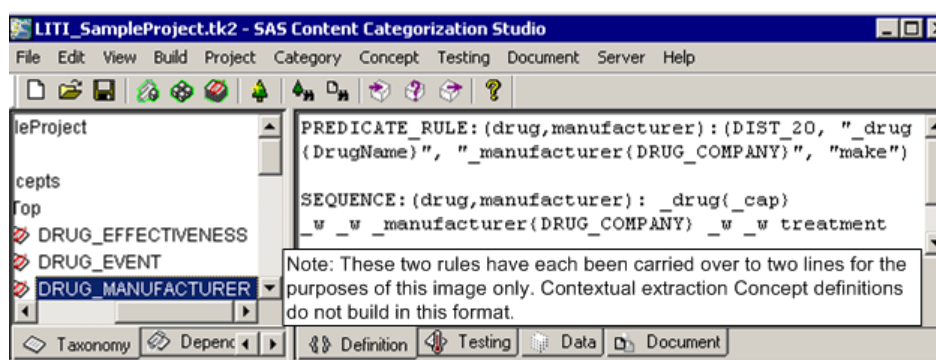


3.11 Writing Multiple Rules for One Definition

Write multiple rules for each contextual extraction concept. This feature increases the recall of your definitions by enabling you to locate more matches as well as matches based on different specifications.

For example, add the SEQUENCE rule shown in Example 3-16 on page 64 to the definition of the DRUG_MANUFACTURER concept to locate matches in documents that might not otherwise match.

Figure 3-53 PREDICATE_RULE with a SEQUENCE Rule



3.12 Troubleshooting Your Rules

If you do not obtain the results that you expect, or if SAS Contextual Extraction Studio returns syntax error messages, troubleshoot your rules.

To troubleshoot your rules, use the following list:

- **LITI** radio button: Did you specify an LITI concept in the Definition window?
- **NO_BREAK rule**: Did you specify that partial matches cannot be returned for a term? If so, did you remember that this rule applies across the entire taxonomy?
- **Case sensitivity**: Have you specified your rules to match the upper- and lowercase words that you want to match?
- **sep** part-of-speech: Did you remember to specify `sep` beginning with a lowercase `s`?
- **Project Settings - LITI**: Are these settings returning the best results?
- **Rule type**: Did you specify the correct rule type using all uppercase letters?
- **Spaces**: Did you remember to use spaces before the colon (`:`) that precedes part-of-speech tags?

-
- **Curly braces** ({}): Did you surround the term that you want to return with curly braces?
 - **Square braces** ([]): Did you surround the new, or other, concept to be matched with square braces when you wrote a coreference rule?
 - **Syntax**: Have you checked the rule syntax using the **Syntax Check** button in the **Definition** tab before compiling your concepts? Is this syntax appropriate for the results that you are trying to return, or is there a better syntax or rule type?

Appendixes

- Appendix A: *Using the Directive and Regex Syntax on page 89*
- Appendix B: *Part-of-Speech Tags on page 93*
- Appendix C: *Recommended Reading on page 99*
- Appendix D: *Glossary on page 101*

Appendix: A

Using the Directive and Regex Syntax

- *Using the Directive in the Configuration File*
- *Regular Expressions*

A.1 Using the Directive in the Configuration File

The example below provides the syntax for the SAS Contextual Extraction Studio directive in the server configuration file that points to a .li binary file. This directive works like the existing `mcats` and `concepts` directives. For more information, see Chapter 3 in the *SAS Content Categorization Server: Administrator's Guide*.

Example A-1: An LITI Directive

```
basedir=C:\Program Files\SAS
        \SAS Contextual Extraction Studio\
mcats=data/English.mco:IPTC
concepts=data/English.concepts:Entities
liti=data/English.li:LITIDemo
```

A.2 Regular Expressions

A.2.1 Rules and Restrictions

The following rules and restrictions apply to regular expressions:

- Any single character **a** (ASCII 1 through 252, subject to escaping restrictions) is a regular expression, and it matches precisely that character.
- If **a** and **b** are regular expressions, then so is **ab** that matches whatever **a** matches followed by whatever **b** matches (concatenation).
- If **a** and **b** are regular expressions, then so is **a|b** that matches either whatever **a** matches or whatever **b** matches.
- If **a** is a regular expression, then so is **(?:a)** that simply serves as a grouping mechanism without remembering what it was grouping. For example, **(?:ababb)|b** matches either **abaab** or **b**. This regular expression is difficult to express without the grouping mechanism.
- A character class is a regular expression. One or more characters inside square braces (**[]**) matches any of the characters inside. For example, you could write **[abc]**. A range inside a character class matches any ASCII character whose value is between the specified characters. For example, **a-z**, matches **a** through **z**, inclusive. Any character can appear in a character class. However, **** (backslash), **-** (hyphen), and **]** (close brace) is preceded by a backslash. A **^** (carat) is preceded by a backslash, if it is the first character in the character class.
- A negated character class is a regular expression. One or more characters are inside square braces, with **^** (carat) being the first character to indicate negation. For example, **[^abc]** matches any character except **a**, **b**, or **c**.
- If **a** is a regular expression, then so is **a*** that matches 0 or more occurrences of whatever **a** matches.
- If **a** is a regular expression, then so is **a+** that matches 1 or more occurrences of whatever **a** matches.
- If **a** is a regular expression, then so is **a?** that matches 0 or 1 occurrences of whatever **a** matches.

-
- If **a** is a regular expression, then so is **a{n,m}** that matches at least **n** but no more than **m** concatenated occurrences of whatever **a** matches.
 - If **a** is a regular expression, then so is **a{n,}** that matches at least **n** concatenated occurrences of whatever **a** matches.
 - If **a** is a regular expression, then so is **a(n)** that matches exactly **n** concatenated occurrences of whatever **a** matches.
 - If filename is the name of a file containing the binary representation of a sub-expression **a**, then the syntax **(?\$filename)** inserts that sub-expression into the current regular expression. To create such a file, use the `_treg` utility as follows:

```
_treg -to_fso 'a' >filename
```

A.2.2 Special Characters

The table below lists and gives extended meaning for special characters with regular expressions.

Table A-1: Special Characters in Regular Expressions

Character	Meaning
\a	Alarm (beep)
\n	Newline
\r	Carriage return
\t	Tab
\f	Form feed
\e	Escape
\d	Digit (same as [0-9])
\D	Not a digit (same as [^0-9])
\w	Word character (same as [a-zA-Z_0-9])
\W	Non-word character (same as [^a-zA-Z_0-9])
\s	Whitespace character (same as [\t\n\r\f])

Table A-1: Special Characters in Regular Expressions (Continued)

Character	Meaning
\S	Non-whitespace character (same as <code>[^\t\n\r\f]</code>)
.	Wildcard (matches any character)
\xh	Hexadecimal number, where h is a hexadecimal digit
\xhh	Hexadecimal number, where h is a hexadecimal digit
0o	Octal number, where o is an octal digit
0oo	Octal number, where o is an octal digit

A.2.3 Special Cases

These are the special cases for regular expressions:

1. For metacharacters to have literal meaning, the metacharacters need to be escaped with a backslash (\). For example, escape `[,],(,?),*,+,.,\,|` with a backslash. If inside a character class, however, only those mentioned explicitly need escaping.
2. No support is provided for backward references or `()` as a remembering grouping mechanism.
3. No support is provided for `^` as the beginning-of-line, zero-width assertion, or `$` as the end-of-line, zero-width assertion. Unlike Perl regular expressions, both of these markers are implicitly assumed.
4. ASCII values 0, 253, 254, and 255 are reserved characters that cannot be used in regular expressions. Regular expressions work only on single-byte characters.

Appendix: B

Part-of-Speech Tags

The table below provides examples of the majority of morphological feature combinations for English parts of speech. For more information about how these parts of speech are used to write rules, see Section 3.5.14 *The Part-of-Speech Tags* on page 25. Also see the language book for each language that you purchased.

Table B-1: Part-of-Speech Morphological Features

Code	Part-of-Speech	Example
A	adjective	The sky is <i>azure</i> .
ABBREV	abbreviation	etc.
Acomp	comparative adjective	The green bag is <i>heavier</i> than the red one.
Adv	adverb	He is <i>easily</i> the best candidate.
Asup	superlative adjective	He cooked the <i>best</i> dish.
C	conjunction	Say nothing of former informers <i>and</i> spies.

Table B-1: Part-of-Speech Morphological Features (Continued)

Code	Part-of-Speech	Example
date	valid date formats YYYY-MM-DD YYYYMMDD YY-MM-DD YYMMDD YYYY-MM YYYYMMs YY-MM Standard US Date Formats MM-DD-YYYY MM/DD/YYYY MM-DD-YY MM/DD/YY	04JAN2001 04jan2001
Det	determinant	Nothing can be further from <i>the</i> truth.
digit	numeric symbols, including floating point decimals	5, 2.14, or 5,254
F	French word	We went to see the <i>chateaux</i> .
inc	unknown word to the part-of-speech tagger	
Int	interjection	Yum!
Md	modal verb	This <i>might</i> be the best idea.
Mdn 't	modal verb negated	I <i>won't</i> elaborate on this any further.
N	noun	The <i>e-mail</i> went to the spam folder.
Npl	plural noun	The <i>geese</i> are leaving for the South.

Table B-1: Part-of-Speech Morphological Features (Continued)

Code	Part-of-Speech	Example
Num	number	She just turned <i>seventeen</i> years old.
PN	proper noun	We are going to <i>England</i> for vacation.
PossDet	possessive determinatant	It is <i>her</i> choice.
PossPro	possessive pronoun	The choice is <i>hers</i> alone.
PreDet	<i>pre</i> determinatant	<i>All</i> the king's soldiers could not put him together again.
Prefix	prefix	The <i>multi</i> -millionaire Soros is going to help us out.
Prep	preposition	Let's go <i>to</i> grandma's house.
Pro	pronoun	Give me one of <i>each</i> .
ProMD	pronoun contracted with modal	If it <i>weren't</i> for him, we'd still be here.
ProV	pronoun contracted with a verb	we're
Ptl	particle	I would go <i>across</i> if I could.
RelPro	relative pronoun	I want the coin <i>that</i> represents King Kong.
sep	separator character	;;,.,.,.

Table B-1: Part-of-Speech Morphological Features (Continued)

Code	Part-of-Speech	Example
time	time formats 23:59:59 235959 23:59:59.9942 235959.9942 23:59:59Z 23:59:59.9942Z 235959.9942Z 23:59:59+HH:MM 23:59:59-HH:MM 235959+HHMM 23:59:59.9942Z 235959.9942Z	12:56:32

Table B-1: Part-of-Speech Morphological Features (Continued)

Code	Part-of-Speech	Example
time (continued)	Standard US and British Time Formats 10:15AM 10:15A.M. 10:15am 10.15a.m. 10AM 10A.M. 10am 10a.m. 10:15PM 10:15P.M. 10:15pm 10.15p.m. 10PM 10P.M. 10pm 10p.m.	9:00PM
url	urls	www.sas.com/success/
v	verb	You should <i>verbalize</i> your wishes.
V3sg	verb, 3 rd person singular	The boy <i>amuses</i> himself throwing rocks.
V3sgn't	verb, 3 rd person singular negated	This <i>isn't</i> funny.
Ving	present participle	Why is the hen <i>crossing</i> the street?
Vn't	negated verb	"it <i>don't</i> mean a thing..."

Table B-1: Part-of-Speech Morphological Features (Continued)

Code	Part-of-Speech	Example
Vpp	past participle	Those tapes were <i>released</i> .
Vpt	verb, past tense	The president <i>hated</i> broccoli.
Vptn't	verb, past tense negated	If it <i>weren't</i> for him, we'd still be here.
WAdv	w adverb	<i>Why</i> do you say that?
WDet	w determinant	<i>What</i> is he saying?
WPossPro	w possessive pronoun	<i>Whose</i> hat is this?
WPro	w pronoun	<i>Whom</i> did you meet?

Appendix: C

Recommended Reading

The following books are recommended as companion guides:

- *SAS Contextual Extraction Studio: Installation Guide*: Install SAS Contextual Extraction Studio.
- *SAS Content Categorization Studio: User's Guide*: Create a SAS Content Categorization Studio project, test, and upload to SAS Content Categorization Server.
- *SAS Content Categorization Studio: Installation Guide*: Install SAS Content Categorization Studio.
- *SAS Content Categorization Studio: Quick Start Guide*: Advanced users can learn how to expeditiously set up a SAS Content Categorization Studio project.
- *SAS Content Categorization Collaborative Server: Administrator's Guide*: Configure the server for multiple subject matter experts. Grant permissions to these users and upload projects to the server.
- *SAS Content Categorization Collaborative Server: User's Guide*: Enable multiple subject matter experts to work together on one SAS Content Categorization Studio project.
- *SAS Content Categorization Server: Administrator's Guide*: Automate the application of the `.mco` and `.concepts` files to input documents.
- Use the language books for each language purchased to see the comprehensive list of part-of-speech tags that are available.

SAS offers instructor-led training and self-paced e-learning courses to help you get started with the SAS add-in, learn how the SAS add-in works with the other products in the SAS Enterprise Intelligence Platform, and learn how to run stored processes in the SAS add-in.

For more information about the courses available, see support.sas.com/training.

For a complete list of SAS publications, see the current SAS Publishing Catalog. To order the most current publications or to receive a free copy of the catalog, contact a SAS representative at

SAS Publishing Sales

SAS Campus Drive

Cary, NC 27513

Telephone: (800) 727-3228*

Fax: (919) 677-8166

E-mail: sasbook@sas.com

Web address: support.sas.com/pubs

* For other SAS Institute business, call (919) 677-8000.

Customers outside the United States should contact their local SAS office.

Appendix: D

Glossary

_c

specifies the context for the matches.

_cap

specifies that a word beginning with an uppercase letter is a match.

argument

is defined by two or more concepts that are related to each other. When these matches are identified, arguments are returned. See *fact*.

canonical form

specifies the full name, or form, of the term. For example, SAS Institute Inc. is the canonical form of SAS.

CLASSIFIER

specifies the terms to be matched. This rule works like the SAS Content Categorization Studio classifier definition because it provides a simple list of terms to be matched.

CONCEPT

locates entities, or ideas, in input documents.

Concordance

displays a list of the matching terms located in a document with the text surrounding them. Specify the number of characters or words that are returned for a match on a concept.

coreference

refers to pronoun resolution. A pronoun is matched to the antecedent that it refers to. Coreference is also known as *anaphora resolution*.

definition

defines a concept, whether it consists of one or more rules. *Definition* is used interchangeably with the word *rule*. See *rule*.

event

is used interchangeably with *fact*. See *fact*.

Fact

refers to two or more concepts or tokens that are specified in one `SEQUENCE` or `PREDICATE_RULE` definition. See *SEQUENCE* and *PREDICATE_RULE* below.

precision

is a measurement of the relevancy of the matched documents. In other words, the concept definition excludes possible matches that do not reflect the subject matter of the concept. For example, texts referring to *rock collections* are not matched for the category *Rock and Roll*.

PREDICATE_RULE

returns matches when an operator is specified with arguments. Unlike the `SEQUENCE` definition, the matches do not need to occur in the order specified by the definition.

SEQUENCE

returns facts when matches occur within the specified context.

priority

ranks concepts. By default, priority is set to 10 in the Data window for contextual extraction concepts. This specification prioritizes contextual extraction concepts over classifier and grammar concepts.

recall

is a measurement of how well the definition matches all of the relevant texts.

referring term

is a term that refers to a canonical form.

REGEX

specifies regular expression syntax.

rule

defines the concept. There can be many rules for each contextual extraction concept definition. This term is used interchangeably with definition, but properly speaking, one definition can contain many rules. See *definition*.

SEQUENCE

returns facts when matches occur within the specified ordering.

string

refers to a group of words or characters that you specify for a rule.

token

is a synonym for a word. *Token* is not a synonym for the word *string* that can refer to several words or characters. *Token* refers only to one word.

Index

%	
usage	26
+	
usage	26
.concepts	
defined	13
.li	
defined	4
directive	89
.mco	
defined	4
>	
usage	23, 38, 74
_c	
context operator	73
usage	22, 38
_cap	
defined	19
usage	22
_coref	
classifier rule	72
_F	
usage	74
_P	
usage	76
_ref	
export symbol	79
new concept	78
usage	72, 74
_w	
usage	22
{}	
usage	64

A

ALIGNED	
defined	29
usage	30
All matches	
Data window	62
usage	11, 67
AND	
defined	29
usage	30
architecture	
image	4
argument	
defined	63
fact	71

B

Best	
Data window	62
usage	11

C

C_CONCEPT	
_ref operator	73
defined	19
spaces	25
canonical form	
coreference	72
case-insensitive	
matching	21
case-sensitive	
matching	18
CLASSIFIER	
coref	77
defined	19, 33
classifier rule	
_coref	72

colons	
usage	24
commas	
usage	24
CONCEPT	
defined	19, 35, 44
spaces	25
concept matching	
preference	18
CONCEPT_RULE	
defined	20, 53, 55, 58, 60, 69
spaces	25
Contextual definition	
defined	47
Priority field	47
coref	
CLASSIFIER	77
coreference	
canonical form	72
operators	72
curly braces	
usage	23

D

Data window	
Priority field	7, 27, 61
Definition window	
usage	17
dictionary entries	
part-of-speech tags	50
disambiguation	
defined	42
DIST	
defined	29
usage	30, 58, 60
document	
PARA	18
SENT	18

Document window	
fact	68
Project Settings	10
duplicate instances	
return	37

E

export feature	
usage	44, 45
export symbol	
_ref	79
exported terms	
not in rule	45

F

fact	67
argument	71
defined	10, 63
Document window	68
multiple	67
PREDICATE_RULE	66
SAS Content Categorization Server	65
view matches	64
filename	91

L

li	
defined	13
LITI directive	89
LITI window	
Project Settings	10
location	
matches occur	81

logical operators	
table	29
Longest	
Data window	62
usage	11, 12, 68

M

Misc tab	
Project Settings	81
multiple rules	
add	84

N

NO_BREAK	
defined	19, 40
usage	41

O

OR	
defined	29
ORDDIST	
defined	29
usage	31, 60
Overlapping Concept Matches	
usage	11

P

PARA	
document	18
paragraph field	
match	84
parentheses	
usage	23

partial match	48
part-of-speech tags	
codes	93
requirements	50
percent	
REGEX	51
usage	26
PREDICATE_RULE	
defined	20, 66, 67
fact	66
Prep	
defined	50
priority	
overlapping matches	47
rank	80
usage	27
Priority field	
Contextual definition	47
Data window	7, 27, 61
PRIORITY specification	
usage	79
Project Settings	
Document window	10
LITI window	10
Misc tab	81
REMOVE_ITEM	12
Project Settings - LITI tab	
rule matches	18

Q

quotation marks	
usage	23

R

rank	
priority	80
REGEX	
defined	20, 51
percent	51
usage	61
regular expressions	
usage	26
relative rankings	
increase	47
Remove duplicate facts	
usage	12, 63, 71
REMOVE_ITEM	
defined	19, 42
Project Settings	12
Return all identical matches	
Data window	63
usage	11, 69
rule matches	
Priority Settings	18
rules	
defined	18

S

SAS Content Categorization Server	
facts	65
SENT	
defined	29
document	18
usage	31
SENT_n	
defined	29
usage	31
SENTEND_n	
defined	29
usage	32
SENTSTART_n	
usage	31

sep	
defined	50
usage	85
SEQUENCE	
defined	20, 63
usage	84
square braces	
usage	23
Start	
Programs	17

T

token	
defined	21
usage	53, 61
Top node	
location	17

U

user interface	
display	6