

# **SAS® Content Categorization Studio 5.2 User's Guide**



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

## **SAS® Content Categorization 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](http://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

---

<b>About This Book .....</b>	<b>1</b>
Audience .....	1
Prerequisites .....	1
Conventions .....	2
 <b>What's New in SAS Content Categorization Studio 5.2 .....</b>	 <b>3</b>
 <b>1 About SAS Content Categorization Studio .....</b>	 <b>5</b>
1.1 What Is SAS Content Categorization Studio? .....	5
1.2 Benefits of Using SAS Content Categorization Studio .....	6
1.3 How Does SAS Content Categorization Studio Work? .....	7
1.4 About the Architecture .....	7
 <b>2 Using the Interface .....</b>	 <b>9</b>
2.1 Your First Look at SAS Content Categorization Studio .....	9
2.2 The SAS Content Categorization Studio Menus .....	11
2.2.1 About the Availability of Menus and Menu Selections .....	11
2.2.2 About Menus .....	12
2.2.3 The File Menu .....	12
2.2.4 The Edit Menu .....	12
2.2.5 The View Menu .....	14
2.2.6 The Build Menu .....	14
2.2.7 The Project Menu .....	15
2.2.8 The Category Menu .....	16
2.2.9 The Concept Menu .....	17
2.2.10 The Testing Menu .....	18
2.2.11 The Document Menu .....	19
2.2.12 The Server Menu .....	20
2.3 The Status Bar .....	21
2.4 The Standard Toolbar .....	21
2.5 The Taxonomy Tab .....	23
2.6 The Dependencies Tab .....	24
2.7 The Right Window Tabs .....	25
2.7.1 About the Tabs .....	25
2.7.2 The Rules or Definition Tab .....	26

---

2.7.2.A The Rules Tab .....	26
2.7.2.B The Definition Tab .....	29
2.7.3 The Testing Tab .....	31
2.7.3.A The Testing Tab for Categories .....	31
2.7.3.B The Testing Tab for Concepts .....	31
2.7.3.C The Testing Tab Components .....	32
2.7.4 The Data Tabs .....	33
2.7.4.A The Data Tab for Categories .....	33
2.7.4.B The Data Tab for Concepts .....	35
2.7.5 The Document and Concordance Tabs .....	39
2.7.5.A About the Document Tab .....	39
2.7.5.B The Document Tab for Categories .....	39
2.7.5.C The Document Tab for Concepts .....	40
2.7.5.D The Document Tab as Concordance for Concepts .....	41
2.7.5.E The Document Tab as Browser Interface .....	43
2.7.5.F The Document Tab Components .....	44
2.7.6 The Automatic Rule Tab .....	46
2.8 The Options Window .....	47
2.9 The Project Settings Windows .....	49
2.9.1 About Project Settings .....	49
2.9.2 The Project Settings for Categories .....	49
2.9.2.A The Category Tab .....	49
2.9.2.B The Query Server Tab .....	53
2.9.3 The Misc(ellaneous) Tab for Categories and Concepts .....	54
2.9.4 The Project Settings for Concepts .....	57
2.9.4.A The Concept Tab .....	57
2.9.4.B The Concordance View .....	60
2.10 The Graphical Report Pages .....	63
2.10.1 The Graphical Full Test Report Page .....	63
2.10.2 The Graphical Populate Result Window .....	67
2.11 The Uploading the Categorizer, or Concepts, to SAS Content Categorization Server Window .....	68
2.12 The Miscellaneous Windows .....	70
2.12.1 The Select a Language Window .....	70
2.12.2 The Enter Names Window for UTF-8 Languages .....	73
2.12.3 The Number of Taxonomy Nodes Window .....	75
2.12.4 The Tree Find Window .....	76
2.12.5 The Tree Replace Window .....	78
2.12.6 The Text Find and Replace Windows .....	78
2.12.7 The Compile Concepts and Build Categorizer Tabs .....	79

---



---

2.12.8 The Syntax Check Window .....	80
2.12.9 The Best Matches Window .....	81
2.12.10 The Concept Priorities Window .....	83
2.12.11 The Rule Matches Window .....	85
2.12.12 The Full Test Report Window .....	87
2.12.13 The Open Window .....	90
2.12.14 Examples of the Status Windows .....	91
2.12.14.A The Upload Complete Window .....	91
2.12.14.B The No Language Entries Window .....	91
2.12.15 The Please Note Window for Old Projects .....	91
2.12.16 The Concordance Windows .....	93
2.13 The Drop-down Taxonomy Node Operations .....	94
2.13.1 The Project Name Node Operations .....	94
2.13.2 The Language Node Operations .....	95
2.13.3 The Categorizer or Concepts Node Operations .....	96
2.13.4 The Individual Category or Concept Node Operations .....	97
<b>2 Creating Projects .....</b>	<b>99</b>
2.1 Overview of Creating Projects .....	99
2.2 Start SAS Content Categorization Studio .....	100
2.3 Creating a New Project .....	101
2.3.1 Create a New Project .....	101
2.3.2 Enable Categorization and Concepts Extraction .....	104
2.3.3 Import a Project from an XML File .....	105
2.4 Saving the Project .....	108
2.4.1 Overview of Saving the Project .....	108
2.4.2 Manually Save an Existing Project .....	108
2.4.3 Save a Duplicate Project .....	109
2.4.4 Automatically Save Your Project Before Testing .....	110
2.4.5 Save a Duplicate Project .....	111
2.5 Access an Existing Project .....	112
2.6 Set Installation-Specific Operations .....	113
2.6.1 Automate Operations for the Installation .....	113
2.6.2 The Option Window Settings That Affect Testing .....	115
2.6.2.A Save before Each Test .....	115
2.6.2.B Always Rebuild before Each Test .....	115
2.6.2.C See the Best Matches .....	116
2.6.3 Automatically Sort the Taxonomy .....	117
2.6.4 Hiding Display Names for UTF-8 Languages .....	119

---

---

2.6.5 Checking Syntax for Classifier Concepts .....	120
2.6.5.A Report Duplicate Entries When Checking Classifier Concepts .....	120
2.6.5.B Report Duplicate Classifier Entries That Have Different Disambiguation Rules .....	123
2.6.6 View the Taxonomy as Text .....	126
2.6.6.A Flag the Categories and Concepts with No Definitions .....	126
2.6.6.B Flag the Categories and Concepts with No Dependencies ....	128
2.7 Specifying Project Settings .....	130
2.7.1 Specifying the Initial Category Project Settings .....	130
2.7.1.A Specify Category Operations .....	130
2.7.1.B Specify Query Operations .....	131
2.7.2 Specify Miscellaneous Operations .....	132
2.7.3 Specifying Concept Project Settings .....	133
2.7.3.A Specify Concept Operations .....	133
2.7.3.B Specify the Concordance Operations .....	134
2.8 Navigating through Categories and Concepts .....	135
2.9 Export a UTF-8 Binary File .....	136
2.10 Upload the Categorizer or Concepts to SAS Content Categorization Server .....	138

## **Part 1: Categories ..... 139**

### **4 Categorization ..... 141**

4.1 Overview of Categorization .....	141
4.2 How to Categorize Documents .....	142
4.3 Choosing a Taxonomy Type .....	143
4.4 Planning Your Taxonomy .....	145
4.4.1 A Sample Flat Taxonomy .....	145
4.4.2 A Sample Hierarchical Taxonomy .....	146
4.4.3 Modifying the Taxonomy .....	148
4.4.4 View the Taxonomy as Text .....	148
4.5 Choosing a Categorizer .....	150
4.5.1 The Two Basic Categorizer Types .....	150
4.5.2 Using the Automatic Rule Generator Tool .....	150
4.6 Optimizing Precision .....	151
4.6.1 About Precision .....	151
4.6.2 About Precision and Recall .....	151
4.7 About the Testing and Training Documents .....	152

---

4.8 How to Build Categorizers .....	153
4.8.1 Build a Statistical Categorizer .....	153
4.8.2 Build the Automatic Rule Generator Tool .....	154
4.8.3 Building a Rule-Based Categorizer .....	155
4.8.3.A Build a Rule-Based Categorizer .....	155
4.8.3.B Finding Uniquely Identifying Terms .....	155
4.8.3.C Specifying Rule Types .....	155
<b>5 Creating Categories .....</b>	<b>157</b>
5.1 Overview of Creating Categories .....	157
5.2 Create a Category .....	158
5.3 Deleting One or More Categories .....	159
5.3.1 Remove One Category .....	159
5.3.2 Delete Two or More Categories .....	161
5.4 Specify a Custom Syntax Checker .....	162
5.5 Provide Metadata for Categories .....	163
5.6 Working with the Taxonomy Structure .....	165
5.6.1 Rename a Category .....	165
5.6.2 Finding and Replacing Category Names .....	166
5.6.2.A Find Text in the Taxonomy Tree .....	166
5.6.2.B Replace Text in the Taxonomy Tree .....	167
5.6.3 Creating Categories Using the Copy Operation .....	169
5.6.3.A Copy and Paste One Category .....	169
5.6.3.B Copy and Paste More than One Category .....	172
5.6.4 Moving One or More Categories .....	174
5.6.4.A Move One Category .....	174
5.6.4.B Move Two or More Categories .....	177
5.7 Evaluating a Referenced Category .....	179
5.8 Noting an Incomplete Category .....	181
<b>6 Using the Statistical Categorizer .....</b>	<b>183</b>
6.1 Overview of the Statistical Categorizer .....	183
6.2 Benefits of the Statistical Categorizer .....	184
6.3 Determining Category Membership .....	184
6.4 Quick Start Guide for the Statistical Categorizer .....	185
6.5 Training the Statistical Categorizer .....	187
6.5.1 Preparing to Train the Categorizer .....	187
6.5.2 Assemble a Training Set of Documents .....	188

---

6.5.3 Set Training Paths to the Training Directory .....	189
6.5.4 Placing the Training Files into the Training Directory .....	192
6.6 Building and Saving the Categorizer .....	192
6.6.1 Build the Statistical Categorizer .....	192
6.6.2 Saving the Project .....	193
6.7 Testing the Statistical Categorizer .....	193
6.7.1 Before You Test the Statistical Categorizer .....	193
6.7.2 Batch Test the Statistical Categorizer .....	194
6.7.3 Test One Document .....	196
6.7.4 Run a Full Test of the Categorizer .....	199
6.8 Revising the Statistical Categorizer .....	200
<b>7 Automatic Rule Generator Tool .....</b>	<b>201</b>
7.1 Overview of the Automatic Rule Generator .....	201
7.2 Benefits of the Automatic Rule Generator Tool .....	202
7.3 Understanding Category Membership .....	203
7.4 Quick Start Guide for the Automatic Rule Generator Tool .....	203
7.5 Automatically Generating Rules .....	205
7.5.1 Prepare to Automatically Generate Rules .....	205
7.5.2 Automatically Generate Rules .....	205
7.6 Exporting Rules .....	206
7.6.1 Determining When and How to Export Rules .....	206
7.6.2 Option 1: Export All Generated Rules .....	207
7.6.3 Option 2: Export the Generated Rules for One Category .....	208
7.7 Clearing the Automatically Generated Rules .....	210
<b>8 Rule-Based Categorizers .....</b>	<b>211</b>
8.1 Overview of Rule-Based Categorizers .....	211
8.2 Benefits and Features .....	214
8.3 Quick Start Guide .....	216
8.4 Preparing to Write Your Rules .....	217
8.4.1 Understanding Rules and Category Membership .....	217
8.4.2 An Example of Category Rules .....	218
8.5 Developing Category Rules .....	219
8.5.1 Select a Rule Writing Operation .....	219
8.5.2 Write Rules .....	222
8.6 Check the Syntax of a Boolean Rule .....	224
8.7 Differentiating Symbolic Links from Dependencies .....	225
8.8 Create Symbolic Links .....	226

---

8.8.1 About Symbolic Links .....	226
8.8.2 Benefits of Symbolic Links .....	227
8.8.3 Define a Symbolic Link .....	228
8.9 Creating Dependencies .....	230
8.9.1 How Dependencies Work .....	230
8.9.2 Benefits of Dependencies .....	231
8.9.3 Creating Dependencies between Categories and Concepts .....	232
8.9.3.A Special Considerations .....	232
8.9.3.B Specify Project Settings with Dependencies .....	232
8.9.3.C Write the Concept Reference Syntax .....	235
8.9.4 Checking Dependencies Before Editing or Deleting a Category or Concept .....	237
8.9.4.A Knowing When to Check Dependencies .....	237
8.9.4.B Checking Dependencies before Deletions and Edits .....	238
8.10 Building the Rule-Based Categorizer .....	242
8.10.1 Manually Build the Categorizer .....	242
8.10.2 Automatically Rebuild the Rule-Based Categorizer .....	244
8.11 Automatically Save the Changes .....	245
<b>9 Relevancy and the Settings That Affect Relevancy .....</b>	<b>247</b>
9.1 Overview of Relevancy .....	247
9.2 Determining What Relevancy Type to Use .....	248
9.2.1 How Frequency-Based Relevancy Works .....	248
9.2.2 How Zone-Based Relevancy Works .....	248
9.2.3 How Operator-Based Relevancy Works .....	250
9.2.4 How Boolean Operators Affect Relevancy Weights .....	251
9.2.5 How Stemming Affects Relevancy .....	253
9.3 How to Set Relevancy Cutoff Settings .....	253
9.3.1 Analyzing Relevancy Cutoff .....	253
9.3.2 Specify Relevancy Cutoff Values .....	254
9.3.3 Test to Compute an Approximate Default Relevancy Cutoff Setting .....	256
9.4 About Relevancy and Category Bias Settings .....	259
9.4.1 How Relevancy and Category Bias Settings Are Determined .....	259
9.4.2 Setting the Default Category Bias .....	260
9.4.3 Set Category and Relevancy Bias .....	261

---

<b>10 Rule-Based Categorizer: Linguistic Terms .....</b>	<b>263</b>
10.1 Overview of the Rule-Based Categorizer .....	263
10.2 Benefits of Linguistic Rules .....	264
10.3 Quick Start Guide .....	265
10.4 About the Three Types of Linguistic Rules .....	269
10.5 Writing Rules in the Rules Window .....	270
10.5.1 Overview of the Components of the Rules Window .....	270
10.5.2 Write a Linguistic Rule .....	271
10.6 Weight Linguistic Rules .....	273
10.7 Specifying the Match Ratio .....	276
10.7.1 How Match Ratio Works .....	276
10.7.2 Optimizing the Match Ratio Setting .....	276
10.8 Selecting Special Symbols .....	278
10.9 Create Symbolic Links .....	281
10.10 Define Dependencies .....	281
 <b>11 Rule-Based Categorizer: Boolean Terms .....</b>	 <b>285</b>
11.1 Overview of Boolean Rules .....	286
11.2 Benefits of Boolean Rules .....	286
11.3 Quick Start Guide .....	288
11.4 About Category Membership .....	291
11.5 Benefits of Modes .....	292
11.6 About Boolean Operators .....	294
11.6.1 Overview of Boolean Operators .....	294
11.6.2 Common Boolean Operators .....	296
11.6.2.A The AND Operator .....	296
11.6.2.B The OR Operator .....	296
11.6.2.C The NOT Operator .....	296
11.6.2.D The MIN_n Operator .....	297
11.6.2.E The MINOC_n Operator .....	297
11.6.2.F The MAXOC_n Operator .....	297
11.6.2.G The SENT Operator .....	298
11.6.2.H The PAR Operator .....	298
11.6.2.I The DIST_n Operator .....	298
11.6.2.J The ORD Operator .....	299
11.6.3 Situational Boolean Operators .....	299
11.6.3.A The NOTIN Operator .....	299
11.6.3.B The NOTINDIST_n Operator .....	299
11.6.3.C The NOTINSENT Operator .....	300
11.6.3.D The NOTINPAR Operator .....	300

---

---

11.6.3.E The START_n Operator .....	300
11.6.3.F The END_n Operator .....	301
11.6.3.G The ORDDIST_n Operator .....	301
11.6.3.H The MAXPAR_n Operator .....	301
11.6.3.I The MAXSENT_n Operator .....	302
11.6.3.J The PARPOS_n Operator .....	302
11.7 Specifying Special Symbols .....	303
11.7.1 Overview of Special Symbols .....	303
11.7.2 About Stemming .....	305
11.7.3 Appending Suffixes .....	307
11.7.3.A The Suffix _C .....	307
11.7.3.B The Suffix _L .....	307
11.7.3.C The Suffix _Q .....	307
11.7.3.D The Suffix _C_Q .....	308
11.7.3.E The Suffix _L_Q .....	308
11.8 Specifying Structured Text Field Names .....	308
11.8.1 How to Use Structured Text Fields .....	308
11.8.2 How to Specify a Structured Text Field .....	309
11.8.3 How to Match Attributes in XML Fields .....	310
11.8.3.A Matching Attributes and Attribute Values .....	310
11.8.3.B Match Only If an Attribute Exists .....	312
11.8.3.C Match If an Attribute Exists and the Field Text Matches the Rule Term .....	313
11.8.3.D Match Only If an Attribute Contains the Specified Value ..	315
11.8.3.E Match Only If an Attribute Contains the Specified Value and the Rule Text Matches .....	316
11.8.4 How to Use Project Settings with Structured Text .....	318
11.8.5 Specifying the Caret and Dollar Symbols .....	320
11.8.6 Testing the Structured Text Rule .....	321
11.9 Editing Rules .....	322
11.9.1 Edit Rules in the Tree View Mode .....	322
11.9.2 About Statements and Operators .....	324
11.9.2.A Add a Statement .....	324
11.9.2.B Add an Operator .....	326
11.9.2.C Change an Operator .....	328
11.9.2.D Delete a Node .....	329
11.9.3 About Statement Commands .....	330
11.9.4 Expand Word Forms .....	331
11.9.5 Flag Categories without Definitions .....	332
11.10 Automating Parent and Child Rule Generation .....	333

---

---

11.10.1 Generate a Parent Rule .....	333
11.10.2 Generate Child Rules .....	334
11.11 Defining Symbolic Links .....	336
11.12 Dependencies between Categories or Categories and Classifier Concepts .....	337
11.12.1 About Dependent Nodes .....	337
11.12.2 Paste a Macro .....	337
11.12.3 Shorten Pathnames .....	339
11.12.4 Flag Categories with No Dependencies .....	340
11.13 Quick Start Guide to Testing Boolean Rules .....	341
11.14 Query an Index .....	342
<b>Part 2: Testing .....</b>	<b>347</b>
<b>12 Assembling Testing Sets .....</b>	<b>349</b>
12.1 Overview of Assembling Testing Sets .....	349
12.2 Creating Testing Folders .....	350
12.2.1 Create a Testing Directory While You Set Paths .....	350
12.2.2 Create and Set a Path to the Central Repository .....	355
12.2.3 Create a Testing Folder and Set a Path for a Newly Created Category .....	358
12.3 Collecting Test Files .....	359
12.4 Manually Populating a Testing Folder .....	359
12.5 Special Usages for a Central Repository .....	360
12.5.1 Automatically Populate Testing Paths .....	360
12.5.2 Create a Directory of Unmatched Testing Files .....	365
12.5.3 Import Test Files from a Central Repository .....	368
12.6 Delete Testing Files .....	371
<b>13 Batch Testing .....</b>	<b>373</b>
13.1 Overview of Batch Testing .....	373
13.1.1 Batch Testing Operations .....	373
13.1.2 About Testing Windows .....	375
13.2 About Testing Window Messages .....	376
13.3 Save and Compare Test Results .....	378



---

13.4 About Batch Testing .....	379
13.4.1 Option 1A: Batch Testing All of the Documents in One Category .....	379
13.4.2 Option 1B: Batch Testing the Testing Taxonomy or Out-of-Category Files .....	381
13.4.3 Comparing Test Results .....	382
13.5 Remove a Testing File .....	383
<b>14 Testing with One Document .....</b>	<b>385</b>
14.1 Overview of Testing with One Document .....	385
14.2 Test a Text in the Document Window .....	386
14.3 Testing a Web Page in the Document Window .....	388
14.3.1 Choosing Browser Operations .....	388
14.3.2 Load and Test the Source Document .....	390
14.4 See a Taxonomy of the Matching Nodes .....	391
14.5 See the Best Matches .....	393
14.6 Editing a Document in the Document Tab .....	394
14.6.1 Choosing Windows Commands .....	394
14.6.1.A Delete and Replace Text .....	394
14.6.1.B Copy and Paste a Test File .....	395
14.6.2 Clear a Test Document .....	396
14.6.3 Refreshing the Taxonomy Tree .....	396
14.6.4 Changing the Font Size of a Tested Document .....	396
14.6.5 Removing Markup Tags .....	396
<b>15 Other Testing Operations .....</b>	<b>397</b>
15.1 Overview of Other Testing Operations .....	397
15.2 Test a Central Repository .....	398
15.2.1 Test against a Single Testing Folder .....	398
15.2.2 Test Against a Central Repository .....	399
15.3 Import Failing Documents .....	402
15.4 About the Graphical Reports .....	404
15.5 About the Full Test Report .....	408
15.5.1 Completely Test the Categorizer .....	408
15.5.2 Interpreting the Report Statistics .....	409

---

<b>Part 3: Concepts .....</b>	<b>411</b>
<b>16 Developing a Concepts Taxonomy .....</b>	<b>413</b>
16.1 What is a Concept? .....	413
16.2 Determining How to Extract Concepts .....	414
16.3 Planning Your Taxonomy Structure .....	415
16.4 Create the Taxonomy Structure .....	416
16.5 Changing the Concepts in a Taxonomy .....	418
16.5.1 Recompile Concept Changes .....	418
16.5.2 About Moving Concepts .....	418
16.5.3 Move a Concept .....	419
16.5.4 Rename a Concept .....	420
16.5.5 Delete a Concept .....	421
16.5.6 Copy and Paste Concepts .....	423
<b>17 Defining Concepts .....</b>	<b>425</b>
17.1 Overview of Defining Concepts .....	425
17.2 Determining the Match Criteria .....	426
17.2.1 Provide Identifying Information for Your Concepts .....	426
17.2.2 Specifying the Project Settings .....	429
17.3 Understanding Concept Types .....	432
17.4 Use the Sample Concepts Project .....	433
17.5 Write a Definition .....	435
17.6 Use the Syntax Check Button .....	437
17.7 Compile Concepts .....	439
<b>18 Writing Classifiers .....</b>	<b>441</b>
18.1 Overview of Writing Classifiers .....	441
18.2 Writing a Classifier Definition .....	442
18.2.1 Format of Classifiers .....	442
18.2.2 Before You Write Classifier Definitions .....	444
18.2.2.A Specifying Project Settings .....	444
18.2.2.B Case Sensitivity .....	445
18.2.3 Writing the match_key .....	447
18.2.4 Writing the Information String .....	448
18.2.5 Matching the Comma Character .....	449
18.2.6 Locating Duplicates in the Match or Information Strings .....	451
18.3 Writing Regular Expression Definitions .....	452
18.4 Using Disambiguation to Increase Matching Precision .....	453

---

18.4.1 Overview of Disambiguation .....	453
18.4.2 Before You Write Disambiguation Definitions .....	454
18.4.3 Disambiguation Definition Examples .....	455
18.5 Write a Definition in a Text File .....	458
18.6 Generating Suggested Concepts .....	461
18.6.1 Overview of Generating Suggested Concepts .....	461
18.6.2 Generate Suggested Concepts .....	461
<b>19 Writing Grammar Rules .....</b>	<b>469</b>
19.1 Overview of Writing Grammar Rules .....	469
19.1.1 The Features and Benefits of Grammar Concepts .....	470
19.2 Specifying Project Settings and Options .....	471
19.2.1 Format of Grammar Rules .....	473
19.3 Specifying Terminal Symbols or Strings .....	474
19.4 Using Nonterminal Symbols .....	476
19.4.1 Understanding Non-Terminal Symbols .....	476
19.4.2 Using Characters .....	476
19.4.3 Specifying Part-of-Speech Tags .....	477
19.4.3.A Overview of Part-of-Speech Tags .....	477
19.4.3.B Part-of-Speech Tags in the English Dictionary .....	477
19.4.3.C PN Tag .....	478
19.4.4 Using the #cap and #w Symbols .....	480
19.4.4.A Available Languages and Case Sensitivity .....	480
19.4.4.B Specifying the #cap Symbol .....	480
19.4.4.C Specifying the #w Symbol .....	481
19.5 Writing Grammar Rules .....	483
19.5.1 Specifying the Root of the Grammar .....	483
19.5.2 Inserting Comment Lines into the Grammar .....	484
19.5.3 Writing Intermediate Concepts .....	485
19.5.4 Defining Dependencies in Grammar Rules .....	487
19.5.5 Write a Complete Grammar Rule .....	488
<b>20 Testing Concepts .....</b>	<b>491</b>
20.1 Overview of Testing Concepts .....	491
20.2 Understanding Testing Results .....	492
20.3 Setting the Priorities .....	493
20.4 Testing with the Concordance in the Document Tab .....	495
20.4.1 An Overview of the Concordance .....	495
20.4.2 Determine How the Concordance Is Displayed .....	495

---

20.4.3 See the Concordance Terms for a Selected Concept .....	498
20.4.4 See the Concordance Terms for All .....	499
20.5 Testing with the Concordance in the Testing Tab .....	501

## **Appendixes ..... 503**

### **A Regex Syntax and Part-of-Speech Tags ..... 505**

A.1 Regular Expressions .....	505
A.1.1 Rules and Restrictions .....	505
A.1.2 Special Characters .....	507
A.1.3 Special Cases .....	508
A.2 Part-of-Speech Tags .....	508

### **B Program Files ..... 511**

B.1 Overview of the Program Files .....	511
B.2 The Projects Folder .....	511
B.2.1 About the Projects Folder .....	511
B.2.2 SAS Content Categorization Studio File Format .....	514
B.3 The Data Folder .....	515
B.4 Configuration Examples .....	519
B.4.1 A Single Language Project .....	519
B.4.2 Multiple Language Projects .....	519
B.5 The Categorization XML File Format .....	520
B.5.1 About the Categorization XML File Format .....	520
B.5.2 The Language Encoding Specifications .....	520
B.5.3 The Project Settings .....	520
B.5.4 The Categories XML File .....	522
B.5.5 Closing the File .....	525
B.5.6 A Sample Categorization XML File .....	525
B.6 The Concepts XML File Format .....	527
B.6.1 About the Concepts XML File Format .....	527
B.6.1.A The Language Encoding Specifications .....	527
B.6.1.B The Project Settings .....	528
B.6.2 The Concepts XML File Format .....	530
B.6.3 The Concept Files .....	532
B.6.4 Closing the File .....	533
B.6.4.A About the Closing Tag .....	533
B.6.4.B The Sample Concept Extraction XML File .....	533

---

<b>C Recommended Reading .....</b>	<b>535</b>
<b>D Glossary .....</b>	<b>537</b>
<b>Index .....</b>	<b>543</b>



---

# About This Book

---

## Audience

SAS Content Categorization Studio is designed for the following users:

- Taxonomists who develop the categories and concepts that comprise the taxonomy for your enterprise
- Linguists and other subject matter experts who write the category rules and concept definitions
- Persons responsible for testing and analyzing the testing results for the rules specified for the categories and concepts

## Prerequisites


Here are the prerequisites for using SAS Content Categorization Studio:

- Load SAS Content Categorization Studio onto your machine
- Obtain access to documents that are representative of the types that you plan to categorize and extract concepts from
- Install the prerequisite fonts if you create a project that uses a UTF-8 language
- (Optional) A supported browser loaded onto your machine if you plan to test Web pages
- (Optional) If you install SAS Contextual Extraction, the **LITI** radio button is enabled in the **Definition** tab

---

## Conventions

This manual uses the following typographical conventions:

Convention	Description
TGM_ROOT	The root directory where SAS Content Categorization Studio is installed, typically the following:  <b>Windows:</b> C:/Program Files/Teragram/SAS Content Categorization Studio <b>UNIX:</b> /opt/SAS Content Categorization Studio
# BOOLEAN FIELDS bol:des	The code examples for _tk240 are shown in a fixed-width font.
<b>TEST</b> button	The labels for user interface controls are shown in a bold, sans-serif font.
Top	The names of taxonomy nodes appear in a fixed-width font.
<a href="http://www.sas.com">www.sas.com</a>	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.



---

# What's New in SAS Content Categorization Studio 5.2

---

New and enhanced features in SAS Content Categorization Studio include the following:

- SAS licensing replaces Teragram license
- Graphical reporting enables you to view precision, recall, and document matching information
- Generate subcategories now uses data from Wikipedia



---

## Chapter: 1

# About SAS Content Categorization Studio

---

- *What Is SAS Content Categorization Studio?*
- *Benefits of Using SAS Content Categorization Studio*
- *How Does SAS Content Categorization Studio Work?*
- *About the Architecture*

## 1.1 What Is SAS Content Categorization Studio?

In most organizations it is necessary to obtain information about, and from, data that is created internally and externally. SAS Content Categorization Studio enables you to define a taxonomy of categories and concepts that develops and identifies metadata about your information.

Using an intuitive, Windows interface, users with various skill sets and levels of expertise can develop a taxonomy. You can write rules for the categories that classify data and the concepts that extract entities, and test these rules by using sample documents.

### Easy taxonomy creation

Use the **Taxonomy** tab to create a visual taxonomy. This taxonomy has branches for different languages if you are building one project that uses multiple languages. The taxonomy also has separate branches for categories and concepts.

### Easy rule development

Use the **Rules**, or **Definition**, tab to write a category rule or a concept definition, respectively. You can also click the **Syntax Check** button that is available in these windows to instruct SAS Content Categorization

---

---

Studio to validate the syntax of the rule. *Rule* is used within this document to refer to a category rule as well as to a concept definition.

#### Easy Testing

Test your rules using groups of 10-20 documents that you assemble into a testing taxonomy. You can also collect documents that should fail. For example, the word *bush* in landscaping documents should not match *President Bush*.

#### Easy Uploading

After you develop and test the taxonomy, you can upload the compiled taxonomy rules as a `.mco` or `.concepts` binary file to SAS Content Categorization Server where the categories in this file are automatically applied to incoming documents.

## 1.2 Benefits of Using SAS Content Categorization Studio

SAS Content Categorization Studio provides users with the following benefits:

Empower subject matter experts and taxonomists by providing a simple, visual user interface where you build a taxonomy, define rules, and test

SAS Content Categorization Studio includes easy-to-use windows that simplify large, complex, and hierarchical taxonomies. You can specify your own rules, test, and generate `.mco` and `.concepts` files. These files are applied by SAS Content Categorization Server to input documents.

Develop metadata for your information

SAS Content Categorization Studio uses advanced linguistic technologies to identify metadata in, and about, your documents.

Improve the business value of information technology and the corporate data that it manages

SAS Content Categorization Studio creates `.mco` and `.concepts` files that automate the classification and extraction of entities from input documents during real time using SAS Content Categorization Server.

---

Save money on information retrieval and organization costs

All of the information created by, or within, your organization can be classified and retrieved. You can find information that is related, whether you know the exact terms that you are seeking.

## 1.3 How Does SAS Content Categorization Studio Work?

SAS Content Categorization Studio is a Windows application that anyone can use to develop taxonomies that classify and extract the information found in your organization. Interactively identify the data that you need without using a programming language.

SAS Content Categorization Studio enables users to easily create taxonomies, write rules, and test these rules against a variety of testing sets. You can upload the output `.mco` and `.concepts` files to SAS Content Categorization Server where they are automatically applied to input documents.

## 1.4 About the Architecture

Use the figure below to understand the processes used during the following two phases:

### Management phase

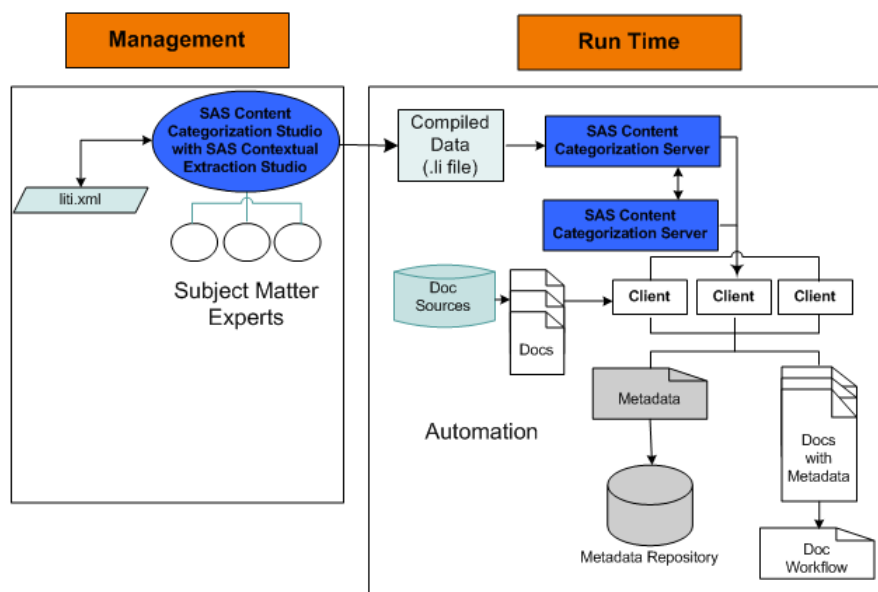
Subject matter experts specify a taxonomy of categories and concepts, or one or the other branch of this organizational structure. During the second part of this phase, these experts write rules to ensure that all of the documents that should match a category or concept are located. This is known as recall. These rules should also ensure precision, meaning that those texts that should not match are not returned as matches for the selected taxonomy node.

---

## Run time

The compiled SAS Content Categorization Studio data (.mco file) is sent to the SAS Content Categorization Server. SAS Content Categorization Server returns metadata about the document such as what categories the document matched.

Figure 1-1 Architecture



---

# Chapter: 2

## Using the Interface

---

- *Your First Look at SAS Content Categorization Studio*
- *The SAS Content Categorization Studio Menus*
- *The Status Bar*
- *The Standard Toolbar*
- *The Taxonomy Tab*
- *The Dependencies Tab*
- *The Right Window Tabs*
- *The Options Window*
- *The Project Settings Windows*
- *The Graphical Report Pages*
- *The Uploading the Categorizer, or Concepts, to SAS Content Categorization Server Window*
- *The Miscellaneous Windows*
- *The Drop-down Taxonomy Node Operations*

### 2.1 Your First Look at SAS Content Categorization Studio

To access the SAS Content Categorization Studio user interface, go to **Start —> Programs —> SAS Content Categorization Studio —> SAS Content Categorization Studio**.

---

---

Display 2-1 Main Window



The components of the main window are listed below from top to bottom:

**Program and Project title bar**

display the name of the program and the title of the current project. (The title only appears after you create a new project.)

**Menu bar**

access drop-down lists for project tasks. For more information, see Section 2.2 *The SAS Content Categorization Studio Menus* on page 11.

**Standard toolbar**

click shortcut buttons for some operations. For more information, see Section 2.4 *The Standard Toolbar* on page 21.

**Taxonomy tab**

create, edit, and see the hierarchical structure of the categories and concepts that define your project. For more information, see Section 2.5 *The Taxonomy Tab* on page 23.

**Dependencies tab**

see any forward and reverse dependent relationships between categories, concepts, or both, in your project. For more information, see Section 2.6 *The Dependencies Tab* on page 24.

**Rules (Definition) tab**

write the definitions that classify input documents into categories. When you define concepts, this tab changes to the **Definition** tab. Here you enter the strings that define your concepts. (Strings are defined as the group of words or characters that you specify for a rule.) For more information, see Section 2.7.2 *The Rules or Definition Tab* on page 26.



---

**Testing tab**

test your rules and definitions against the testing set of documents that you assemble. For more information, see Section 2.7.3 *The Testing Tab* on page 31.

**Data tab**

specify the metadata, the paths to testing and training documents, and see other identifying information about your categories and concepts here. For more information, see Section 2.7.4 *The Data Tabs* on page 33.

**Document tab**

see the matches for the tested category or concept in a single tested document. For more information, see Section 2.7.3 *The Testing Tab* on page 31.

## 2.2 The SAS Content Categorization Studio Menus

### 2.2.1 About the Availability of Menus and Menu Selections

All of the following conditions influence whether a menu or menu selection is available to use:

- Your location in the SAS Content Categorization Studio application. For example, some tasks are available only if you select a tab.
- Whether, or not, you created a project.
- The type of model that you are building.
- The selections that you choose.

---

## 2.2.2 About Menus

Menus contain operations that apply to the entire project, or to the currently displayed tab. For example, create a new project, access an existing project, or build a project.

## 2.2.3 The File Menu

Here are the operations that are available in the **File** menu:

### **New Project**

access the New Project window where you name, set the path, and choose a language, for your new project.

### **Open Project**

locate and access an existing project using the Open window that appears.

### **Save Project**

preserve the current project.

### **Save Project As**

save the current project and rename a new, duplicate project.

### **Exit**

close SAS Content Categorization Studio.

The remaining operations are specific to the SAS Content Categorization Collaborative Server. See *SAS Content Categorization Collaborative Server: User's Guide* for more information

## 2.2.4 The Edit Menu

The standard **Undo**, **Redo**, **Cut**, and **Copy** Window commands are located here. The following operations are also included in this menu:

### **Cut All Selections**

remove several taxonomy nodes if you click either the Shift, or the Ctrl, key.

### **Copy All Selections**

copy all of the selected nodes. You can then paste them into a different area of the taxonomy as duplicates of the existing nodes.

---

**Note:** The **Cut All Selections** and **Copy All Selections** operations delete and copy children, as well as parent, nodes.

---

### **Paste**

paste a single node into your taxonomy. If you select a parent node, all of the children (subnodes) of the selected parent are pasted into the taxonomy. See the related operation **Paste Single Node** below.

### **Paste Single Node**

paste *one* copied node into the taxonomy, as a child of the selected parent node.

### **Paste Symbolic Link**

create a placeholder to a category, or concept, in the taxonomy with an at sign (@). For more information, see Section 11.11 *Defining Symbolic Links* on page 336.

### **Paste as Macro**

paste a macro for a Boolean rule to the target category. The target category uses the entire referenced Boolean rule as part of its own rule. For more information, see Section 11.12.2 *Paste a Macro* on page 337.

### **Paste Macro into Rule**

paste a macro into a category rule.

### **Text Find**

locate text in the **Document** tab.

### **Text Replace**

enter text into the Replace window to locate and replace in the **Taxonomy** tab.

### **Tree Find**

use the Find window that appears to search the **Taxonomy** tab for categories and concepts.

### **Tree Replace**

access the Replace window where you enter the text to locate and replace in the **Taxonomy** tab.

### **Find in All Rules**

---

use the Find in All Rules window that appears in order to search for a matching string in the category rules or in concept definitions.

**Options**

access the Options window where you specify the settings that apply to all of the projects created in this installation.

## 2.2.5 The View Menu

Use these commands to hide, or show, the standard **Toolbar** and **Status Bar**. You can also access the following commands:

**Refresh Tree**

update the directory tree in the **Taxonomy** tab when you remove testing messages.

**Taxonomy as Text**

see the taxonomy in text format.

**Number of Taxonomy Nodes**

see a list of the taxonomy nodes and a count of the subnodes in the Number of Taxonomy Nodes window that appears.

## 2.2.6 The Build Menu

The following commands are located in this drop-down menu:

**Build Rulebased Categorizer** (default operation)

build a linguistic or Boolean categorizer that uses category rules to match documents.

**Build Statistical Categorizer**

build a categorizer using a training set of documents.

**Compile Concepts**

build a concept extractor. The **Compile Concepts** tab appears at the bottom of the SAS Content Categorization Studio interface where you can see the results of this operation.

**Upload Categorizer**

---

select a category node and choose this operation to upload your categories to SAS Content Categorization Server. The Upload Categorizer to SAS Content Categorization Server window appears.

**Upload Concepts**

highlight a concept node and select this operation to upload this concept to SAS Content Categorization Server. The Upload Concepts to SAS Content Categorization Server window appears.

**Upload Liti**

highlight a LITI concept node and select this operation to upload this concept to SAS Content Categorization Server. The Upload Concept to SAS Content Categorization Server window appears. (This operation is enabled only when SAS Contextual Extraction Studio is installed and after you define these types of concepts.)

**Abort Compiling Concepts**

stop the process of compiling the concepts. This operation can be used with large concepts projects. When large concepts projects are built, the process of compilation can be lengthily.

## 2.2.7 The Project Menu

The following commands are located in this drop-down menu:

**Add Language**

enable the project to be built in a language that you purchased. When you select this operation, the Select a Language window appears. This window contains a drop-down list of the languages that you purchased.

**Delete Language**

select this operation and a SAS Content Categorization Studio status window appears. You can remove the language applicable to the selected taxonomy node.

---

**Note:** If you click **Yes** in the SAS Content Categorization Studio window, you lose all of the nodes and branches that use this language.

---

**Enable Categorization**

---

add categorization to the project.

**Import Categorization from XML**

create the taxonomy tree from categories defined in SAS Content Categorization Studio XML format.

**Create Categorization from Directories**

use this selection in a new project to import categories from an existing project. A taxonomy tree is created based on the folder hierarchy.

**Remove Categorization**

select a language and choose this operation to remove categories from your project.

**Enable Concepts**

enable concept extraction in this project.

**Remove Concepts**

select the language node in the **Taxonomy** tab and choose this operation to delete all of the concepts in the taxonomy.

**Settings**

specify project-wide settings using the Project Settings window that appears.

## 2.2.8 The Category Menu

The following commands are located in this drop-down menu:

**Add Category**

add a child category below the selected parent category.

**Delete Category**

remove the selected category, with any children, from the taxonomy tree.

**Delete All Selected Categories**

remove all of the categories. Also see **Edit --> Cut All Selections**.

**Rename Category**

type in a different name.

**Import Category from Repository**

use only with SAS Content Categorization Collaborative Server. Import a category that was created in another project.

---

### **Create Directory Tree**

impose a directory structure from your project into a folder on your hard drive. For example, use this operation to store testing and training documents.

### **Generate Subcategories**

set a training path to automatically create subcategories for the selected category.

The following operations are specific to the automatic rule generator tool:

### **Generate Rules Automatically**

generate linguistic rules using the automatic rule generator tool after you set a training path for each category. The **Automatic Rules** tab becomes available on the lower right side of the SAS Content Categorization Studio interface.

### **Export All Generated Rules**

export the generated rules to each category, one at a time. You can also export the generated rules to all of your categories at one time.

### **Clear Generated Rules**

delete the automatically generated rules.

## 2.2.9 The Concept Menu

The following commands are located in this drop-down menu:

### **Add Concept**

add a child node to the parent node that you selected.

### **Delete Concept**

remove a node.

### **Delete All Selected Concepts**

remove all of the selected concepts.

### **Rename Concept**

enter the new name of the selected concept.

### **Sort classifier**

sort the words alphabetically that comprise a classifier definition from top to bottom.

---

**Import Concept from Repository**

import a concept from another project into your current project. Use with SAS Content Categorization Collaborative Server, only.

**Priorities**

access the Concept Priorities window that displays the priority setting for each concept. This setting is specified in the **Priority** field of the **Data** tab.

**Create Directory Tree:**

impose a directory structure from the disk to your project or from the project to disk.

**Generate Suggested Concepts**

use this operation to compare two versions of a concepts taxonomy.

**Import all Suggested Concepts**

bring all of the suggested concepts into your project.

**Clear Suggested Concepts**

remove all of the suggested concepts.

## 2.2.10 The Testing Menu

The following testing operations are located in this drop-down menu:

**Import Test Files**

display the names of your test documents in the **Testing** tab.

**Import Failing Test Files**

bring test documents that could, but should not, pass the test for the selected node into the Testing window to test them. For example, you might want to ensure that the term *server* that applies to a restaurant category does not match a computer category.

**Delete Selected Test File**

remove the test file that you selected from the **Testing** tab.

**Save Test Results**

test the documents against a category and save the results in a file.

**View Saved Results**



---

see the **Saved Results** column with the last set of testing results in the Testing window. You can see this column after you use the save test results operation.

#### **Populate Testing Paths**

use this operation only after you create your taxonomy, write the rules for your categories, and collect a large repository of testing documents. SAS Content Categorization Studio takes the test documents that are located in a central directory and places them into the appropriate folders for each taxonomy node.

---

**Note:** You should also select a **Directory for Unmatched Populate Files** that copies documents without category matches into this folder. (This field is located in the **Misc** tab of the Project Settings window.)

---

#### **Restore Populate Results**

use the **Restore Populate Results** operation to display the testing results if you erased these numbers.

#### **Show Graphical Populate Results**

display the number of documents matched to each category in the **Populate Testing Paths** operation. These results appear in the **Graphical Populate Result** window.

#### **Full Test Report**

test the entire categorizer and generate a report of the results.

#### **Show Last Test Report**

access the Category Test Report window to display the last test results.

## 2.2.11 The Document Menu

The following operations are located in this drop-down menu:

#### **Clear Test Document**

removes the contents of the **Document** tab.

#### **Open Test Document**

makes a test document appear in the **Document** tab.

---

**Save Test Document**

perform a Save operation. This operation copies the testing document that appears in the **Document** tab into the folder of your choice.

**Save Test Document As**

perform a Save as operation. Save the changes in a testing document, shown in the **Document** tab, into the directory of your choice.

**Decrease Font Size**

minimize the size of the font for the displayed test file.

**Increase Font Size**

enlarge the size of the font for the displayed test file.

**Remove Tags**

remove any markup language from the testing document.

**Browser**

use this operation and its suboptions with a Web document in the **Document** tab. These selections are related to the use of the **Browser** selection:

**Forward**

jump to the next page.

**Back**

return to the previous page.

**Refresh**

refresh the current Web page.

**Stop**

stop loading the current page.

**Home**

return to the first page that was loaded into the browser.

## 2.2.12 The Server Menu

Maintain a project on the server when you run SAS Content Categorization Collaborative Server by using these operations. For more information, see *SAS Content Categorization Collaborative Server: User's Guide*.

---

## 2.3 The Status Bar

The **Status Bar** is the horizontal area at the bottom of the SAS Content Categorization Studio interface that indicates the status of the operation that is currently running.

*Display 2-1 Status Bar*

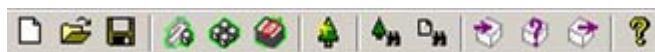


Select **View** —> **Status Bar** to hide, or show, the status bar.

## 2.4 The Standard Toolbar

Use the standard toolbar, located below the menu bar, to access some operations. These standard toolbar icons are shortcuts to some, but not all, of the commands available from the menu bar.

*Display 2-2 Standard Toolbar*



Select or deselect the standard toolbar to hide or show the **Toolbar** operation in the **View** menu.

Table 2-1: Standard Toolbar Buttons











Icon	Command
	Click <b>New</b> and the New Project window appears. Name the project, choose a path, and a language for the new project.
	Click <b>Open</b> and the Choose a project file window appears where you locate an existing project file (.tk2).
	Click <b>Save</b> to preserve the changes to the project.

Table 2-1: Standard Toolbar Buttons (Continued)

Icon	Command
	Click <b>Build Rulebased Categorizer</b> to build the Rulebased categorizer.
	Click <b>Build Statistical Categorizer</b> to build the statistical categorizer.
	Click <b>Compile Concepts</b> to build the .concepts file.
	Click <b>Refresh Tree</b> to clear the testing messages from the taxonomy tree.
	Click <b>Tree Find</b> to access the Tree Find window to search the taxonomy.
	Click <b>Text Find</b> and the Text Find window appears where you can type the text that you want to locate.
	Click the question mark icon to access the <i>SAS Content Categorization Studio: User's Guide</i> .
The remaining buttons are used only for SAS Content Categorization Collaborative Server.	

---

## 2.5 The Taxonomy Tab

By default, the **Taxonomy** tab is displayed when you start SAS Content Categorization Studio. Use this window to see the taxonomy of categories and concepts that you define. If you build your taxonomy with more than one language, an additional language branch is added.

*Display 2-3 Project with Two Languages*



The following nodes appear in the taxonomy:

Sample

name of the project.

English (and Russian-UTF8)

language branch for each language in the taxonomy.

Categorizer

unchangeable node name for the categorizer.

Top

unchangeable node name for the root of each taxonomy.

Concepts

unchangeable node name for concept extraction.

---

**Notes:** Some of the nodes that are listed above appear only after the related functionality is added to the project.

---

Some of the command shortcuts that are available on the menu and standard toolbars, are also accessible when you right-click on a node in the **Taxonomy** tab.

---

## 2.6 The Dependencies Tab

A dependency is created between two category, concept, or both types of nodes when one node uses the entire rule of another node as part of its rule. Use the **Dependencies** tab to see these dependent relationships.

*Figure 2-1 Dependencies Window*



The **Dependencies** tab provides these views:

### Forward

display the target node above the source node. The target node is the category or concept that uses the entire rule of the source category or concept as part of its rule.

### Reverse

display the source category or concept above the target category. The source node is the category or concept whose rule is used by the target category.

If you create dependencies, use the **Dependencies** tab to check for the interrelated rules before you delete or change a source category. If you delete a

---

dependent node, unexpected behaviors might occur. For more information, see Section 11.12 *Dependencies between Categories or Categories and Classifier Concepts* on page 337 and Section 19.5.4 *Defining Dependencies in Grammar Rules* on page 487.

---

**Hint:** Highlight a *source* or a *target* category in the **Dependencies** tab and click the **Taxonomy** tab. The same category is highlighted in the **Taxonomy** tab.

---

## 2.7 The Right Window Tabs

### 2.7.1 About the Tabs

The following tabs are located on the bottom right-hand side of the user interface. Click these tabs to develop rules, enter data, test the taxonomy, and so on.

*Display 2-4 Category Tabs*



The **Automatic Rule** tab appears only when the automatic rule generator tool is used. For more information, see Section 7.5.2 *Automatically Generate Rules* on page 205. Select **Category --> Generate Rules Automatically** to see this tab. To hide this tab, select **Category --> Clear Generated Rules**.

If you selected a concept in the **Taxonomy** tab, the **Rules** tab changes to the **Definition** tab.

*Display 2-5 Concept Tabs*



---

**Note:** There is no automatic rule generator tool for concepts.

---

---

The table below describes the components of these tabs:

Table 2-2: Window Tab Commands

Tab	Category or Concept	Purpose
Rules	category	Write linguistic or Boolean rules.
Definition	concept	Write concept rules.
Testing	both	Test documents against the rules. The components of the <b>Testing</b> tab are the same for both categories and concepts. However, the <b>Concordance</b> selections are available only for concepts. The test result returns are computed differently for categories or concepts.
Data	both	Enter metadata for categories and concepts. Some of the fields are informational only and do not affect the behavior of the categorizer or the concept extractor. Other fields determine how the <language>.mco (categories) and <language>.concepts (concepts) files are built. Although many of the components for both tabs are the same, there are some differences.
Document	both	See the testing results for one document. The Document window becomes the concordance window when the <b>Concordance</b> check box is selected with either <b>Selected concept</b> or <b>All concepts</b> . The concordance window is available only for concepts.
Automatic Rule	category	Display the rules that are automatically generated from your training set of documents.  <b>Note:</b> This tab only appears after you use the automatic rule generator tool.

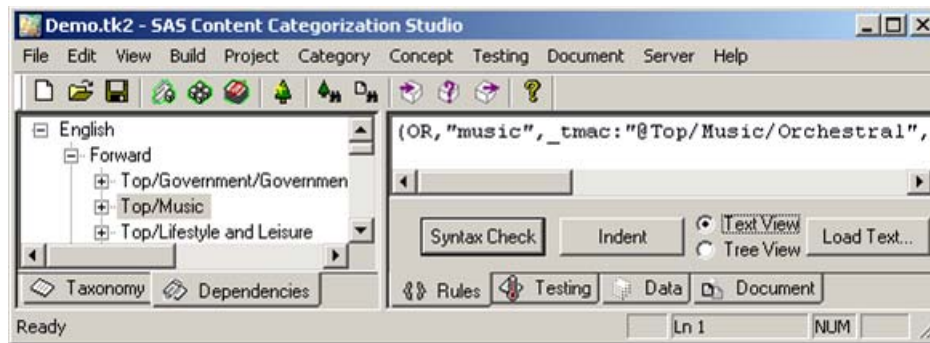
## 2.7.2 The Rules or Definition Tab

### 2.7.2.A The Rules Tab

The **Rules** tab displays the area where you write and review your rules for the rule-based categorizer. The default mode is text view.

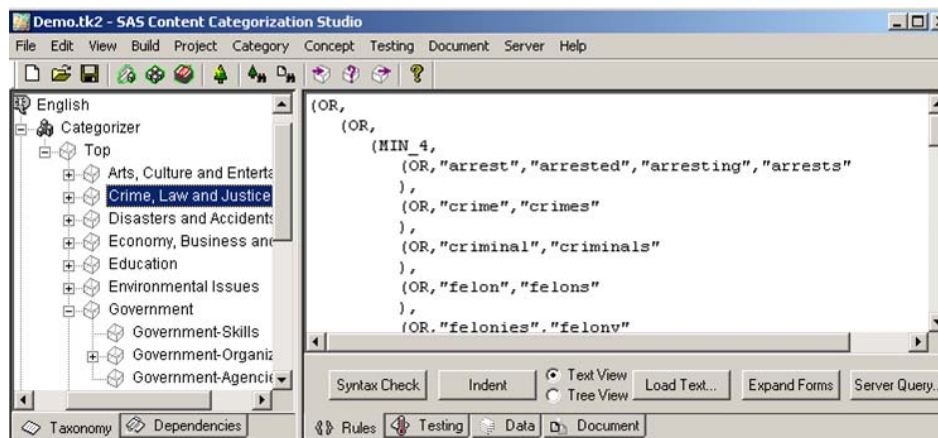


Display 2-6 Text View



Click **Indent** in the Text View mode to see a Boolean rule. This selection is available only for Boolean rules. The rule is displayed in tree format.

Display 2-7 Indented Boolean Rule



Click **Tree View** to see a Boolean category rule in a taxonomy format. This selection works only with Boolean rules.

---

## Display 2-8 Tree View



Use any of the operations that are available in the **Rules** tab when you write category rules:

### Syntax Check

check the grammar of your definition. The **Category Syntax Check** tab appears at the bottom of the SAS Content Categorization Studio interface. This window displays a message about the status of the grammar.

### Indent

(only available for Boolean rules) see the rule in an indented style. Each new line begins with either an opening ( ( ) or the closing ( ) parenthesis that is used to qualify the respective Boolean operator.

### Load Text

load the linguistic or Boolean rules that you developed using another file into the **Rules** tab. For example, write your rules in *Notepad*.

### Expand Forms

Use this operation when you append an at sign (@) to the end of a word in a Boolean rule. This symbol makes it possible to expand the word into its grammatical possibilities. These forms are displayed only within the Boolean category rule.

---

**Note:** You can treat each *word@* as a literal to be matched at run time. To do this, append an @ sign to a word in your category rule and select **Never expand word forms** in the **Category** tab of the Project Settings window.

---

#### Server Query

query an index using a Boolean rule.

#### Text View

see your rules as a single line of text.

#### Ln

see the line number where your cursor is located. For example, your cursor might appear on Ln 56.

---

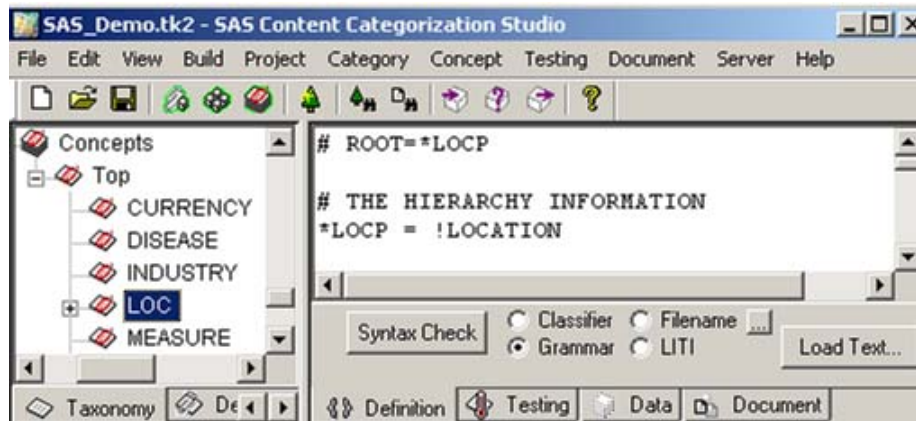
**Note:** You can see the buttons listed above in the **Text View** mode. You can use the **Expand Forms**, **Load Text**, and **Server Query** buttons in the **Tree View** mode only.

---

### 2.7.2.B The Definition Tab

Use the **Definition** tab to specify the rule for the selected concept.

*Display 2-9 Definition Tab*



---

Use the buttons in the **Definition** tab when you define your concepts:

**Syntax Check**

check the syntax of a definition in the Concept Syntax Check window that appears.

**Classifier**

specify a classifier concept.

**Grammar**

write a grammar definition.

**Filename**

set the path to an existing classifier definition using this radio button and



. Use this operation when you have a large classifier definition. For example, when the classifier has one million lines.

**... (ellipsis button)**

locate a file on your machine, after you select the **Filename** radio button above.

---

**Note:** The filename operation only works when you define a classifier concept. Unlike the **Load Text** operation, the name of the file and path to that file are loaded into the **Definition** tab.

---

**LITI**

use this selection to write contextual extraction concept definitions. If you do not install SAS Contextual Extraction Studio, the **LITI** radio button is not enabled.

**Load Text**

load the full text of a file into the **Definition** tab as your concept definition. For example, write a complex definition using a `.txt` document. Click **Load Text** to access the Open window where you can locate the definition text that you want to load into the **Definition** tab.

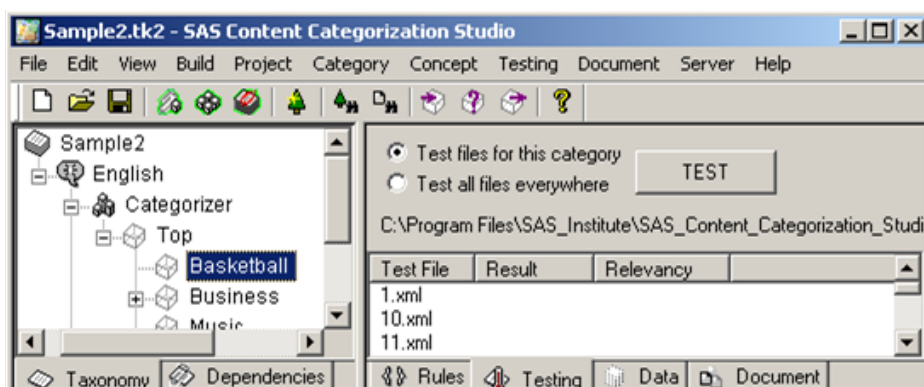
---

## 2.7.3 The Testing Tab

### 2.7.3.A The Testing Tab for Categories

Use the **Testing** tab to select and test a single category against a set of test documents.

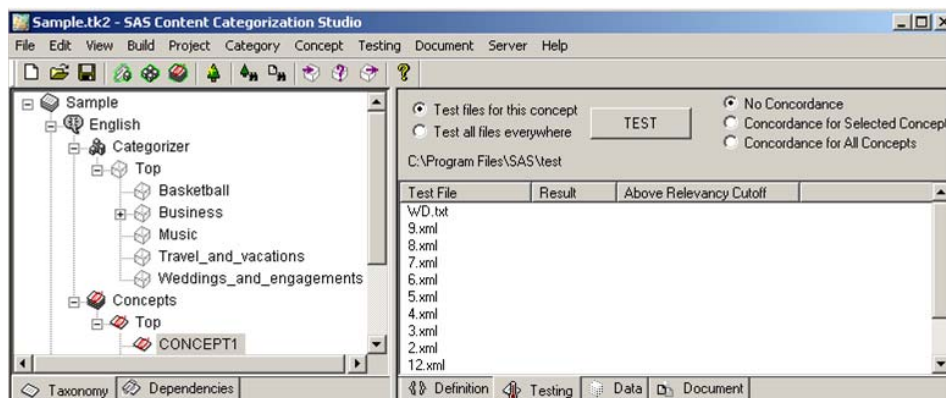
*Display 2-10 Testing Tab for Categories*



### 2.7.3.B The Testing Tab for Concepts

Use the **Testing** tab for concepts to check the accuracy of a concept definition against a set of test documents.

*Display 2-11 Testing Tab for Concepts*



---

### 2.7.3.C The Testing Tab Components

The operations that are available in the **Testing** tabs for both categories and concepts are explained below:

#### **Test files for this category (or concept)**

test only the test files that are mapped to this node in the **Data** tab.

#### **Test all files everywhere**

test all of the files in the testing repository against the selected category or concept. This operation expands the testing process to simulate real-time results.

#### **TEST**

start the testing process and SAS Content Categorization Studio displays the results in the **Testing** window.

The path to the testing file is displayed below these operations and above the following headings:

#### **Test File**

see a list of the names of all of the test files below this heading. (This list appears after you specify the path to the testing directory in the Data window.)

#### **Result**

These are the possible test results for categories:

PASS

this test document meets the rule requirements. In the case of linguistic rules, only documents that meet the match ratio specification are assigned this tag.

PASS\*

the test file *conditionally* passes, because the document fell below the relevancy requirements.

FAIL

this document fell below the relevancy requirements or, in the case of linguistic rules, the match ratio setting.

For concepts, this column displays the number of matches for this concept definition.

---

### **Relevancy or Above Relevancy Cutoff**

See the relevancy scores for all passing categories displayed beneath the **Relevancy** heading. The number of matching terms that exceed the number specified by the relevancy cutoff setting for concepts is listed beneath **Above Relevancy Cutoff**.

The operations that are available only for concepts in the **Testing** tabs are explained below:

#### **No concordance**

(default) no concordance operations are performed.

#### **Concordance for Selected Concept**

display the terms that match the selected concept in the input document in the concordance window.

#### **Concordance for All Concepts**

display the matched concepts for all of the terms in your definitions. These terms appear in the concordance window, with the names of the concepts that they match.

## 2.7.4 The Data Tabs

### 2.7.4.A The Data Tab for Categories

Use the **Data** tab for categories to enter metadata, testing and training paths, and other information about each category.

Figure 2-2 Data Window for Categories

ory Concept Testing Document Server Help

Fill in these fields at this time, or as you build your project.

ID:  Author  Created:  Modified:

Relevancy Cutoff  Relevancy Bias  Category Bias  Match Ratio  0

☒ Completed  
☐ Pending  
☐ Test Disabled

Description

Thesaurus

Use these fields to query an index.

Query

Comments

Related Links

Use the Testing Path and the Propagate Options for testing.

Testing Path  ... Propagate

Propagate Options  
☐ Identical Path  
☐ Create Folders

Training Path  ... Propagate

Rules

Use the Training Path for the statistical categorizer and the automatic rule generator tool, only.



---

### 2.7.4.B The Data Tab for Concepts

Use the **Data** tab for concepts to enter metadata, the testing path, and other information about each concept.

*Figure 2-3 Data Tab for Concepts*

Fill in these fields at your discretion.

ID: \_\_\_\_\_ Author: \_\_\_\_\_ Created: \_\_\_\_\_ Modified: \_\_\_\_\_

Relevance Cutoff: \_\_\_\_\_ Priority: 0

Completed ☐ Pending ☐ Test Disabled ☐

For classifier concepts.

☐ Case Sensitive Matching ☐ Case Insensitive Matching ☒ Use Project Default

Description: \_\_\_\_\_

Thesaurus: \_\_\_\_\_

Query: \_\_\_\_\_

Comments: \_\_\_\_\_

Related Links: \_\_\_\_\_

Testing Path: \_\_\_\_\_ Propagate

Use the Testing Path field and the Propagate Options for testing.

Propagate Options

☐ Identical Path ☐ Create Folders

Definition Testing **Data** Document

Compare the **Data** tab components for categories and concepts. These settings affect other settings and testing results.

Table 2-3: Data Window Components

Com- ponent	Category or Concept	Description	Informa- tional
ID	both	(Optional) Track categories and concepts using a unique identification number for each node.  <b>Note:</b> Select <b>Allow Duplicate ID's</b> in the Project Settings - Concept window and you can enter the same identifier multiple times.	
Author	both	Specify the name of the person who created the category or concept.	X
Created	both	Specify the development date for this node. This date is automatically entered for you.	X
Modified	both	See the automatically entered date when this concept, or category, was last modified.	X
Relevancy Cutoff	both	Specify the minimum threshold for frequency-based ranking. Unless this number of instances of matching terms appears in the text, a match does not occur.  <b>Note:</b> (categories only) If a document has a relevancy number below this specification, the document is considered <i>conditionally</i> passing (PASS*).	
Relevancy Bias	category	(Default: 1) Specify the number that is multiplied by all of the relevancy scores for this category to boost its relevancy in relation to the other categories in the taxonomy. This setting applies to both linguistic and Boolean rules and is used when third-party software is not a concern.	
Category Bias	category	(Default: 0) Specify a number that is multiplied by the <b>Default Category Bias</b> setting in the Project Settings - Category window. When specified, this number is used to boost the frequency-based relevancy score for third-party software. For this reason, this setting is typically used with category rules that are defined by one term only. (If you reset the <b>Default Category Bias</b> setting in the <b>Category</b> tab, the number specified in the <b>Category Bias</b> field is added to this number.)	

Table 2-3: Data Window Components (Continued)

Com- ponent	Category or Concept	Description	Informa- tional
Match Ratio	category	(Default: 10%, used with linguistic rules only) Specify a percentage of the terms in the linguistic category rule that is the threshold necessary to make a document a category match. This field can be set individually for each category in the taxonomy. SAS Content Categorization Studio uses this setting, internally, to convert linguistic rules to Boolean rules.  <b>Note:</b> These symbols +, **, and --, override the match ratio setting.	
Priority	concept	(Default: 0) Determine the matching concept when a document matches more than one concept.	
Completed	both	(Default) Flag this node as finished.	
Pending	both	Signal that this node requires more work. This specification does not affect the <language>.mco or the <language>.concepts file. This setting is used only in the metadata.	
Test Disabled	both	Define helper categories or concepts that are evaluated but not exposed to the user. This is a useful flag to use when you create macros.	
Case Sensitive Matching	classifier concepts	Match a string in an input document that is an exact match for both the specified text and case.	
Case Insensitive Matching	classifier concepts	Locate a match on a string in an input document when the text of the string is a match, regardless of the case specified by the concept.	
Use Project Default	classifier concepts	(Default setting) Use the case sensitivity setting specified in the Project Settings - Concept window.	
Description	both	Explain the purpose of this category or concept.	X
Thesaurus	both	Specify a comma-separated (,) list of words that are alternative names (synonyms) for the category or concept. A search on an alternate name matches the related category or concept.	

Table 2-3: Data Window Components (Continued)

Com- ponent	Category or Concept	Description	Informa- tional
Query	both	Enter a search term that locates documents related to this topic in an index.	
Comments	Both	Enter explanations or notes.	X
Related Links	both	Specify URLs that contain related information. This text functions like a <i>see also</i> list.	X
Testing Path	both	Enter the pathname to the directory that contains the testing documents that are used to refine this category rule or concept definition.	
Training Path	category	Enter the pathname of the directory that contains training documents for this category. <b>Note:</b> This field is used only for the statistical categorizer and the automatic rule generator tool. Both of these categorizers generate rules automatically.	
Propagate button	both	Set the testing or training paths.	
Propagate Options	both	Use either, or both, of the operations under this heading:  <b>Identical Path</b>  specifies testing paths to the same repository of testing documents.  <b>Create Folders</b>  Automatically create folders for all of the child categories or concepts.	

---

## 2.7.5 The Document and Concordance Tabs

### 2.7.5.A About the Document Tab

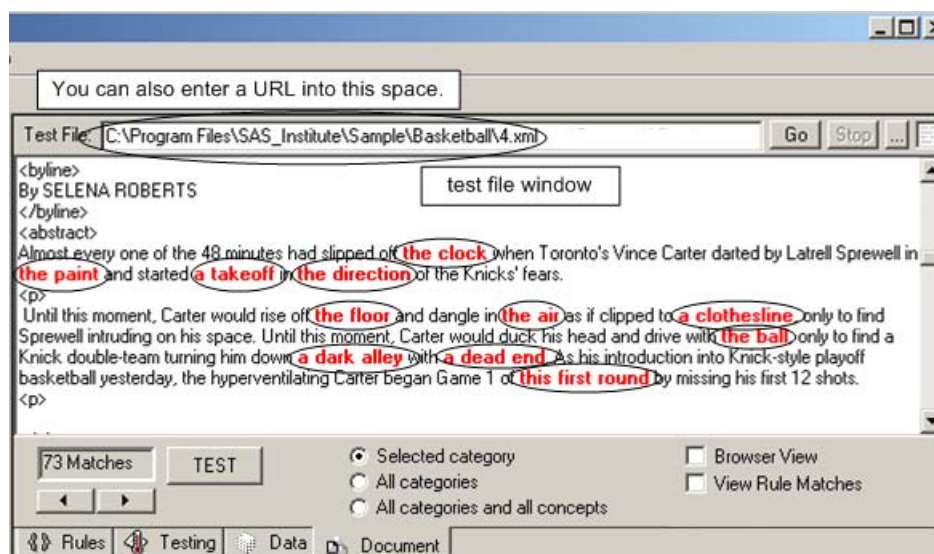
The **Document** tab is used to test the text of a test document, a Web page, or text that you type, or copy and paste, into this window. There is a 1 MB limit for text that is typed, or copied and pasted into this window.

If you choose to edit the text in a document that you test in the Document window, you can save this change in the original file. For more information, see Section 14.6 *Editing a Document in the Document Tab* on page 394.

### 2.7.5.B The Document Tab for Categories

The **Document** tab displays the matching category rule terms in red in the selected input document.

Figure 2-4 Document Tab



Select one of the following test operations in the **Document** tab and see the results in the input document:

---

**Selected category**

(default setting) test the text that appears in the **Document** tab against the selected category.

**All categories**

test the selected document against all of the categories in the taxonomy.



**All categories and all concepts**

test the document against all of the categories and concepts in this project.

**View Rule Matches**

test the document against one category only and see the results in the Rule Matches window.

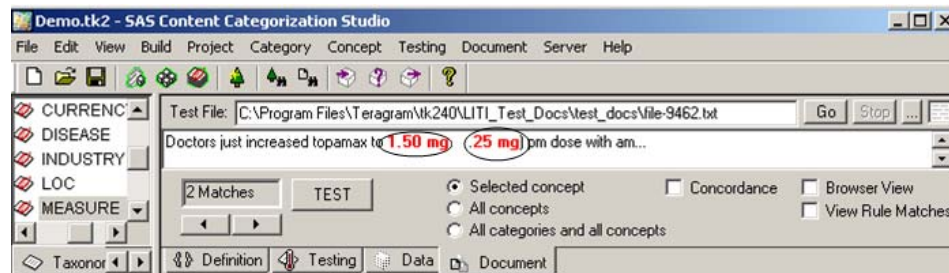
After you select one of the operations listed above, click **TEST** and see the following test results for the selected document:

- The matching terms are highlighted in red for **All categories** or **All categories and concepts**.
- The **PASS** or **FAIL** result is displayed in the testing field to the left of the **TEST** button.
- Jump to the preceding or following match when you click either the  or  arrow.
- Select the **View Rule Matches** check box to see the rule matches in a hierarchical format. This selection is available only when you test a selected category that is defined by a Boolean rule. When you make this selection, the Rule Matches window appears. For more information, see Section 2.12.11 *The Rule Matches Window* on page 85.

### 2.7.5.C The Document Tab for Concepts

The **Document** tab highlights the matching concept rule terms in an input document. This tab is similar to the **Document** tab for categories, but it specifies concepts instead of categories. The **Document** tab for concepts also includes the concordance selection. The concordance view enables you to see matching terms in list format.

Figure 2-5 Document Tab for Concepts



Select one of the following test operations in the **Document** tab and see the results in the input document:

**Selected concept**

(default setting) test the text that appears in the **Document** tab against the selected concept.

**All concepts**

test the selected document against all of the categories, or concepts, in the taxonomy.

**Concordance**

test the document against all of the categories and concepts in this project.

**View Rule Matches**

test the document against one category only and see the results in the Rule Matches window.

After you select one of the operations listed above, click **TEST** and see the following test results for the selected document:

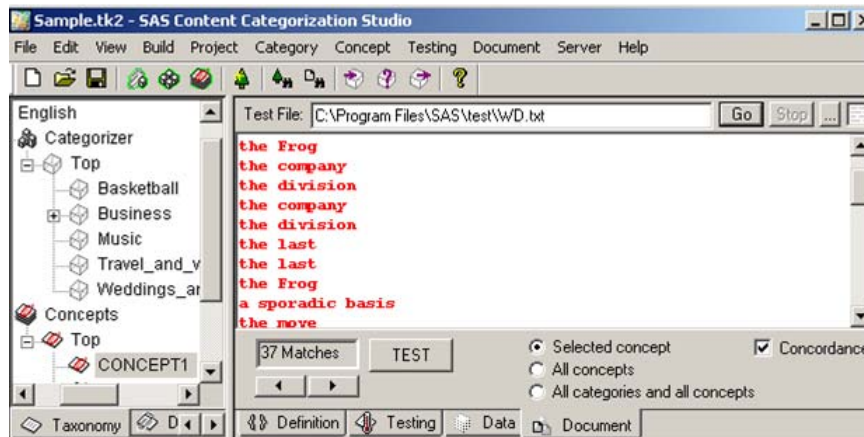
- The matching terms are highlighted in red.
- The PASS or FAIL result is displayed in the testing field to the left of the **TEST** button.

## 2.7.5.D The Document Tab as Concordance for Concepts

A concordance is an ordered list of matched terms for the selected concept. You specify this ordering in the Project Settings - Concordance window. The concordance view displays only the terms that match the concept definition in the **Document** tab. These terms are highlighted in red.

---

## Display 2-12 Concordance View



When you select **Concordance**, the test results displayed depend on the other selection that you make. Choose one of the following combinations:

### **Concordance for Selected Concept Only**

display the terms that match the selected concept in the input document in the concordance window that appears.

### **Concordance for All Concepts**

display the matched concepts for all of the terms in your definitions. These terms appear in the concordance window, with the names of the concepts that they match.

Select **Concordance** and click **TEST**. The concordance view appears in the **Document** tab. Definition matches appear in list format and the Best Matches window displays a total count of the matches. For more information, see Section 2.7.5.D *The Document Tab as Concordance for Concepts* on page 41.



---

### 2.7.5.E The Document Tab as Browser Interface

The **Document** tab can also be used as a Web browser to test Web documents.

*Display 2-13 Document Window as Web Browser*



To test a Web document, select **Browser View**. Web documents can be tested only against categories that are defined by a Boolean rule. When you make this selection, you can also use the Best Matches window to see the total count of the matches. For more information, see Section 14.3.2 *Load and Test the Source Document* on page 390 and Section 14.5 *See the Best Matches* on page 393.

---

### 2.7.5.F The Document Tab Components

The components of the **Document** tab enable you to test one text using several operations. Use the information in the following table to determine how to use each component of this window. Some of the selections work only with categories or concepts, but not with both.

Table 2-4: Document Tab Components





Field or button	Category or Concept	Description
Test File field	both	Specify one of the following operations: <ul style="list-style-type: none"><li>- a path to a document</li><li>- a URL to test a Web page</li></ul>
Go	both	Begin loading the document.
Stop	both	Stop loading the document.
	both	Use the Open window that appears to locate the document on your machine that you want to test.
	both	When active, SAS Content Categorization Studio is loading a Web page into the <b>Document</b> tab.
test file window	both	Use the test file window to perform one of the following operations: <ul style="list-style-type: none"><li>- See a tested document: Double-click on a single document in the <b>Testing</b> tab and the <b>Document</b> tab appears. The matching rule, or definition, terms are highlighted in red in the test document.</li><li>- Test a single document: Open a text in the <b>Document</b> tab when you specify the path to the document using the <b>Test File</b> field or enter a URL.</li></ul>
status window	both	See the status of the document, or the number of matches, for the selected category or concept. The status window is located to the left of the <b>TEST</b> button.
	both	Navigate through the matched category or concept terms in the tested document when you click the forward and backward buttons.

Table 2-4: Document Tab Components (Continued)

Field or button	Category or Concept	Description
TEST button	both	Test the loaded document.
Selected category or concept	both	Test only against the selected category or concept.
All categories or All concepts	both	Test this document against all of the categories or concepts in this project.  <b>Note:</b> Select this radio button and click <b>Test</b> . The Best Matches window appears. This window displays a list of category, or concept, matches ordered from best (top of the list) to worst (bottom). This is true if you select <b>Show best matches when testing all</b> in the <b>Edit --&gt; Options</b> window.
All categories and all concepts	both	Test this document against all of the categories and concepts in your taxonomy.  <b>Note:</b> If you also select <b>Edit --&gt; Options</b> and choose <b>Show best matches when testing all</b> , the Best Matches window appears. See a list of the matches ordered from best (top of the list) to worst (bottom).
Browser View 	both	Select this operation and these buttons appear in the lower right-hand side of this interface:  <ul style="list-style-type: none"> <li>- <b>Home:</b> Go to the home page.</li> <li>- <b>Back:</b> Return to the last viewed page.</li> <li>- <b>Forward:</b> Go to the next page.</li> <li>- <b>Refresh:</b> Update the Web page.</li> <li>- <b>Stop:</b> End the current process.</li> </ul>
View Rule Matches	categories with Boolean rules, only	(applies to the <b>Selected category</b> operation, only). The Rule Matches window appears and displays the terms that match the category in tree format.
Remove Tags	both	See the text of the selected Web page without any mark-up tags.

---

## 2.7.6 The Automatic Rule Tab

See the rules that are automatically generated by the automatic rule generator tool in the **Automatic Rule** tab. Use this window to see the terms that SAS Content Categorization Studio extracted from the training documents and to export the rule to the **Rules** tab.

To automatically generate category rules, complete these steps:

1. Select **Category --> Generate Rules Automatically** and the **Automatic Rule** tab appears.



2. After you examine your rule, click **Export** to copy the rule for the selected category into the **Rules** tab.
3. Edit the rule terms in the **Rules** tab.
4. To clear these rules, select **Category --> Clear Generated Rules**.

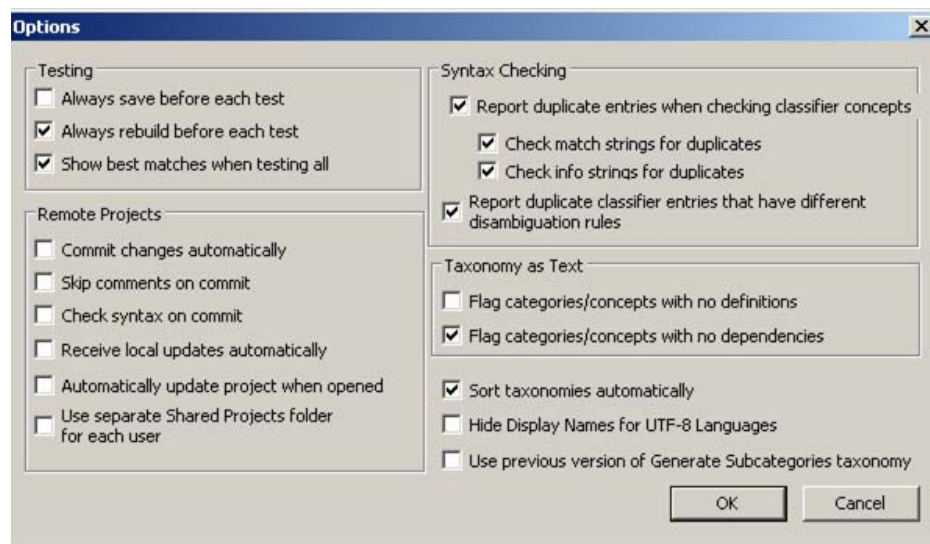
---

## 2.8 The Options Window

The Options window enables you to automate certain operations that apply to the *installation* of SAS Content Categorization Studio. These settings affect all of the projects created with this installation. Project settings, on the other hand, are *taxonomy-specific*. For more information, see Section 2.9 *The Project Settings Windows* on page 49.

The default settings in the Options window are shown below:

Figure 2-6 Options Window



---

**Hint:** The operations that are available under **Remote Projects** are available only after you also install SAS Content Categorization Collaborative Server.

---

The operations in the Options window apply to all SAS Content Categorization Studio projects. This is true whether you choose to combine both categories and concepts within one project or to develop a categories-only, or a concepts-only, project:

### Testing

---

**Always save before each test**

(default) automatically save the project before each testing operation.

**Always rebuild before each test**

(default) automatically rebuild the project binary file before each test. If the categorizer or concepts extractor is not up-to-date, a recompile operation is performed. Then the testing operations are run.

**Show best matches when testing all**

the Best Matches window appears when you test either **All categories** or **All categories and concepts** in the **Document** tab.

**Syntax Checking****Report duplicate entries when checking classifier concepts**

select this operation when you choose one, or both of the selections below.

**Check match strings for duplicates**

examine only the match part of the classifier concept definition.

**Check info strings for duplicates**

examine only the information part of the classifier concept definition.

**Report duplicate classifier entries that have different disambiguation rules**

locate duplicates. Also select **Check match strings for duplicates**, **Check info strings for duplicates**, or both.

**Taxonomy as Text****Flag categories/concepts with no definitions**

see these categories and concepts in the Notepad window that appears.

**Flag categories/concepts with no dependencies**

see these categories and concepts in the Notepad window that appears.

**Sort taxonomies automatically**

(default) sort each branch of the taxonomy alphabetically, beginning with the letter A.

---

**Note:** Click the plus sign (+) to the left of the *Categorizer*, *Concepts*, *Top*, *language*, and *project name* nodes. This

---

action enables you to see the reordered taxonomy after each of these nodes is closed and reopened.

---

**Hide Display Names for UTF-8 Languages**

(applies to UTF-8 languages, only) display the Latin-1 internal category names, while the UTF-8 names are hidden. This operation works in coordination with the Enter Names window.

**Use previous version of Generate Subcategories taxonomy**

use the previous taxonomy, instead of the Wikipedia taxonomy, to generate subcategories.

## 2.9 The Project Settings Windows

### 2.9.1 About Project Settings

Use the Project Settings windows to set taxonomy-wide operations. If you choose to develop a SAS Content Categorization Studio project that uses more than one language, set the project settings for each language taxonomy separately. Project settings differ from options. Options are *installation* specific. For more information, see Section 2.8 *The Options Window* on page 47.

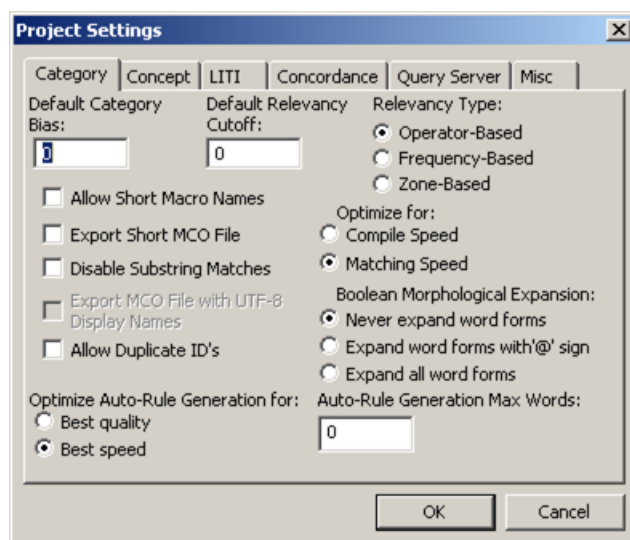
You can specify some of the project settings as you add categories or concepts to the taxonomy. For more information, see Section 2.7 *Specifying Project Settings* on page 130. Modify these settings after testing or during the various stages of project development. For example, change your project settings if you do not obtain the testing results that you require.

### 2.9.2 The Project Settings for Categories

#### 2.9.2.A The Category Tab

Use the Category tab to set project-wide settings for your categorizer.

Display 2-14 Category Tab



Choose the following operations in the **Category** tab to customize the results returned by your project:

#### Default Category Bias

(default: 0) assign more weight to your categories. Use this setting to boost the relevancy of your categories into the range used by a third-party software application. If a number is entered into this field, this number is multiplied by the **Category Bias** setting in the **Data** tab.

#### Default Relevancy Cutoff

(default: 0) true, unless you specified another number for the **Relevancy Cutoff** field in the **Data** tab. Specify the minimum relevancy required for a document to be a match on the selected category. This setting applies to each of the relevancy types.

#### Relevancy Type

Specify the type of relevancy that is used to determine the category that is the best match for an input document:

##### Operator-Based

(default) Boolean operators, and matching terms that these operators modify, determine the category matches. (Linguistic rules are



---

converted to Boolean rules before you test. This conversion is an automatic, internal operation.)

#### **Frequency-Based**

the number of matching terms in a document determines the degree of its relevancy for a specific category.

#### **Zone-Based**

weight matches that occur in certain sections of an input document more heavily than matches in other areas.

#### **Allow Short Macro Names**

(Boolean rules, only) enable the use of short macro names in Boolean rules. You specify macro names with the `_tmac` symbol such as:

```
(OR, _tmac: "@Top/Music/Baroque/Composers")
```

By default, the unique name of a category is its full path such as:

```
Top/Music/Baroque/Composers
```

When you specify the **Short Macro Names** operation, you can refer to the short form of the category name in a macro rule. For example, you can specify the following syntax:

```
(OR, _tmac: "@Composers")
```

#### **Export Short MCO file**

produce a `*.short.mco` file. This is a categorization binary file where the category names that are returned are the short paths, instead of the full pathnames.

#### **Disable Substring Matches**

prevent a partial match on a string that defines a category rule. For example, if *business processes* and *business* are specified in the rule, a match is not returned for the word *business*. This is true unless the word *processes* immediately follows *business*.

#### **Export MCO file with UTF-8 Display Names**

build a taxonomy using a UTF-8 language and use this operation to display the UTF-8 names in the category binary file. An additional `<language>.mco` file is created in the following format:

```
<language>.utf8.mco
```

The `.mco` file contains the Latin-1 internal names. The `<language>.utf8.mco` file enables you to see the taxonomy in the UTF-8

---

language that appears in the **Taxonomy** tab. For example, if you create a taxonomy structure of categories using Japanese, you might see the following line of text instead of `Top/School`:

`Top/学校`

#### **Allow Duplicate ID's**

enable duplicate identification numbers to be entered into the ID field of the **Data** tab for categories. Otherwise, ensure that the identification numbers are unique.

#### **Optimize Auto-Rule Generation for**

optimize automatic rule generation when you use one of the following selections:

##### **Best Quality**

(default) apply a new algorithm that returns more accurate rules.

##### **Best Speed**

deploy the standard automatic rule generation algorithm.

#### **Optimize for**

leave the default setting, unless you are building a taxonomy with thousands of nodes. In this case, select **Best Quality**:

##### **Compile Speed**

prioritize category building.

##### **Matching Speed**

(default) prioritize category matching.

---

**Note:** This setting can be different from that specified in the **Concept** tab of the Project Settings window. This is true only if you build a taxonomy that uses both categories and concepts.

---

#### **Boolean Morphological Expansion**

(Boolean rules, only) select one of these choices to specify the type of word form expansion:

**Never expand word forms**

---

(default) matches occur only on the words that explicitly appear in rules. Words that are followed by an at sign (@) are treated as literals to be matched. For example, `run@` matches only `run@` in an incoming text. If the words *run* and *running* also appear in this text stream, they are not matched.

#### **Expand word forms with '@' sign**

expand only the words followed by an @ sign during the compile operation. The expanded forms appear in the `.mco` file. When the word ends with the following symbols, expansion is applied:

@: both noun and verb forms

@V: verb forms only

@N: noun forms only

#### **Expand all word forms**

treat every word in a category rule as if it ended with an @ sign when your project is compiled. The expanded word forms appear in the `.mco` file.

---

**Note:** You can use the **Expand Forms** button in the **Rules** tab to see any expansions that you are unsure about before you compile your project. In this case, select **Edit --> Undo** to return the @ signs.

---

#### **Auto-Rule Generation Max Words**

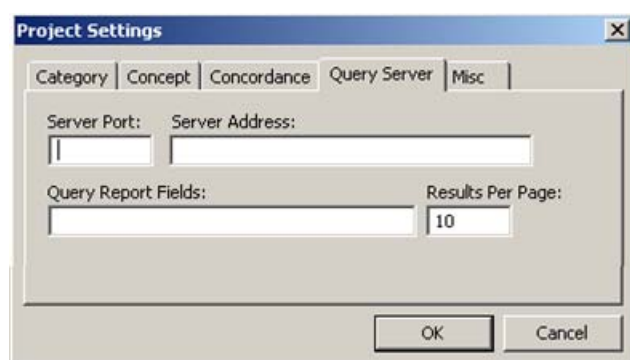
(default: 0, means that there is no upper bound on the number of words returned as a match) specify a maximum number of words or phrases for an automatically generated rule.

### **2.9.2.B The Query Server Tab**

The **Query Server** tab in the Project Settings window enables you to use a search engine to find documents with this topic. This operation is enabled for projects that include categories, and is used to set the project-wide settings for the project.

---

Display 2-15 Query Server Tab

The image shows a screenshot of the 'Project Settings' dialog box with the 'Query Server' tab selected. The dialog has a title bar with a close button. Below the title bar are five tabs: 'Category', 'Concept', 'Concordance', 'Query Server', and 'Misc'. The 'Query Server' tab is active. It contains four input fields: 'Server Port:' (a small text box), 'Server Address:' (a larger text box), 'Query Report Fields:' (a text box), and 'Results Per Page:' (a spinner box set to '10'). At the bottom right are 'OK' and 'Cancel' buttons.

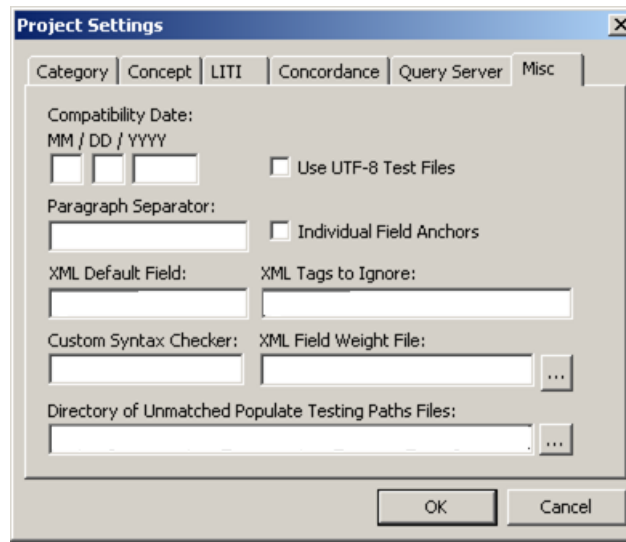
### Server Port

To specify a query server port, complete these steps:

1. Type the port number into the **Server Port** field.
2. Specify the address for the server port in the **Server Address** field.
3. (Boolean rules, only) Type the field names that set the values that are returned in the query report into the field below **Query Report Fields**. These fields are also the XML tags for the stored documents on the server.
4. (Default: 10) Specify the number of results that can be returned in each results page in the **Results per Page** field.

## 2.9.3 The Misc(ellaneous) Tab for Categories and Concepts

Use the **Misc** tab in the Project Settings window to specify the various settings that apply to both the categorizer and the concepts extractor.



Use the **Misc** tab to specify settings for both categories and concepts:

**Compatibility Date**

Use the MM/DD/YYYY settings to set the compatibility date for the .mco and the .concepts files that are automatically generated by SAS Content Categorization Studio. Use this field only with deprecated versions of this application. Enter the date of the older version of SAS Content Categorization Server that you are running. SAS Content Categorization Studio generates a binary file (.mco or .concepts) that is compatible with the older version of SAS Content Categorization Server. Use this operation until you have time to install and run a newer version of SAS Content Categorization Server.

**Use UTF-8 Test Files**

select this field when your testing documents are in UTF-8 format, but the language of the categorizer might not be UTF-8.

**Paragraph Separator**

(only for the rule-based categorizer that uses Boolean rules and for some concepts) enter the string that is used as a paragraph separator within your documents. For example, type <P>.

---

### **Individual Field Anchors**

specify this setting with Boolean category rules and with classifier concepts that use disambiguation. By default, if you have more than one instance of the same XML tag in a Web document, SAS Content Categorization Studio collapses the sections into one searchable area. When you select this check box, each section of a Web-based document is searched separately. This feature has implications for some Boolean operators. For more information, see Section 11.8.4 *How to Use Project Settings with Structured Text* on page 318.

Use the following **Misc** tab settings for categories only:

#### **XML Default Field**

specify one or more XML fields when you write your category rule to limit search to the specified field. If you leave this field blank, all of the XML fields in the input XML document are searched.

#### **XML Tags to Ignore**

choose to exclude one or more XML fields when processing your XML documents.

#### **Custom Syntax Checker Executable**

specify the path to an external, custom grammar checker program that is used in place of the internal syntax checker program.

#### **XML Field Weight File**

specify the location of a text file that weights the XML tags in input documents.

#### **Directory for Unmatched Populate Files**

set the path to the testing documents that do not match any categories in your taxonomy.

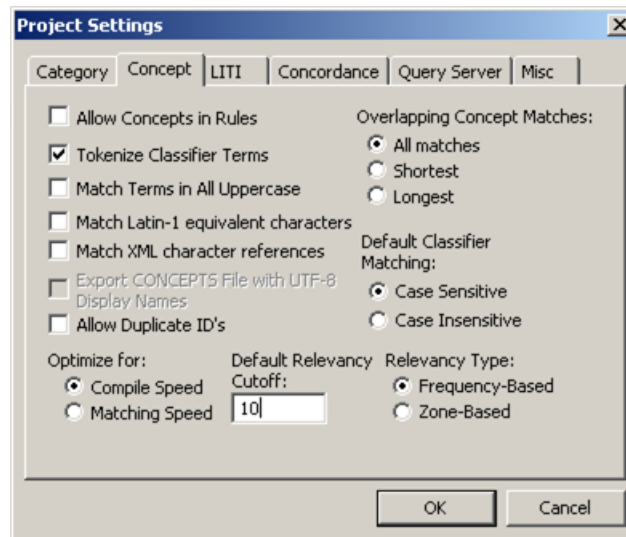
---

## 2.9.4 The Project Settings for Concepts

### 2.9.4.A The Concept Tab

Select the Project Settings - Concept window to set project-wide settings for concept matching. The default settings are displayed below.

*Display 2-17 Concept Tab*



To customize concept matches, use the following settings:

#### **Allow Concepts in Rules**

Use classifier concepts in category rules for the purpose of creating dependencies.

---

**Note:** This check box is available only when you enable both categories and concepts.

---

---

### Tokenize Classifier Terms

(default setting) enable SAS Content Categorization Studio to automatically break the definition text of classifier concepts into words. This default setting should be maintained for new projects.

---

**Note:** Turn off this operation if you choose to use a backslash (\) instead of a space between terms in a concept definition.

---

### Match Terms in All Uppercase

add all uppercase versions of the specified rule terms to the classifier rule. For example, a rule containing the word *Cat* adds *CAT* to the concept definition.

### Match Latin-1 equivalent characters

use for Latin-1 languages that contain accented characters in their texts. When you select this classifier concept operation, you choose to match the Latin-1 equivalent characters as if they were unaccented. For example, match Pokémon as if it were Pokemon.

### Match XML character references

(classifier concepts only) match XML character references that appear in a document. For example, match `&amp;` for the ampersand character.

### Export CONCEPTS File with UTF-8 Display Names

(only enabled when you build a project using a UTF-8 language) create an additional concepts binary file where only UTF-8 display names appear. In other words, an additional file `language.concepts` is created. This file is `language.utf8.concepts`.

The `language.concepts` file contains the Latin-1 internal names, while the `language.utf8.concepts` file enables you to see the taxonomy in the UTF-8 language that appears in the **Taxonomy** tab. For example, if you created a taxonomy structure of concepts using Japanese, you might see:

Top/学校

instead of Top/School

### Allow Duplicate ID's



---

make it possible to assign duplicate identification numbers for two or more categories. You can enter these numbers in the **Data** tab for each affected concept.

### **Overlapping Concept Matches**

determine the behavior of SAS Content Categorization Studio when an input document contains terms that match more than one concept. For example, if your document contains the terms *Boston Market* and *Boston Scientific*, the following results are returned.

#### **All matches**

return all matched terms. In the example above, *Boston Market* and *Boston Scientific* are returned. In this case, you can specify a priority setting in the Data window to determine this match.

#### **Shortest**

return the shortest match. In the example above, *Boston Market* is returned.

#### **Longest**

(default setting) return the longest match. In the example above, only the term *Boston Scientific* is returned.

### **Default Classifier Matching**

(default setting: classifier concepts, only) select **Case Insensitive Matching** to locate all of the matching terms, regardless of case.

### **Optimize for**

make one of the following selections to optimize SAS Content Categorization Studio project building:

#### **Compile Speed**

(default setting) prioritize concept compilation.

#### **Matching Speed**

make concept matching the priority.

---

**Notes:** Unless you are developing large binary files, there is little performance difference between these settings. When you build a taxonomy that uses both categories and concepts, you can choose a setting independent of the selected operation in the **Category** tab.

---

---

### Default Relevancy Cutoff

(default setting: 0) set the **Default Relevancy Cutoff** for all of the concepts in the taxonomy. Unless the **Relevancy Cutoff** is specified in the **Data** tab for a specific concept, this cutoff applies to all of the concepts in the taxonomy.

### Relevancy Type

relevancy is used to determine the concept that is the best match for an input document:

#### Operator-Based

(default setting) use Boolean operators to determine the degree of relevancy for an input document.

#### Frequency-Based

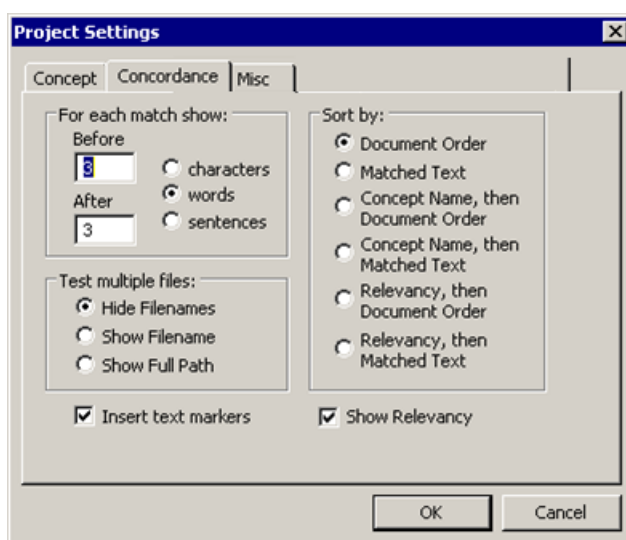
compute the number of matching terms in the text. Use this number to determine the degree of relevancy for an input document.

#### Zone-Based

weight matches in specific sections of a document to make them count more than matches in other areas.

## 2.9.4.B The Concordance View

Select the Project Settings - Concordance window to choose the display parameters for concept matches. Click **Concordance** in the **Document** tab and the **Document** tab becomes a concordance window. A concordance provides a list of the terms in the document that match the rule.



Use the following parameters:

**For each match show**

specify how many matching characters, words, or sentences are displayed in the concordance window:

**Before** (default: 3)

specify how many characters, words, or sentences to display before the match.

**After**

(default: 3) choose how many characters, words, or sentences to display after the match.

**characters**

(default setting) apply the numbers set in the **Before** and **After** fields to the letters in the alphabet, numbers, hyphens, and so on.

**words**

apply the numbers set in the **Before** and **After** fields to individual words.

---

## **sentences**

return the specified number of sentences, set in the **Before** and **After** fields.

## **Sort by**

classify the matching terms in the concordance view of the **Document** tab:

### **Document Order**

display the matches in the order in which the concepts occur in the document.

### **Matched Text**

sort the matches alphabetically.

### **Concept Name, then Document Order**

sort by concept name. Then sort by the order of appearance in the text.

### **Concept Name, then Matched Text**

sort the matches by concept name and then alphabetically.

### **Relevancy, then Document Order**

sort results according by relevancy to the concept and then in the order in which they appear in the input text.

### **Relevancy, then Matched Text**

display the most relevant results first and then by alphabetical ordering.

## **Test multiple files**

specify these operations when you use more than one testing file:

### **Hide Filenames**

(default setting) do not show the names of the files that match in the concordance view.

### **Show Filename**

display the test results, and to the right of this, the name of the file.

### **Show Full Path**

display the test results with the name of the file. The full path of that file appears to the right of the results.

---

### Insert text markers

display text markers in the concordance view of the **Document** tab when you test a single file against multiple concepts. The match text fields display the concept that is the best match for the matched term that is returned. One example of these tags is <CONCEPT1> . . . </CONCEPT1>.

### Show Relevancy

display the relevancy of each matched term.

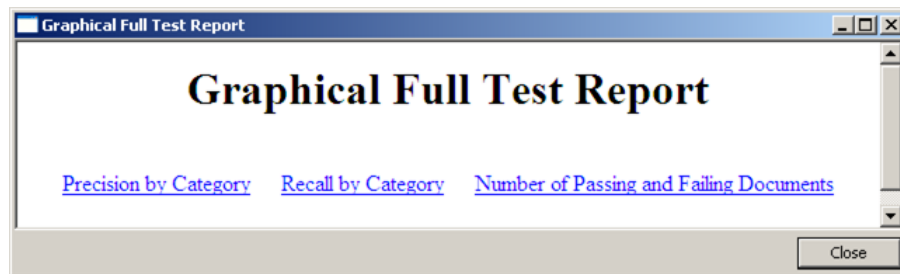
## 2.10 The Graphical Report Pages

### 2.10.1 The Graphical Full Test Report Page

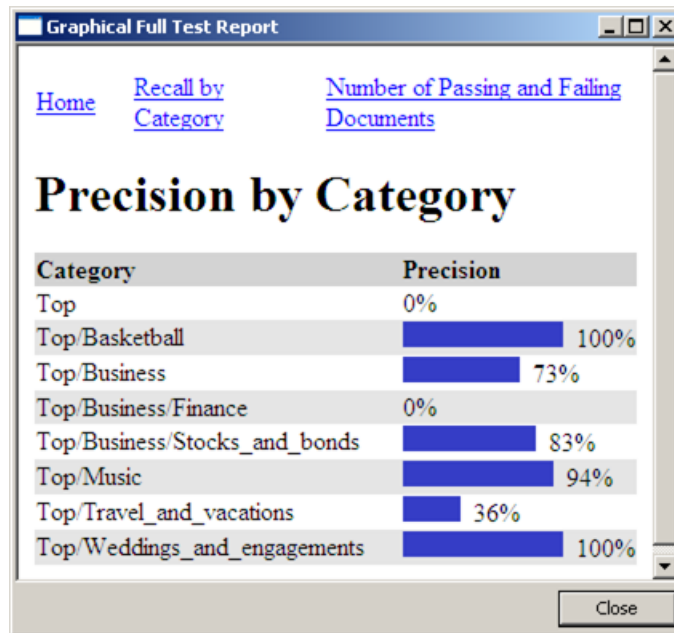
Use the graphical reports to see the statistics for category matches. You can see the precision, recall, and numbers of passing and failing documents in these reports.

To access and use the Graphical Full Test Report pages, complete these steps:

1. Select **Testing --> Graphical Full Test Report**. The Graphical Full Test Report page appears.

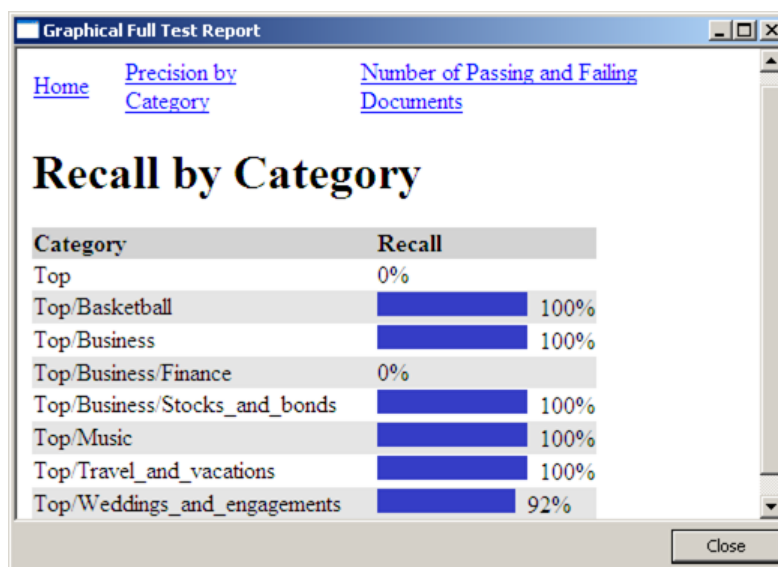


2. Click **Precision by Category**. The Precision by Category page appears.

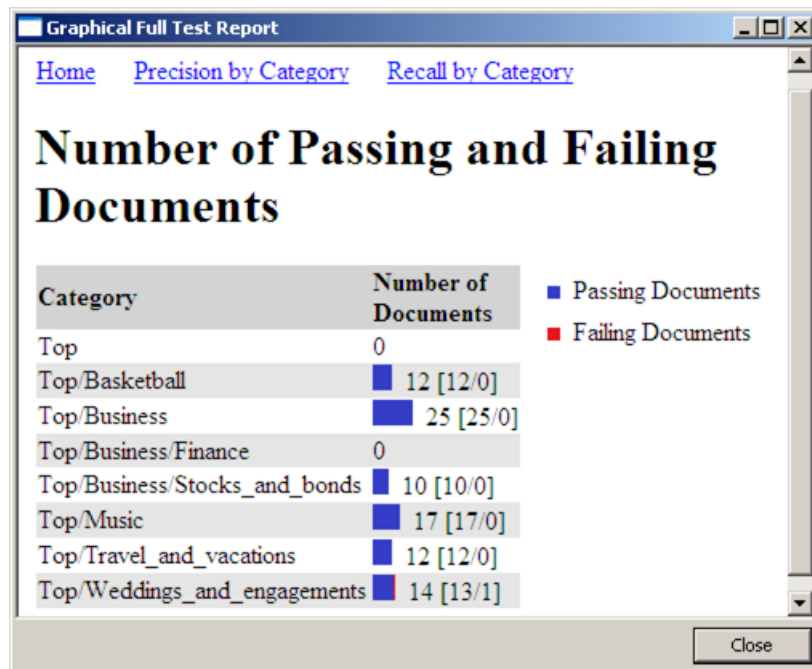


3. (Optional) Click the **Category** heading to display the categories alphabetically starting from the letter Z down or from the letter A down.
4. (Optional) Click the **Precision** heading to display the results starting from the 0%, or from 100%, down.

- 
5. Click **Recall by Category**. The Recall by Category page appears.



6. (Optional) Click the **Category** heading to display the categories alphabetically starting from the letter Z down or from the letter A down.
7. (Optional) Click the **Recall** heading to display the results starting from the 0%, or from 100%, down.



8. See the number of **Passing Documents** in blue and the number of **Failing Documents** in red.
9. (Optional) Click the **Category** heading to display the categories alphabetically starting from the letter Z down or from the letter A down.
10. (Optional) Click the **Number of Documents** heading to display the results starting from the 0%, or from 100%, down.
11. Click **Close**.
12. (Optional) Click **Testing --> Show Last Full Graphical Testing Report** after you close this report. This operation restores the last report.



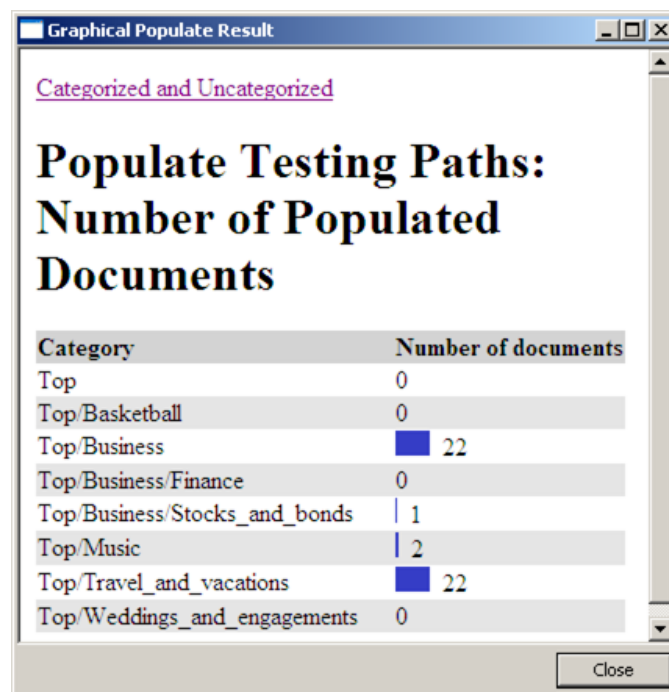
---

## 2.10.2 The Graphical Populate Result Window

Use the Graphical Populate Result window to see the graphed results for the categorized and uncategorized documents. This window is used after the **Testing --> Populate Testing Paths** operation.

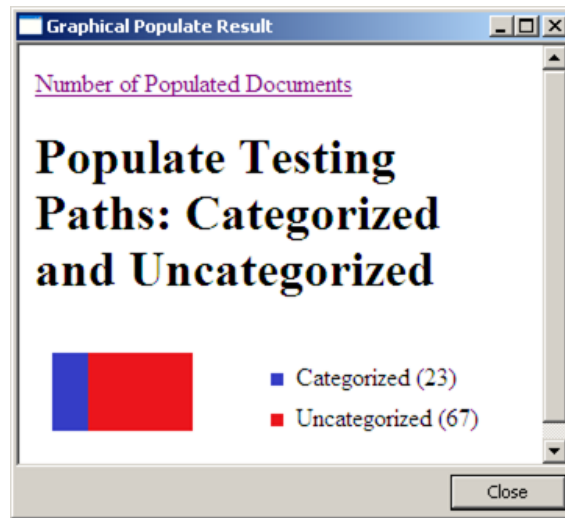
To access and use the Graphical Full Test Report pages, complete these steps:

1. Select **Testing --> Show Graphical Populate Results**.



2. Click **Category** to see the categories in alphabetical order.
3. Click **Number of documents** to order the documents by number of matches.

- 
4. Click **Categorized and Uncategorized** to see a bar chart representing the numbers of matched input documents.



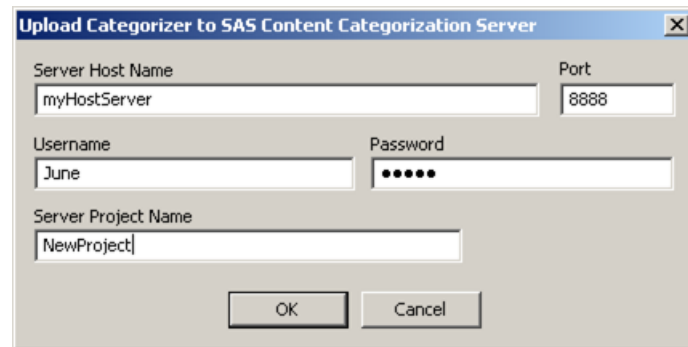
5. Click **Close** to close this window and to return to the user interface.

## 2.11 The Uploading the Categorizer, or Concepts, to SAS Content Categorization Server Window

Specify the server information that is necessary to upload the .mco file in the Upload Categorizer to the SAS Content Categorization Server window. The .mco file is created by SAS Content Categorization Studio.

---

*Display 2-19 Upload Categorizer to SAS Content Categorization Server Window*



Upload Categorizer to SAS Content Categorization Server

Server Host Name: myHostServer Port: 8888

Username: June Password: •••••

Server Project Name: NewProject

OK Cancel

Specify your upload settings in the fields that appear in the Upload Categorizer to the SAS Content Categorization Server window:

**Server Host Name**

name of the SAS Content Categorization Server host.

---

**Port**

port number for the SAS Content Categorization Server.

**Username**

your name as specified in the SAS Content Categorization Server configuration file.

**Password**

your password, as specified by the server administrator in the SAS Content Categorization Server configuration file.

**Server Project Name**

project name, as it appears in the SAS Content Categorization Server configuration file. For more information, see the *SAS Content Categorization Server: User's Guide*.

---

**Notes:** For more information about the SAS Content Categorization Server configuration file, see the *SAS Content Categorization Server: User's Guide*.  
The format for specifying a username and password in the configuration files is:  
`user=<username>:<password>`  
You can have as many `user=` directives as you like, for example:  
`user=andersen:csandersen`  
`user=boyle:csboyle`  
`user=cabot:cscabot`

---

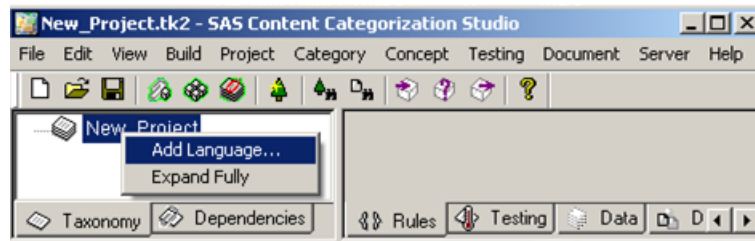
## 2.12 The Miscellaneous Windows

### 2.12.1 The Select a Language Window

Use the Select a Language window to choose a language that you purchased for the entire taxonomy or for a branch of your project.


To access and use the Select a Language window, complete these steps:

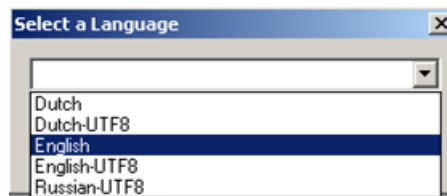
1. Right-click on the project node in the **Taxonomy** tab and select **Add Language** from the drop-down menu that appears.



The Select a Language window appears.



2. Click  to the right of the blank field and select the language from those languages that you purchased.



Languages followed by -UTF8 are in UTF-8 encoding. These languages include English, Chinese, Japanese, Korean, and Russian (Cyrillic characters), and so on. If the language is not followed by -UTF8, Latin-1 is used as the character set encoding.

---

**Notes:** When UTF-8 encoding is specified, test only documents that are UTF-8 encoded.  
If you specify a language with UTF-8 encoding, make

---

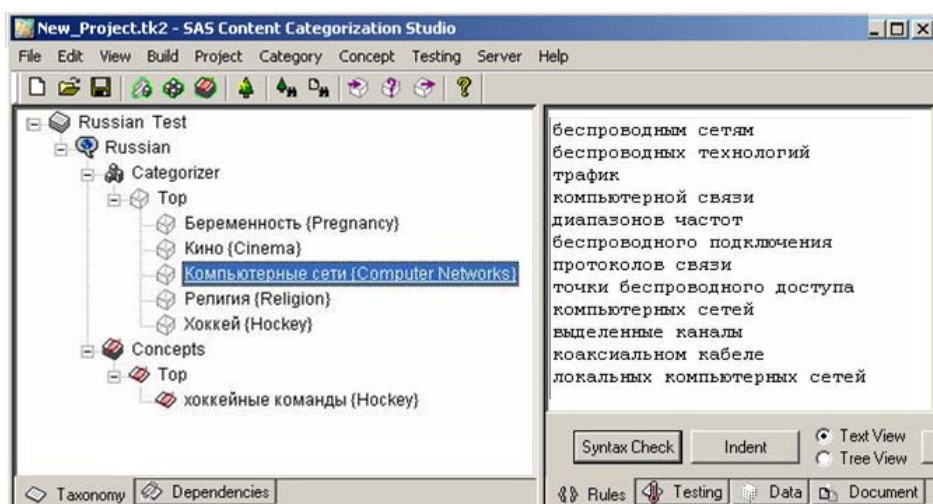
sure that your computer has the proper language fonts installed.

---

3. Click **OK**. The selected language is added to your project. See the English example below.



See the example of a Russian taxonomy below.



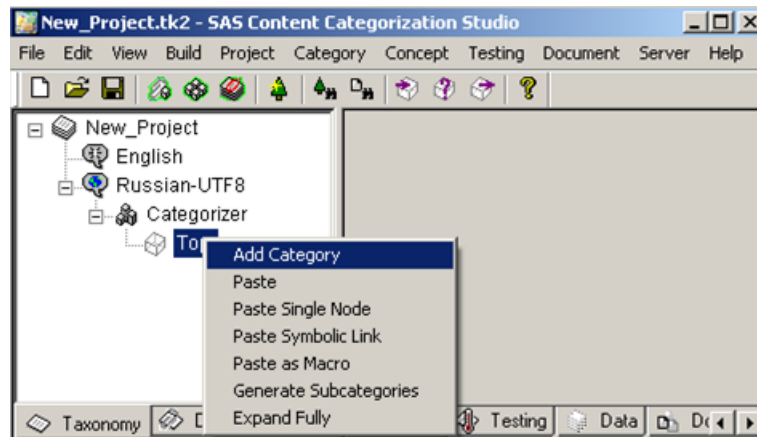
---

## 2.12.2 The Enter Names Window for UTF-8 Languages

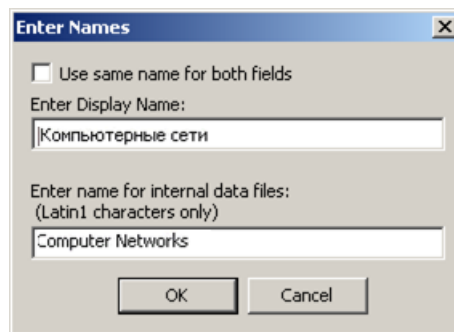
Categories and concepts that use UTF-8 encoding require two names in the Enter Names window. Both of these names appear in the **Taxonomy** tab.

To access and use the Enter Names window, complete these steps:

1. Right-click on the **Top** node in the **Taxonomy** tab and select **Add Category** from the menu that appears.



The Enter Names window appears.



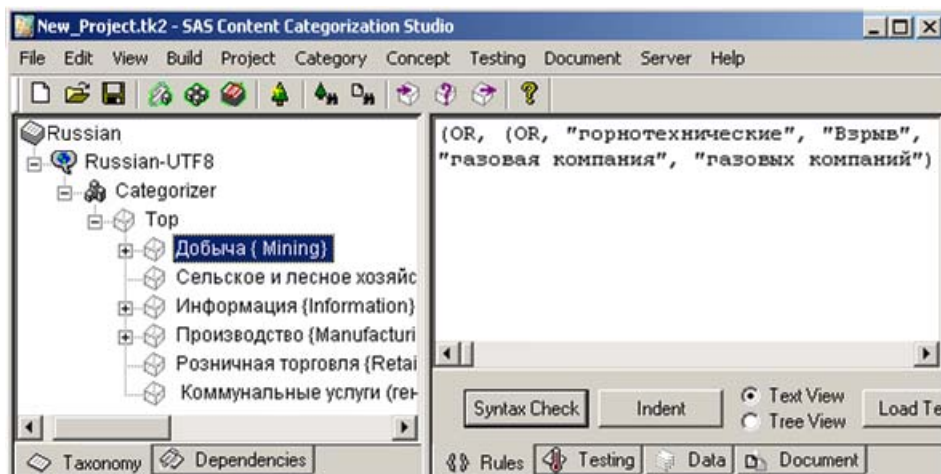
2. (Optional) Select **Use same name for both fields** and the **Enter Display Name** field is dimmed.

---

**Notes:** If you decide to enter only one name for each node in the taxonomy, you should select **Hide Display Names for UTF-8 Languages**. For more information, see Section 2.8 *The Options Window* on page 47. In this case, the **Enter Display Name** field in the Enter Names window is automatically dimmed. This is true for each category and concept that is added, or renamed, in your project.

---

3. Type the name of the category or concept into the **Enter Display Name** field using UTF-8 language characters.
4. Type the English name for your category or concept into the **Enter name for internal data files (Latin-1 characters only)** field.
5. Click **OK**. The new category or concept name appears in the **Taxonomy** tab.





---

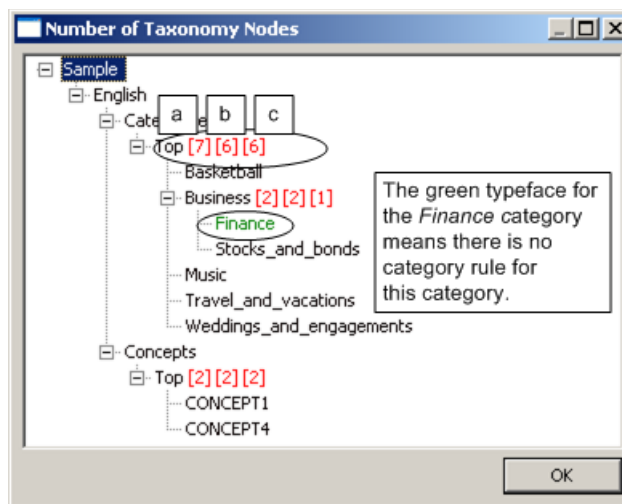
### 2.12.3 The Number of Taxonomy Nodes Window

Use the Number of Taxonomy Nodes window to see the following information about the taxonomy nodes:

- number of nodes
- number of subnodes
- nodes without a category rule or concept definition

To access and use the Number of Taxonomy Nodes window, complete the following steps:

1. Select **View --> Number of Taxonomy Nodes**. The Number of Taxonomy Nodes window appears.



2. Use the Number of Taxonomy Nodes window to obtain the following types of counts (the list below correlates to the numbers in the figure above):
  - a. The number of taxonomy nodes represents all of the subnodes for the selected node in the **Taxonomy** tab. In the example above, 7 appears to the right of the **Top** node.
  - b. The count of the children of the selected node that do not have subnodes is the second number that is displayed. In the example above, this number is 6. **Business** has two subnodes and there are no children for these two child nodes.
  - c. The number of subnodes that have a rule or definition is the last count. In the example above, there are six subnodes. The **Finance** category appears in green highlighting because it does not have a rule.
3. Click **OK**.

## 2.12.4 The Tree Find Window

Use the Tree Find window to locate a category or a concept in a large taxonomy.

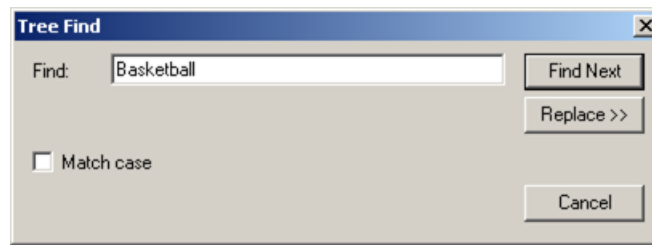
To find a category, complete these steps:

1. Select the **Tree Find** icon on the standard toolbar.



---

The Tree Find window appears.



2. Type the name of the category or concept that you want to locate into the **Find** field.
3. (Optional) Select the **Match case** box to locate the specified name in the same case.
4. Select **Find Next** to locate a match.
5. (Optional) Select **Replace** to access the Tree Replace window. For more information, see Section 2.12.5 *The Tree Replace Window* below.
6. Click **Cancel**.

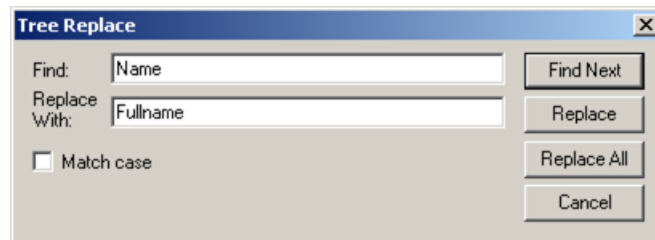
---

## 2.12.5 The Tree Replace Window

Use the Tree Replace window to substitute a new name for the name that appears on one or more nodes in the **Taxonomy** tab.

To access and use this window, complete these steps:

1. Select **Edit --> Tree Replace** and the Tree Replace window appears.



2. Type the text that you want to locate into the **Find** field.
3. Enter the text that you want to substitute for the located term into the **Replace With** field.
4. If you want to replace all of the original terms with the specified text, click **Replace All**.

---

**Note:** Use the **Replace All** button with care. This operation cannot be undone.

---

For more information, see Section 2.12.4 *The Tree Find Window* on page 76.

## 2.12.6 The Text Find and Replace Windows

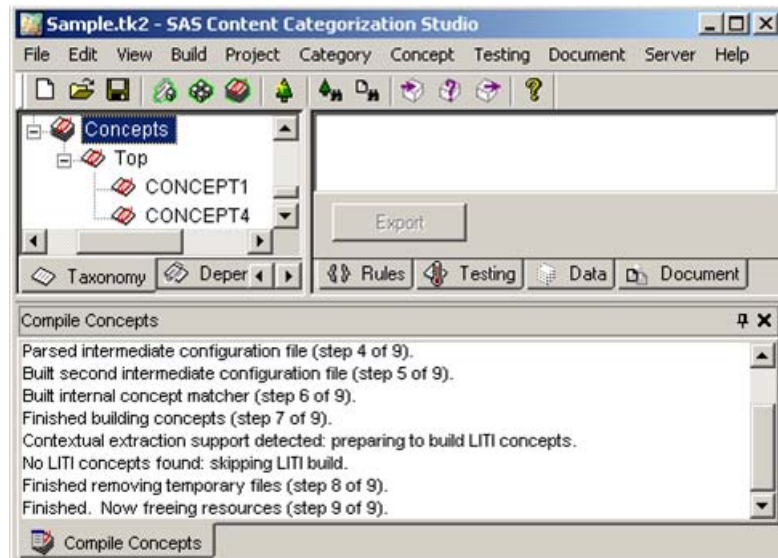
Use the Text Find and the Text Replace windows like you use the Tree Find and Replace windows, or these operations in other applications. In SAS Content Categorization Studio, these operations work in the **Definition**, **Testing**, and **Document** tabs.

---

## 2.12.7 The Compile Concepts and Build Categorizer Tabs

The **Compile Concepts** tab appears at the bottom of the SAS Content Categorization Studio interface when you select **Build --> Compile Concepts**. This tab provides status information about the build process.

*Display 2-20 Compile Concepts Tab*



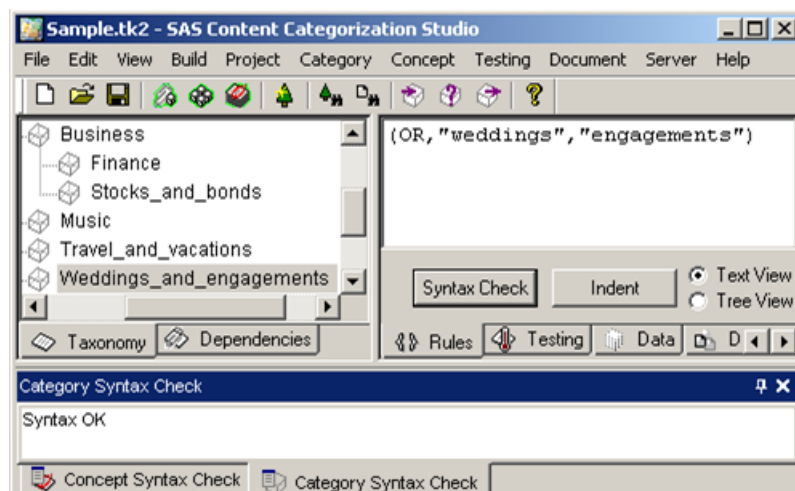
Select **Build --> Build Rulebased Categorizer** to access the Build Rulebased Categorizer window. Choose **Build Statistical Categorizer** to access the Build Statistical Categorizer window.

---

## 2.12.8 The Syntax Check Window

Use the syntax check operation to check the grammar of your definition or rule. Click **Syntax Check** and the **Category Syntax Check**, or the **Concepts Syntax Check**, tab appears at the bottom of the user interface. This tab displays the results of the grammar check for the selected category or concept.

*Display 2-21 Syntax Check Tab*



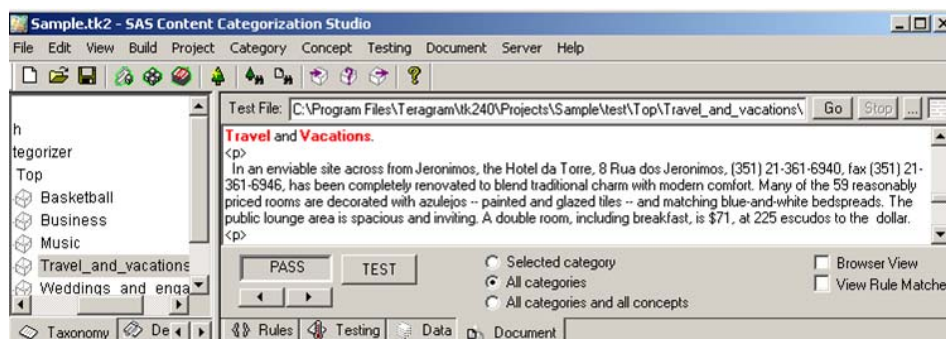
---

## 2.12.9 The Best Matches Window

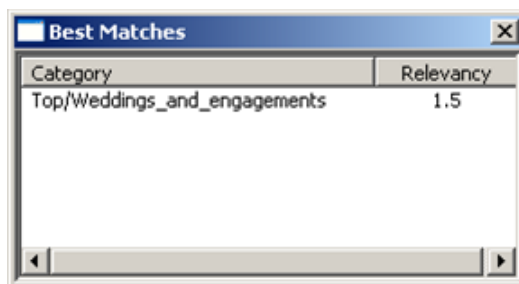
Use the Best Matches window to see a list of the highest ranking categories and concepts for your document. This window appears when you select **Show best matches when testing all** in the Options window and choose to test more than one node.

To access the Best Matches window, complete these steps:

1. Access a testing document in the **Document** tab.



2. Select one of the following radio buttons **All categories** (All concepts for a concept) or **All categories and all concepts**.
3. Click **TEST**. The Best Matches window appears. See the example provided below.



There are two headings in the Best Matches for categories window. Under **Category**, see the path from the **Top** node to the tested category. The relevancy score for the tested document is listed under **Relevancy**. The category with

---

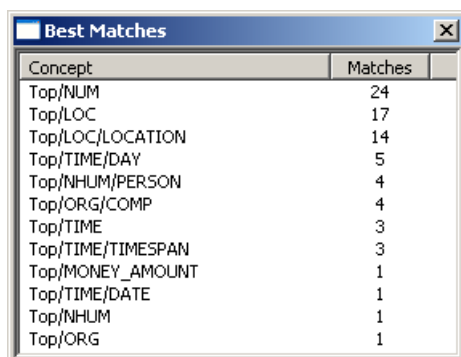
the highest relevancy score is listed first and the least relevant category is listed last.

---

**Note:** To set, or change, your relevancy settings use the Project Settings tabs to affect all of the categories project-wide. Alternatively, use the **Data** tab to change the relevancy settings for one node only.

---

See the following example of a Best Matches window that appears when you test concepts.



Concept	Matches
Top/NUM	24
Top/LOC	17
Top/LOC/LOCATION	14
Top/TIME/DAY	5
Top/NHUM/PERSON	4
Top/ORG/COMP	4
Top/TIME	3
Top/TIME/TIMESPAN	3
Top/MONEY_AMOUNT	1
Top/TIME/DATE	1
Top/NHUM	1
Top/ORG	1



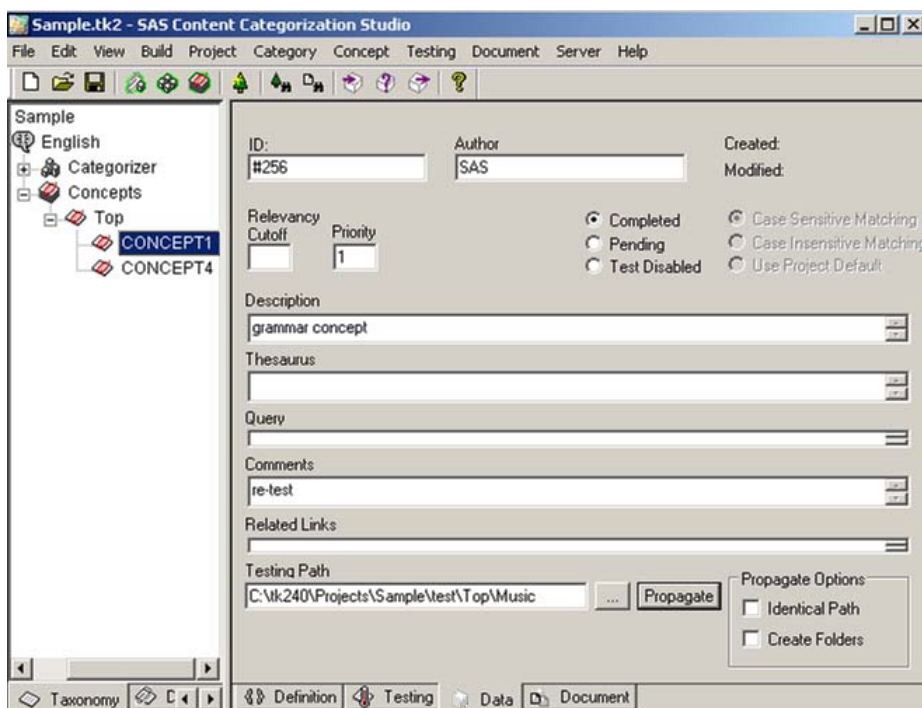
---

## 2.12.10 The Concept Priorities Window

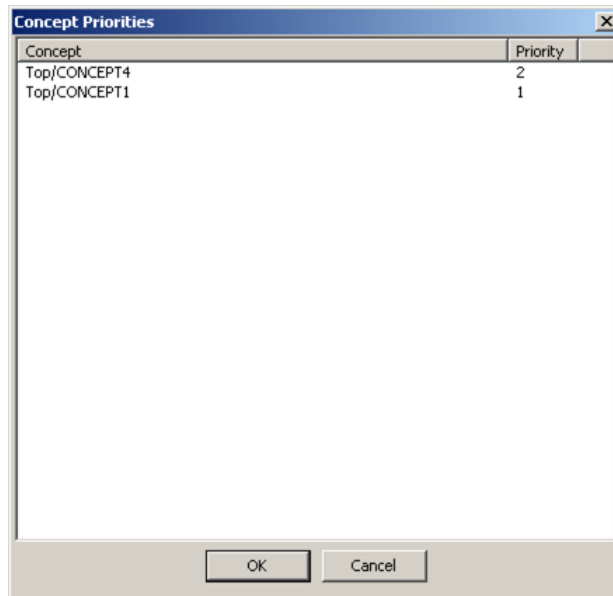
The Concepts Priorities window displays the priority settings for concepts. Priority determines the matching concept when one input document matches two or more concepts and no other determiner makes one concept a better match than another. The Concepts Priorities window displays the priorities setting from each Data window in the concepts taxonomy and ranks these concepts.

To access the Concept Priorities window, use these steps:

1. Specify a priority setting in the Data window for each concept that you want to rank. For example, type 2 into the **Priority** field for one concept and 1 into the **Priority** field for another concept.



- 
2. Select **Concept --> Priorities**. The Concept Priorities window appears.



3. See a ranked list of concepts according to the priorities that you specified.
4. (Optional) Click **Concept** to list the concepts from A - Z.
5. (Optional) Click **Priority** to prioritize the concepts from highest to lowest.
6. Click **OK**.

---

**Hint:** The Concept Priorities window does not list concept matches. This window displays only the concepts that have a priority setting with their priority specifications.

---

---

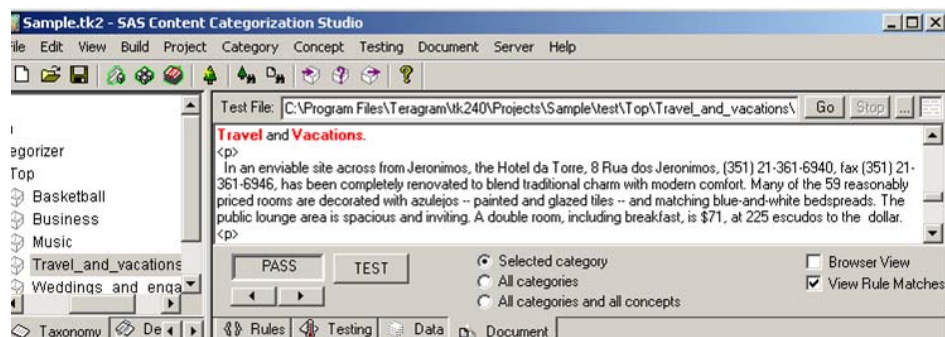
## 2.12.11 The Rule Matches Window

Use the Rule Matches window to see a list of the matching Boolean rule terms that define only the tested category. These terms are highlighted in red in the taxonomy format.

This window automatically appears when you select **View Rule Matches** in the **Document** tab and test a Boolean category rule.

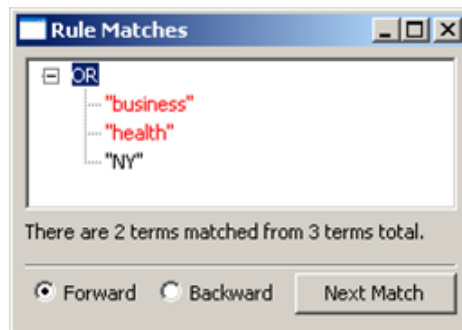
To use the Rule Matches window, complete these steps:

1. Access a testing document in the **Document** tab.



2. Select **Selected category**.
3. (Optional) Select **View Rule Matches**, unless you made this selection before you accessed the testing document in the **Document** tab.
4. Click **TEST**.

- 
5. The Rule Matches window appears. The taxonomy of matched terms in the selected Boolean rule is displayed.



Use the components of the Rule Matches window as explained below:

#### Taxonomy window

This window displays the taxonomy of matched terms in the Boolean rule. The terms are highlighted in red.

#### Explanatory text

Below the taxonomy window see an explanation of the matches. For example, see There are 2 terms matched from 3 terms total.

#### Forward radio button

Select this operation and click **Next Match** to see the next matched term in this taxonomy.

#### Backward button

Select this operation and click **Next Match** to see the previously viewed term in the taxonomy.

---

**Hint:** Select either the **All categories** or **All categories and concepts** radio button and click **TEST**. The Best Matches window appears instead of the Rule Matches window.

---

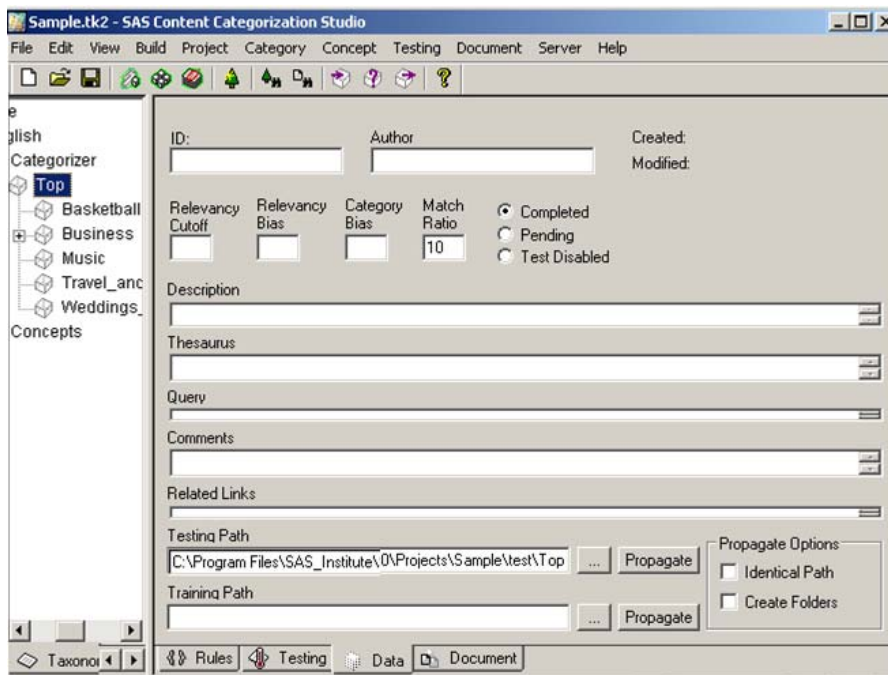
---

## 2.12.12 The Full Test Report Window

The Full Test Report window is used to see a range of testing results for the category taxonomy.

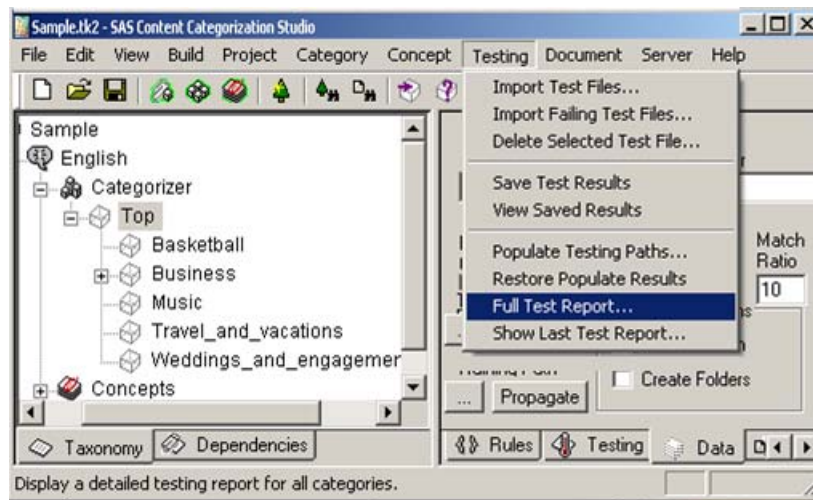
To use the Full Test Report window, complete the following steps:

1. Set the **Testing Path** for each category in the taxonomy.



---

2. Select **Testing --> Full Test Report**.



The **Category Test Report** window appears.

Path	All Docs	In-Cat	Total	In-Cat %	Neg	N-Tot	Neg %	Prec %	Popul...	Pop Rel
Top	0	0	0	0	0	0	0	0	0	0
Top/Business	34	25	25	100	0	0	0	73	0	0
Top/Business/Stocks_and_bonds	12	10	10	100	0	0	0	83	0	0
Top/Business/Finance	0	0	0	0	0	0	0	0	0	0
Top/Music	18	17	17	100	0	0	0	94	0	0
Top/Basketball	12	12	12	100	0	0	0	100	0	0
Top/Weddings_and_engagements	13	13	13	100	0	0	0	100	0	0
Top/Travel_and_vacations	33	12	12	100	0	0	0	36	0	0

OK View as Text

---

Use the following headings in the Category Test Report window to analyze this report:

**Path**

route, from the `Top` node to the specified category.

**All Docs**

total number of tested documents that are a match for this category. These testing documents are both in-category and out-of-category.

**In-Cat**

the number of test documents that are assigned to this category that are a match for this category.

**Total**

total number of test documents that are assigned to this category.

**In-Cat%**

percentage of in-category test documents that match this category.

**Neg**

number of in-category texts that are located in a folder, such as the Fail folder.

**N-Tot**

total number of failing test documents for this category in the Fail folder.

**Neg %**

percentage of all in-category failing test documents that match the selected category.

**Prec %**

precision percentage measures the in-category matches as a percentage of the out-of-category matches for this category.

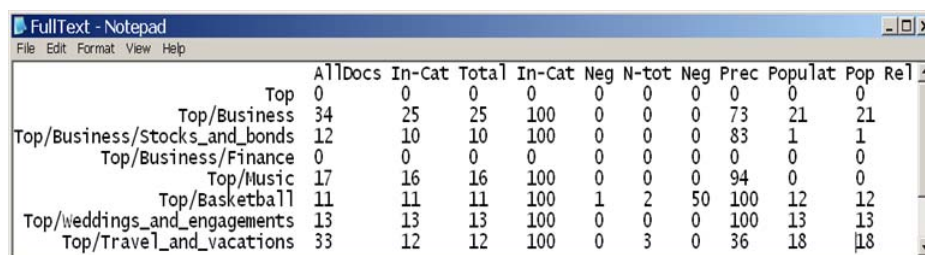
**Populate**

number of files that were assigned to this category the last time a populate testing paths operation was performed.

**Pop Rel**

number of files assigned to this category that are above the relevancy threshold. This number reflects the results of the latest Populate Testing Paths operation.

3. (Optional) Click **View as Text** to see the results in a *Notepad* window. To close the *Notepad* window, click **X** in the upper right-hand corner of the FullText window.

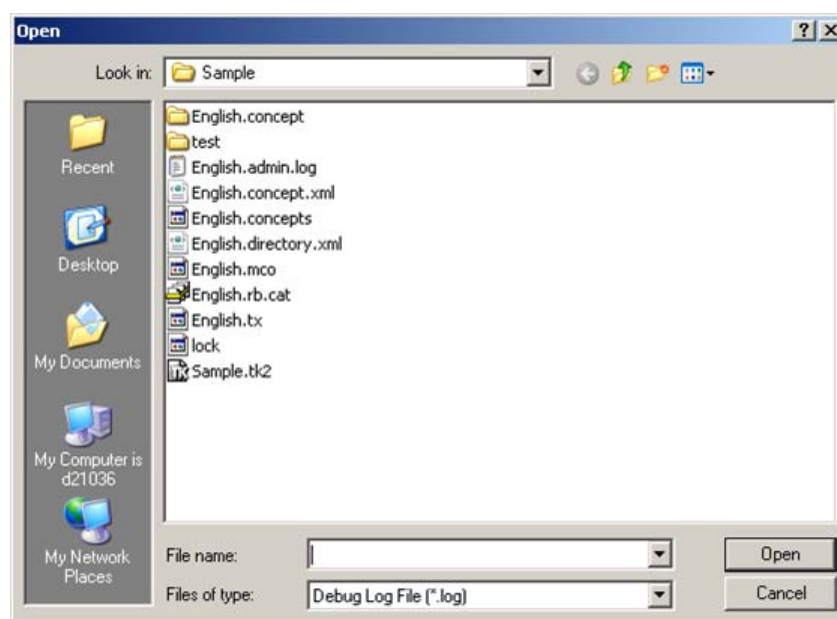


	AllDocs	In-Cat	Total	In-Cat	Neg	N-tot	Neg	Prec	Populat	Pop Rel
Top	0	0	0	0	0	0	0	0	0	0
Top/Business	34	25	25	100	0	0	0	73	21	21
Top/Business/Stocks_and_bonds	12	10	10	100	0	0	0	83	1	1
Top/Business/Finance	0	0	0	0	0	0	0	0	0	0
Top/Music	17	16	16	100	0	0	0	94	0	0
Top/Basketball	11	11	11	100	1	2	50	100	12	12
Top/weddings_and_engagements	13	13	13	100	0	0	0	100	13	13
Top/Travel_and_vacations	33	12	12	100	0	3	0	36	18	18

4. Click **OK**.

## 2.12.13 The Open Window

Use the Open window to locate a file, or a folder.





---

## 2.12.14 Examples of the Status Windows

### 2.12.14.A The Upload Complete Window

After you use the Upload categorizer to the SAS Content Categorization Server window to upload your categories, a SAS Content Categorization Studio confirmation window appears.

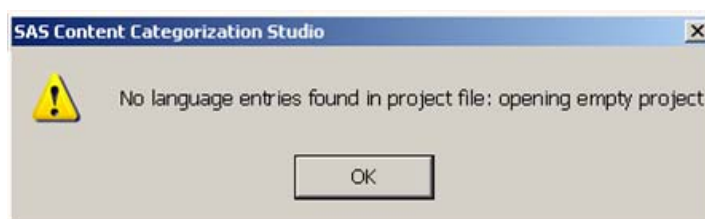
*Display 2-22 SAS Content Categorization Studio Confirmation Window*



### 2.12.14.B The No Language Entries Window

If you name and save a new project before you add a language, a SAS Content Categorization Studio status window appears the next time you access this project.

*Display 2-23 SAS Content Categorization Studio Status Window*



Click **OK**. Add a language in the Select a Language window. For more information, see Section 2.12.1 *The Select a Language Window* on page 70.

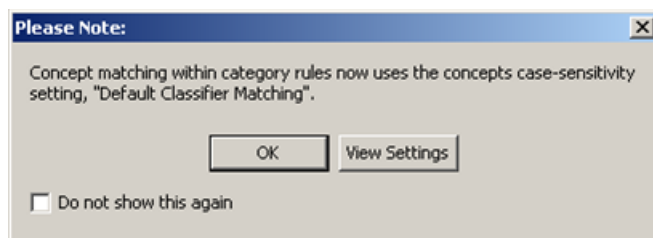
## 2.12.15 The Please Note Window for Old Projects

If you access an old project that has categories with dependencies on concepts, the Please Note window appears. This window advises you about the use of

---

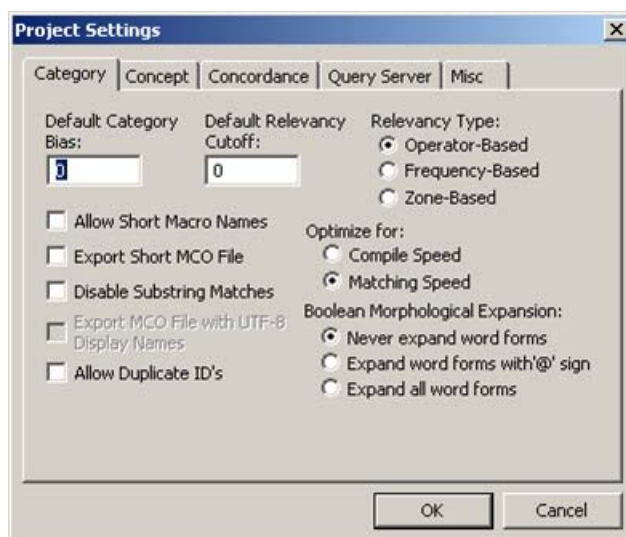
case sensitivity. This window also enables you to check this setting in the Project Settings window before you access your project.

*Display 2-24 Please Note Window*

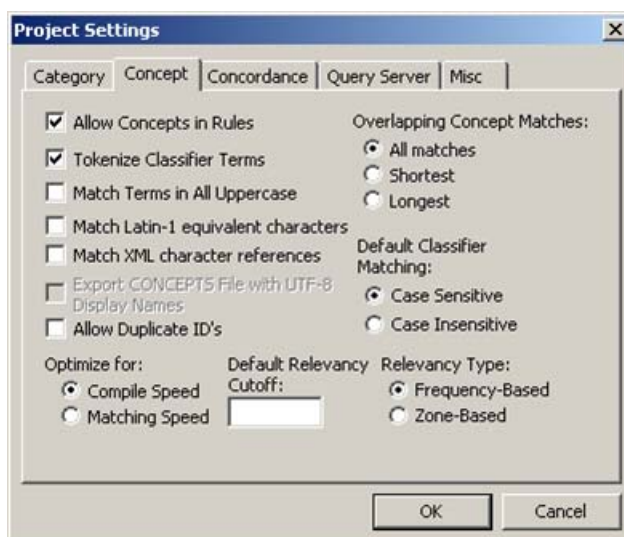


To use the Please Note window, complete these steps:

1. (Optional) If you click **Do not show this again** in the Please Note window, this window is not displayed the next time you access this project.
2. Click **OK**.
3. (Optional) Click **View Settings** to access the Project Settings window.



4. Click the **Concept** tab to see the **Default Classifier Matching** setting.



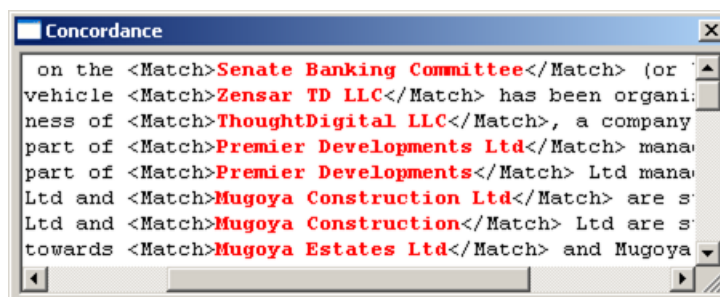
5. Click **OK**.

## 2.12.16 The Concordance Windows

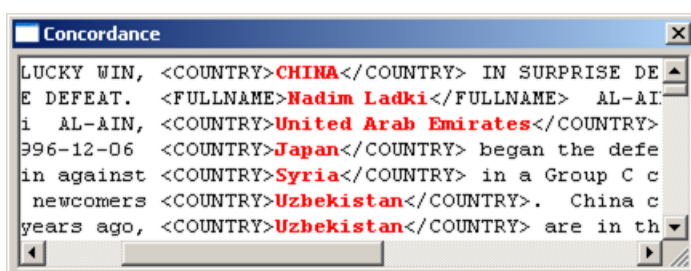
When you select either **Concordance for Selected Concepts** or **Concordance for All Concepts**, the Concordance window appears displaying the selected matches.

See the following examples:

*Display 2-25 Concordance for Selected Concept*



Display 2-26 Concordance for All Concepts



## 2.13 The Drop-down Taxonomy Node Operations

### 2.13.1 The Project Name Node Operations

Right-click on the first node that appears after you name your project. This is the name of the project.

Display 2-27 Add Language and Expand Fully Operations



#### Add Language

specify a language for this branch of your taxonomy. The Select a Language window appears with a drop-down list of the languages that you purchased. For more information, see Section 2.12.1 *The Select a Language Window* on page 70.

#### Expand Fully

see all of the nodes in this taxonomy.

---

## 2.13.2 The Language Node Operations

Right-click on the language node in your taxonomy in order to access the drop-down operations.

*Display 2-28 Language Node Drop-down Operations*



Use the following operations to change your taxonomy structure:

### **Delete Language**

remove the language node for this taxonomy.

---

**Warning:** When you choose to use the **Delete Language**, all of the child nodes that follow the language node are deleted with the selected node.

---

### **Enable Categorizer**

add the `Categorizer` node that is used to add category nodes.

### **Import Categorizer from XML**

jump-start the development of a new taxonomy when you import an existing taxonomy from another project in the form of a `.xml` file

### **Create Categorizer from XML**

develop a categorizer from the imported XML file.

### **Create Categorizer from Directories**

develop a categorizer from the subfolders of the imported file.

### **Enable Concepts**

add the `Concepts` node that is used to add concepts.

---

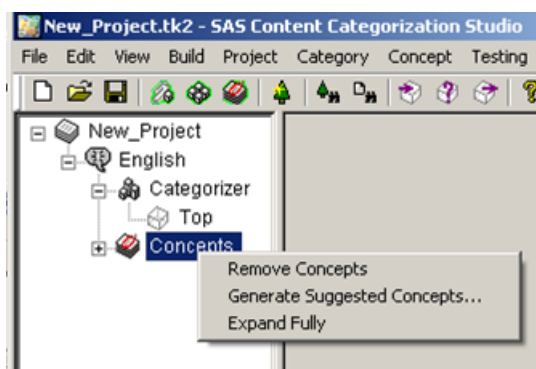
## Expand Fully

access the taxonomy to see all of its nodes.

### 2.13.3 The Categorizer or Concepts Node Operations

Right-click on the `Categorizer` or the `Concepts` node. This node specifies the name of a project.

*Display 2-29 Concepts Node Operations*



Select from the following operations. Substitute the word *concept* for *category* where necessary:

#### Remove Concepts

delete this node from the taxonomy when you choose this operation.

---

**Warning:** When you select this operation, all of the child nodes below the language node are deleted with the `Categorizer` or `Concepts` node.

---

#### Generate Suggested Concepts

(this operation is not available for categories) generate a list of classifiers for the concepts in your taxonomy from the matching classifier concepts in another `.tk2` file. Import these classifiers into the definitions for this project. For more information, see Section 18.6 *Generating Suggested Concepts* on page 461.

---

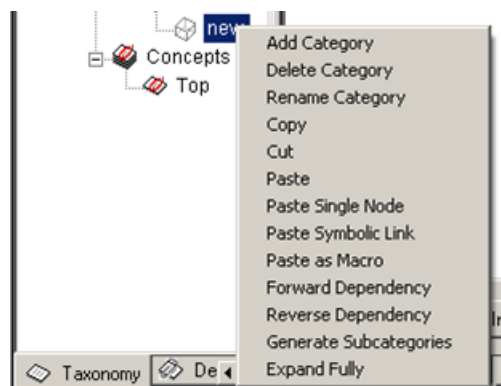
### Expand Fully

click to display all of the nodes in the selected branch of the taxonomy.

## 2.13.4 The Individual Category or Concept Node Operations

Right-click on a category or concept node and a list of operations appears in the drop-down menu:

*Display 2-30 Individual Category Operations*



Select from the following operations for categories (and concepts). The **Cut**, **Copy**, and **Paste** operations are self-explanatory:

#### **Add Category**

add a child category to the selected parent node.

#### **Delete Category**

remove the selected category node.

#### **Rename Category**

change the name of the category.

#### **Paste Single Node**

paste one copied node as a child of the selected category.

#### **Paste Symbolic Link**

---

(this operation is category-specific) copy the source category and use this selection to create a target category that is only a pointer to its source. For more information, see Section 8.8.3 *Define a Symbolic Link* on page 228.

**Paste as Macro**

create a dependency. For more information, see Section 11.12.2 *Paste a Macro* on page 337.

**Forward Dependency**

access the **Dependencies** tab. If there are no dependencies, a SAS Content Categorization Studio window appears with this statement.

**Reverse Dependency**

access the **Dependencies** tab. If there are no dependencies, a SAS Content Categorization Studio window appears with this statement.

**Generate Subcategories**

(available only from child nodes) select this operation after you set a **Training Path** in the **Data** tab and child categories are automatically generated.

**Expand Fully**

access the selected branch of your taxonomy.



---

# Chapter: 2

## Creating Projects

---

- *Overview of Creating Projects*
- *Start SAS Content Categorization Studio*
- *Creating a New Project*
- *Saving the Project*
- *Access an Existing Project*
- *Set Installation-Specific Operations*
- *Specifying Project Settings*
- *Navigating through Categories and Concepts*
- *Export a UTF-8 Binary File*
- *Upload the Categorizer or Concepts to SAS Content Categorization Server*

### 2.1 Overview of Creating Projects

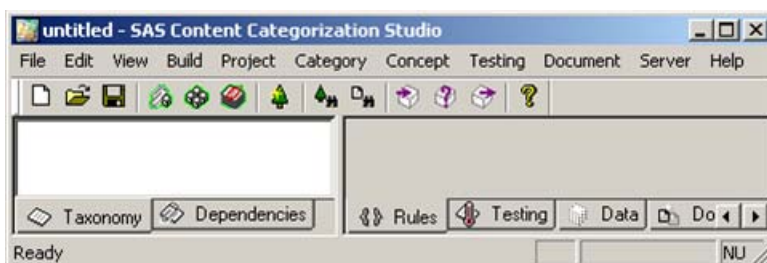
Build a categorizer, a concepts extractor, or both in the framework of a project. The taxonomy is the tree structure that organizes the category and concept nodes alphabetically. You write rules to define categories and concepts and test them using the **Testing** and **Document** tabs to ensure that these rules perform as expected. The rules can be exported as `.mco` and `.concepts` files to be used by SAS Content Categorization Server in real time.

---

## 2.2 Start SAS Content Categorization Studio

To start SAS Content Categorization Studio, complete these steps:

1. Select **Start --> Programs --> SAS Content Categorization Studio** and the untitled user interface appears.



2. Choose between using Section 2.3 *Creating a New Project* below, or Section 2.5 *Access an Existing Project* on page 112.

---

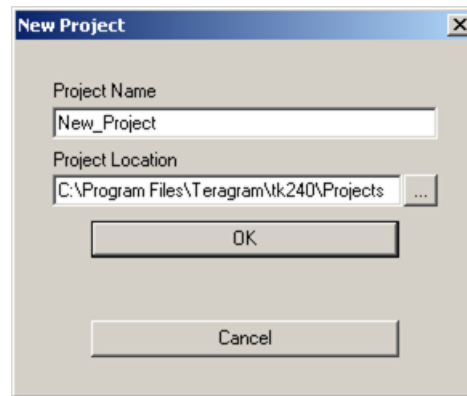
## 2.3 Creating a New Project


### 2.3.1 Create a New Project

Use this section to develop a new project the first time you use SAS Content Categorization Studio. You can also use this section any time you need a new project.

To create a new project, complete these steps:

1. Select **File --> New Project** and the New Project window appears.



2. Type the name of the new project into the **Project Name** field.
3. (Optional) Click  to locate a file and load this file into the **Project Location** field. The default location for a 32-bit machine running an English version of Windows is:  
`c:\ProgramFiles\Teragram\tk240\projects.`  
For a 64-bit machine or for other languages, the default folder `ProgramFiles` might be different. For example, the folder might be entitled `ProgramFilescx867` or `Archivosdeprograma`.
4. Click **OK**.

- 
5. The newly named project node appears in the **Taxonomy** tab. For example, see the `New_Project` node below



---


**Hints:** After you create a new project, set your project-wide settings. You can also choose to set your project-wide settings at a later stage in project development. For more information, see Section 2.7 *Specifying Project Settings* on page 130.

Remember to save your project frequently throughout development. For more information, see Section 2.4 *Saving the Project* on page 108.

---

6. Right-click on the project icon and select **Add Language** from the drop-down list that appears.

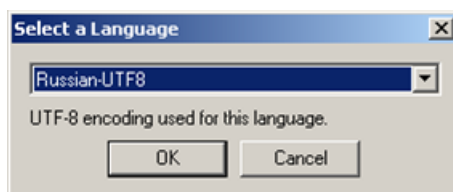
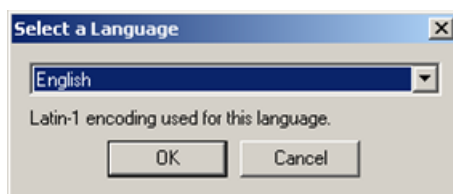


7. The Select a Language window appears. Click  to the right of the blank field to select a language and the encoding for this language.

Languages that are represented in both Latin-1 and UTF-8, such as western European languages, have two entries in the drop-down list. All other languages use the multibyte character set encoding because

---

UTF-8 can represent every character encoding in the Unicode character set.



---

**Note:** If UTF-8 encoding is used, make sure that all of the testing and input documents are UTF-8 encoded. Also ensure that your computer has the proper language fonts installed.

---

8. Click **OK**. The **Taxonomy** tab shows the new project node and the language node.



9. Enable categorization and concepts. For more information, see Section 2.3.2 *Enable Categorization and Concepts Extraction* on page 104. Alternatively, you can decide to import a taxonomy from an XML file. When you choose this second operation, the categorization taxonomy is

---

built for you. For more information, see Section 2.3.3 *Import a Project from an XML File* on page 105.

## 2.3.2 Enable Categorization and Concepts Extraction

Whether you choose to enable both categorization and concepts, or one of these nodes only, complete these steps:

1. Right-click the language icon that appears in the **Taxonomy** tab. For example, right-click on *English*.
2. Select **Enable Categorizer** or **Enable Concepts** from the drop-down menu that appears.



3. (Optional) If you choose to enable both categories and concepts, repeat Step 2 above and select the other operation.

---

When you choose to enable both the categorizer and concepts extraction, the **Taxonomy** tab displays these nodes.



4. Select **File --> Save**. For more information, see Section 2.4 *Saving the Project* on page 108.

---

**Note:** If you have not specified your project settings, you should do so now. For more information, see Section 2.7 *Specifying Project Settings* on page 130.

---

After you create your project, define and build your taxonomy by adding categories or concepts to the project. For more information, see Chapter 4 and Chapter 16.

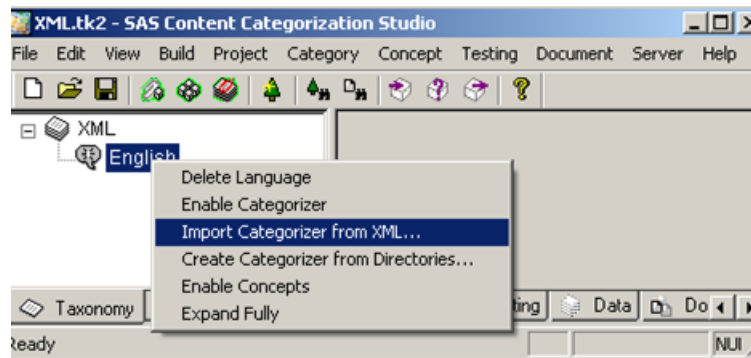
### 2.3.3 Import a Project from an XML File

After you develop a project, one or more XML files are created that store the category taxonomy, rules, and other information defined in the project. This file can be imported into another project to jump-start its development. You can import `<language>.directory.xml`, where `language` is the same as the language node for the selected taxonomy branch. If you build a project using more than one language, you can import multiple `.xml` files.

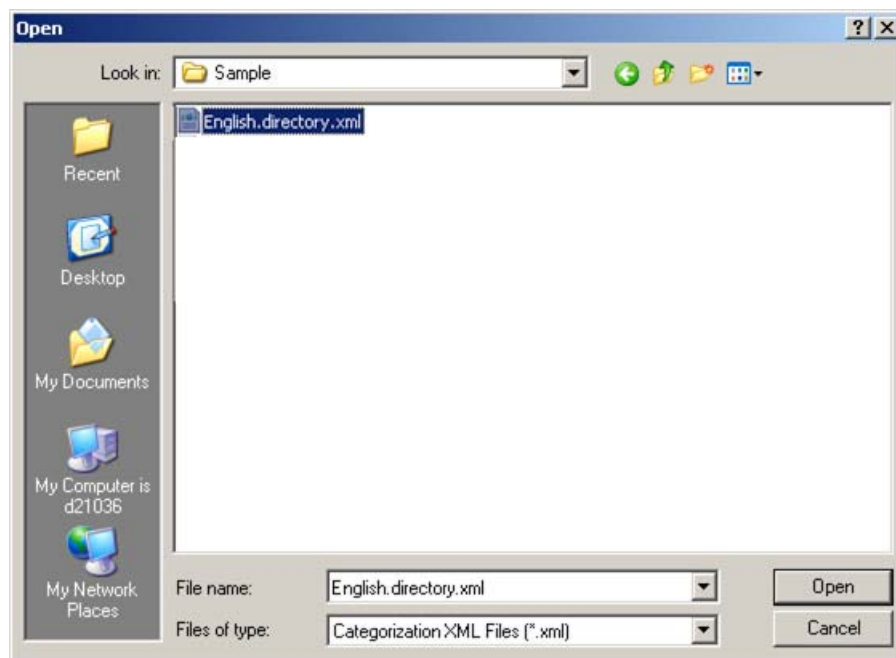
For more information about XML taxonomy files and for an example of this file type, see Appendix B *Program Files* on page 511.

To import a taxonomy of categories, complete the following steps:

1. Beginning with Step 1 on page 101 work through Step 7 on page 102 to create a new project.
2. After the language node is added to the taxonomy, right-click the language icon and select **Import Categorizer from XML** in the drop-menu that appears.

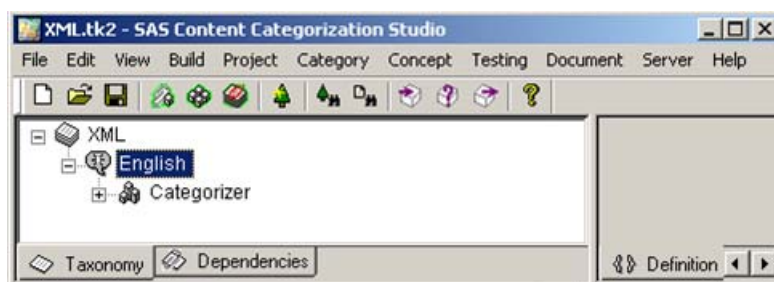


3. The Open window appears.





4. Double-click on an XML file icon that uses the same language as the language in the selected taxonomy branch (.directory.xml extension for categories).
5. Click **Open**.



6. Right-click on the language node and select **Expand Fully** to access the imported taxonomy. All of the project tabs and windows display the imported information.



7. (Optional) Select **File --> Save**. For more information, see Section 2.4 *Saving the Project* on page 108.
8. (Optional) Select **Project --> Settings** and specify your project settings. For more information, see Section 2.7 *Specifying Project Settings* on page 130.

---

## 2.4 Saving the Project

### 2.4.1 Overview of Saving the Project

By default, the project is saved every time you test your project. However, you might want to save your project as you build the taxonomy and define concepts. You might also want to save duplicate projects at different stages of taxonomy development.

### 2.4.2 Manually Save an Existing Project

The name of the project that appears in the title bar is the same name of the project folder that the application automatically creates. Manually save a project to keep different stages, or versions, of the project during development. See the following example:

```
c:\ProgramFiles\Teragram\tk240\projects\XML.tk2.
```

To save your project, select **File --> Save Project**.

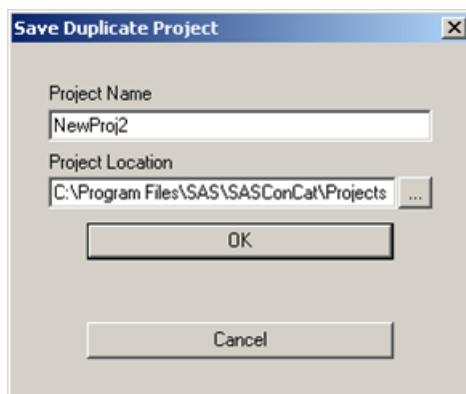
---


### 2.4.3 Save a Duplicate Project

You can save your project as a duplicate project using another name. Use this operation when you want to preserve a specific stage or version of the project.

To create a duplicate project, complete these steps

1. Select **File --> Save Project As**. The Save Duplicate Project window appears.



2. Type the name of the duplicate project into the **Project Name** field. For example, enter `NewProj2`.
3. (Optional) Click  to the right of the **Project Location** field to access the Select a Directory window. Alternatively, use the default project name and path that is automatically entered for you.
4. Click **OK**. The renamed project appears in the **Taxonomy** tab.

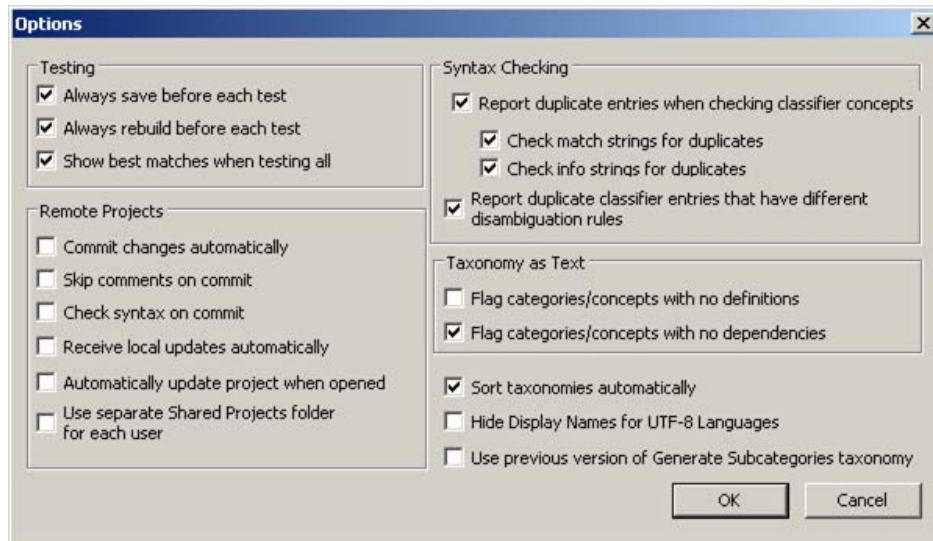
---

## 2.4.4 Automatically Save Your Project Before Testing

You can automate the process of saving your project before testing.

To specify this automated process, complete these steps:

1. Select **Edit --> Options** and the Options window appears.



2. Select **Always save before each test**.
3. Click **OK**.

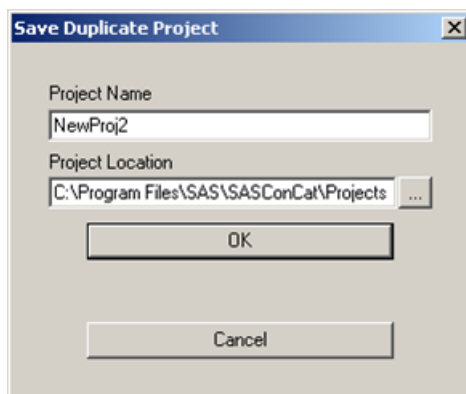
---


## 2.4.5 Save a Duplicate Project

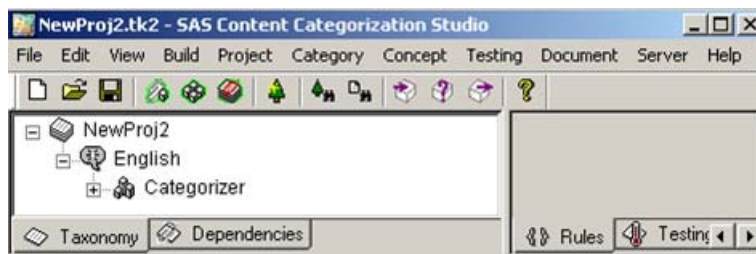
You can save your project as a duplicate project using another name. This is beneficial when you want to preserve a specific stage or version of the project.

To create a duplicate project, complete these steps

1. Select **File --> Save Project As**. The Save Duplicate Project window appears.



2. Type the name of the duplicate project into the **Project Name** field.
3. (Optional) Click  to the right of the **Project Location** field, or use the default project name and path that is automatically entered for you.
4. Click **OK**. The renamed project appears in the **Taxonomy** tab.



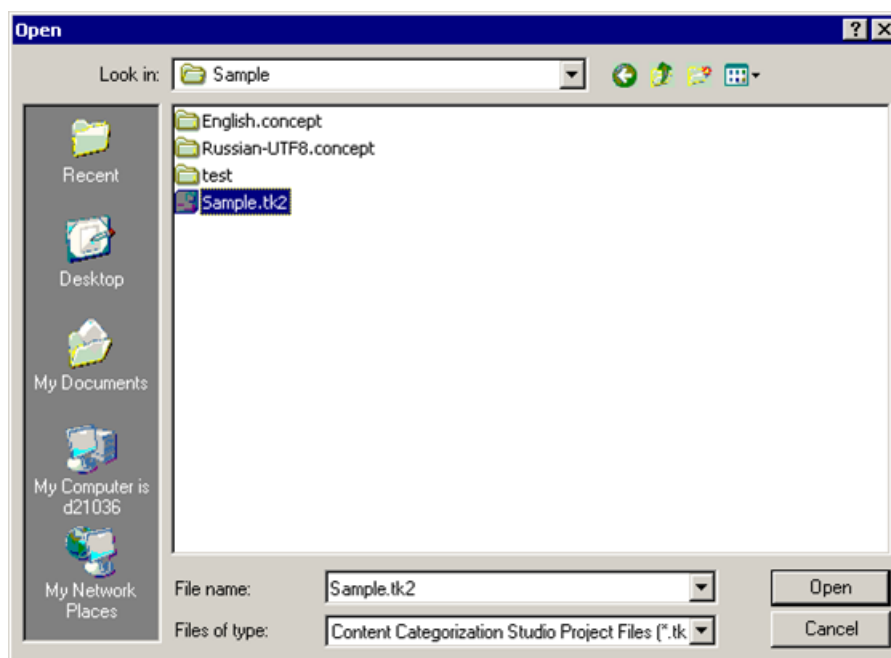
---


## 2.5 Access an Existing Project

Access a project that you created and then closed using this operation.

To access an existing project, complete these steps:

1. Select **File --> Open Project** and the Open window appears.



2. Click  to navigate through the program files and the `Projects` folder on your hard drive until you locate a `.tk2` file. For example, find `Sample.tk2`.

---

**Hint:** The files for your projects are saved in a Windows folder that has the project name. For example, the files for the Sample project are stored in the Sample folder. See the following path:

C:\Program Files\SAS\TK240\Projects\Sample.

---

- 
3. Double-click the selected project and it appears in the user interface.



## 2.6 Set Installation-Specific Operations

### 2.6.1 Automate Operations for the Installation

To automate several operations across an installation of SAS Content Categorization Studio, use the Options window. These operations affect the current project, as well as all of the other projects that you develop.

---

**Hint:** It is important to remember that when you specify your options that these settings affect all of the projects unless you reset these specifications.

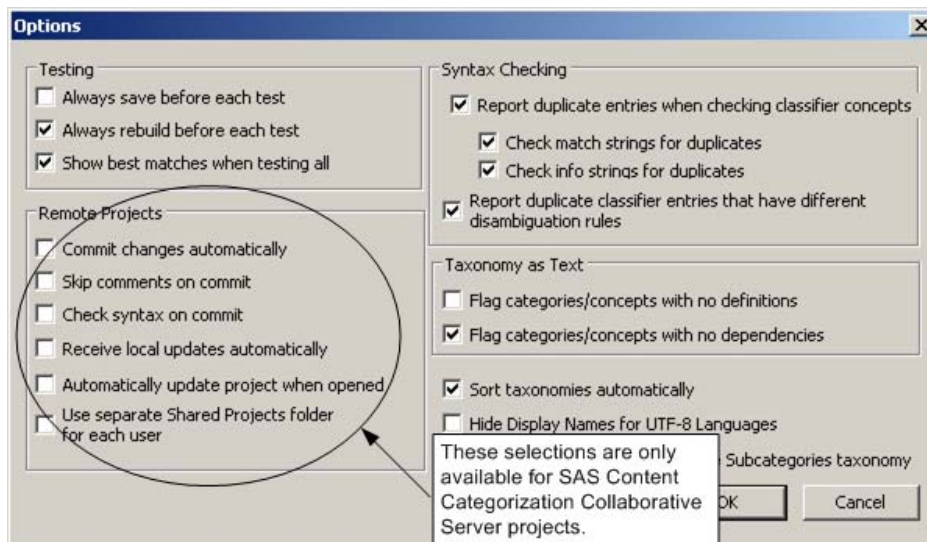
---

The project settings can be changed at any point during project development.

---

To reset the default settings in the Options window, complete this step:

Select **Edit --> Options** and the Options window appears. The **Remote Projects** options, circled below, are used only for SAS Content Categorization Collaborative Server. The selected check boxes are enabled by default.





---

## 2.6.2 The Option Window Settings That Affect Testing

### 2.6.2.A Save before Each Test

Select **Edit --> Options --> Always save before each test** to automatically save the changes made to your project before testing. If you do not save your changes before you select **File --> Exit**, the SAS Content Categorization Studio confirmation window appears.

*Display 2-1 SAS Content Categorization Studio Window*



### 2.6.2.B Always Rebuild before Each Test

By default, **Always rebuild before each test** is selected in the Options window. When you click **TEST** in the **Testing** tab, the binary file for your project is automatically rebuilt before it is tested.

*Display 2-2 Testing Results*



---

If you deselect **Always rebuild before each test**, make changes to your project, and try to test without rebuilding your project, the SAS Content Categorization Studio status window appears.

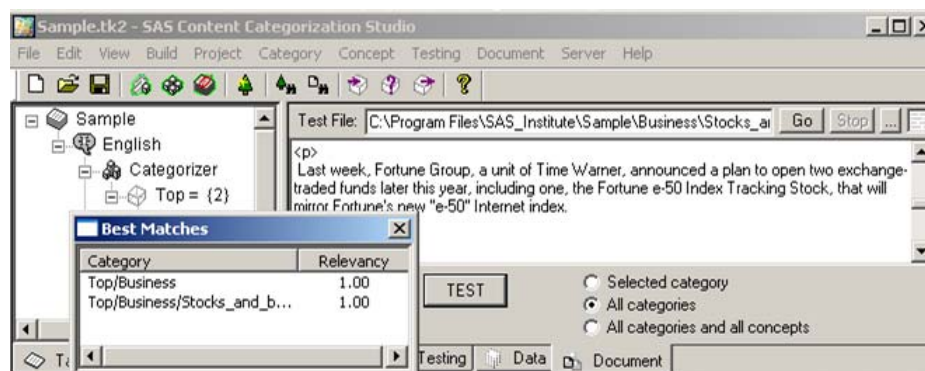
*Display 2-3 SAS Content Categorization Studio Window*



### 2.6.2.C See the Best Matches

By default, **Show best matches when testing all** is selected. When you test **All categories** or **All categories and concepts**, the Best Matches window appears. Otherwise, results appear only in the Taxonomy pane.

*Display 2-4 Best Matches Window*



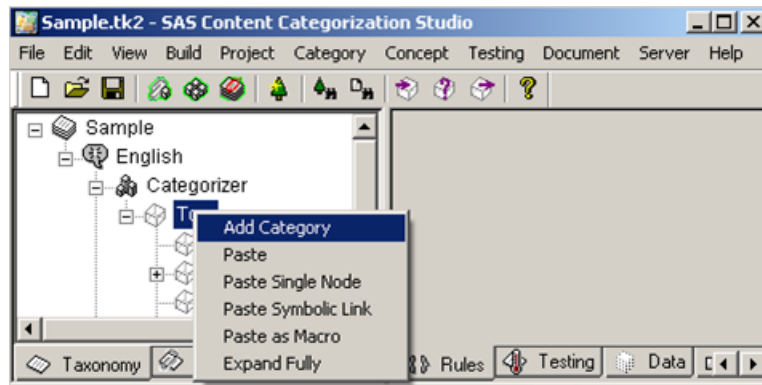
---

### 2.6.3 Automatically Sort the Taxonomy

An automatically alphabetized taxonomy makes it easy to locate categories and concepts. For this reason, **Sort taxonomies automatically** is selected by default in the Options window.

To add a new category and to automatically sort the taxonomy alphabetically, complete these steps:

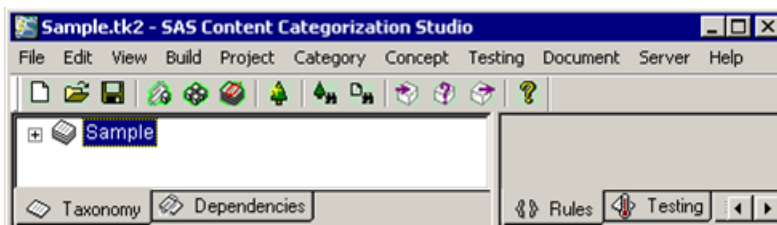
1. Make sure that the default selection **Sort taxonomies automatically** is selected in the Options window.
2. Right-click on the **Top** node in the **Taxonomy** tab and select **Add Category**.



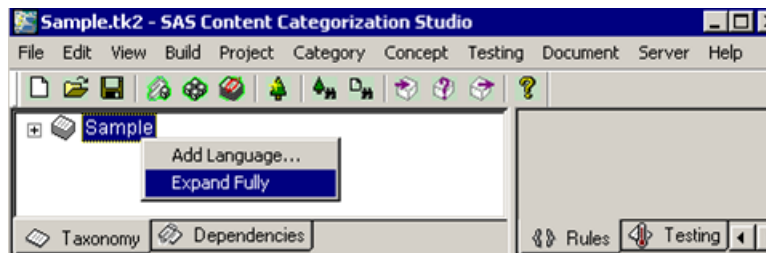
3. Type the name of the new category into the text box that appears. For example, enter New.



4. To contract the taxonomy, click the minus (-) sign to the left of each parent node. This includes Top, Categorizer, English, and the project name node.

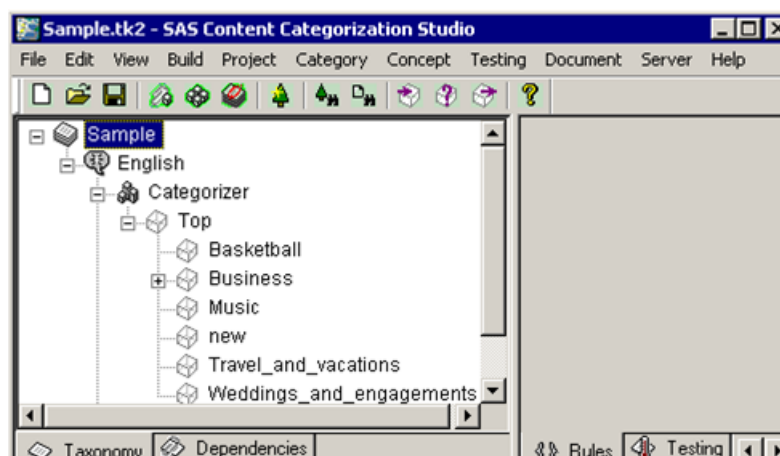


5. To see the reordered taxonomy, right-click on the project name node and select **Expand Fully** from the drop-down menu that appears.



---

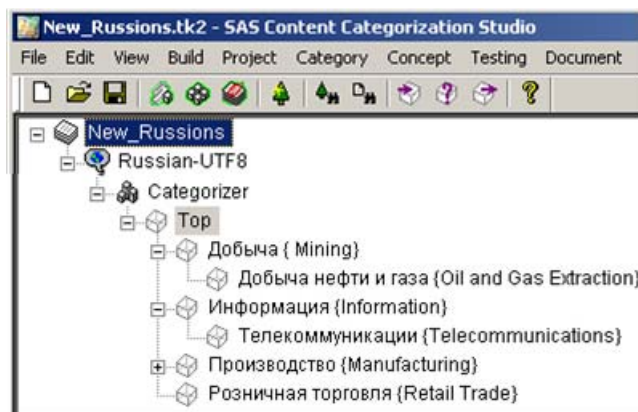
When the taxonomy expands, it is automatically reordered.



## 2.6.4 Hiding Display Names for UTF-8 Languages

If **Hide Display Names for UTF-8 Languages** is not selected, both the UTF-8 version and the Latin 1 versions of the category names are displayed. This is true for UTF-8 languages only.

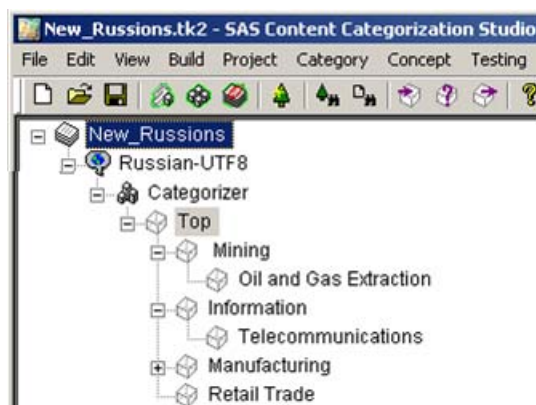
*Display 2-5 UTF-8 and Latin 1 Category Names*



---

If you select **Hide Display Names for UTF-8 Languages**, the **Taxonomy** tab displays only the Latin 1 category names.

*Display 2-6 Latin 1 Category Names*



## 2.6.5 Checking Syntax for Classifier Concepts

### 2.6.5.A Report Duplicate Entries When Checking Classifier Concepts

Use this case-sensitive operation to find two or more instances of the same entry in classifier concept definitions. For more information about writing definitions, see Section 18.2 *Writing a Classifier Definition* on page 442. These entries can be limited to either the `match_key` or to the `returned_information` strings that together define a classifier concept.

If you select both **Check match strings for duplicates** and **Check info strings for duplicates**, SAS Content Categorization Studio locates any concept definition lines where both entries are duplicated.

---

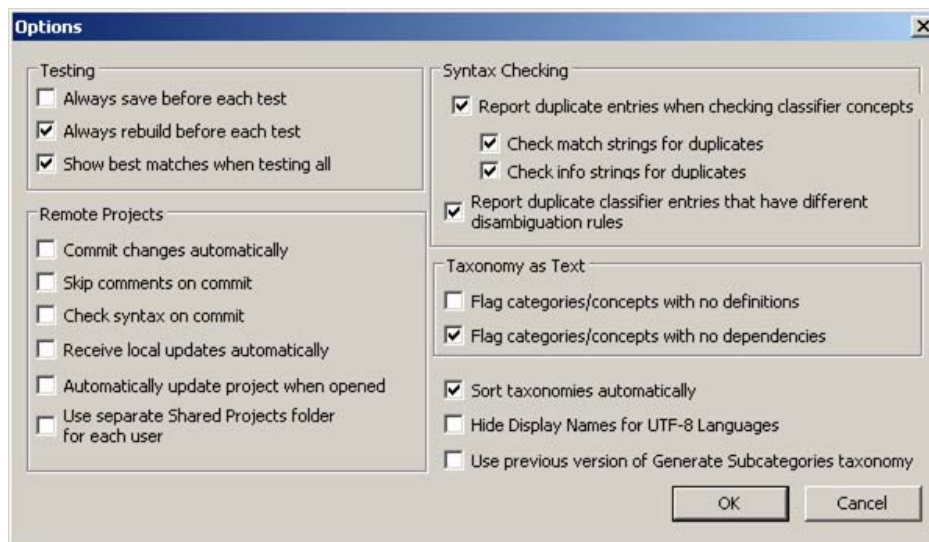
**Note:** The **Check match strings for duplicates** and **Check info strings for duplicates** operations are enabled only when you select **Report duplicate entries when checking classifier concepts**.

---

---

To perform syntax checking that locates duplicate information strings in classifier concepts only, complete these steps:

1. Select **Edit --> Options** and the Options window appears.
2. Click the **Report duplicate entries when checking classifier concepts** check box in the Options window.



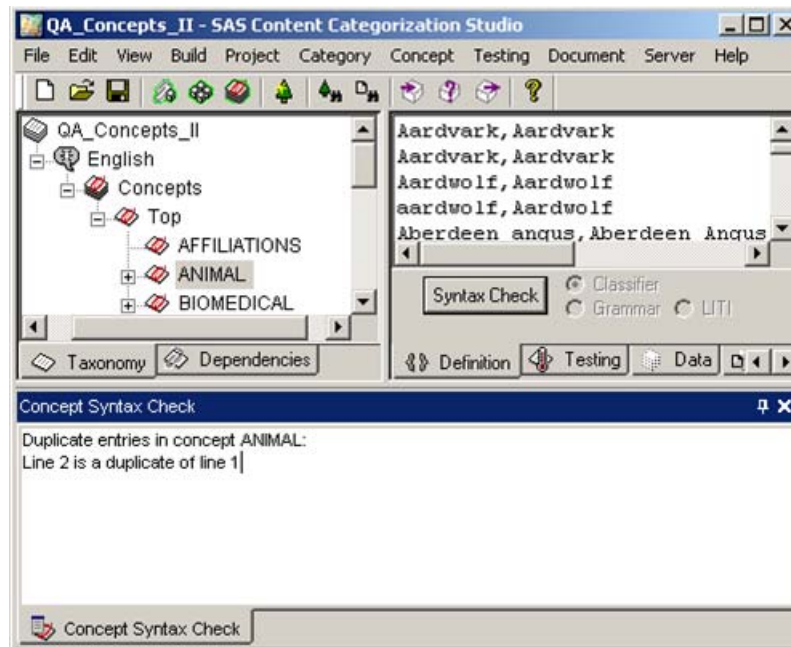
3. When you select **Report duplicate entries when checking classifier concepts**, the **Check match strings for duplicates** and **Check info strings for duplicates** selections become available. Click one, or both, of these check boxes. For more information, see Section 2.8 *The Options Window* on page 47.

If you do not select one of these operations, the following SAS Content Categorization Studio window appears.



4. Click **OK**.

5. Select a concept with a classifier definition and click **Syntax Check** in the **Definition** tab.



6. The **Concept Syntax Check** tab appears at the bottom of the user interface. Click on the line number in the **Concept Syntax Check** tab that reports the duplicate. The cursor moves to that line in the **Definition** tab. Edit the selected line in the concept definition.



---

### 2.6.5.B Report Duplicate Classifier Entries That Have Different Disambiguation Rules

Write Boolean rules to differentiate, or disambiguate, a term that is used in different contexts. For example, the word *server* has multiple meanings. You can disambiguate this term in the contexts where it appears. For example, the term *project on the server* applies to a computer while the term *buffet server* applies to furniture.

For this reason, you could write disambiguation rules that match specified instances of *server*. However, you might want to check all of your disambiguation rules to see whether different usages of the word *server* are specified. These duplicate instances could occur in the match string, info string, or in the Boolean rule section of the classifier string. You can choose to run this check before, or after, you write a new disambiguation rule.

To use the Options window to locate duplicate classifier strings with different definitions, complete the following steps:

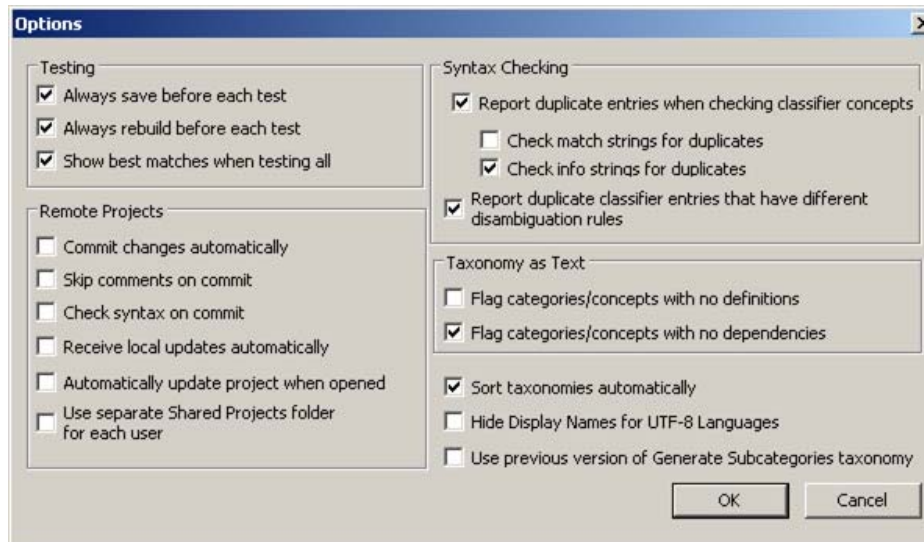
1. Write Boolean disambiguation rules in the classifier definitions for the concepts in your taxonomy. For more information, see Section 18.4 *Using Disambiguation to Increase Matching Precision* on page 453.

---

**Note:** You can write more than one disambiguation rule into each classifier definition.

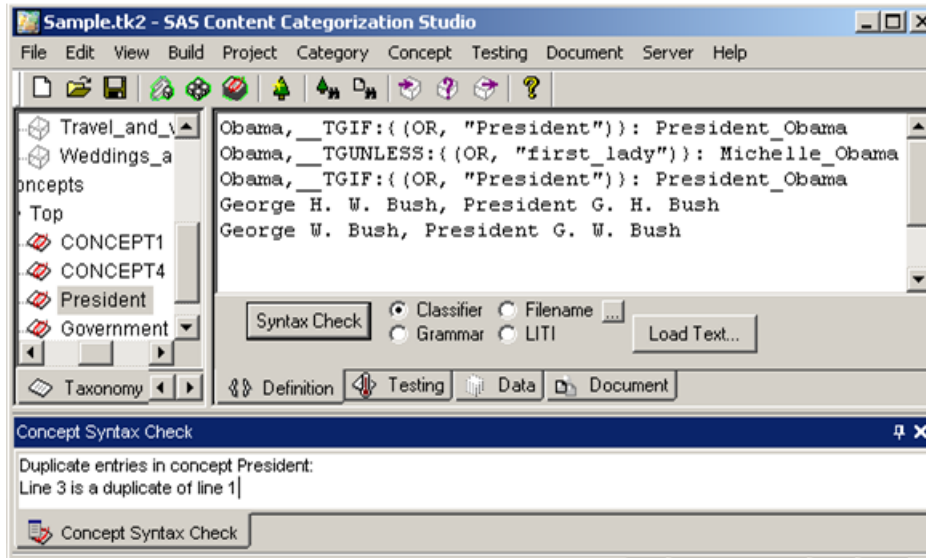
---

2. Select **Edit --> Options** and the Options window appears.



3. Under **Syntax Checking** select the **Report duplicate entries when checking classifier concepts** check box.
4. Select either, or both, **Check match strings for duplicates** and **Check info strings for duplicates**.
5. Select **Report duplicate classifier entries that have different disambiguation rules**.
6. Click **OK**.

- Click **Syntax Check** and the **Concept Syntax Check** tab appears at the bottom of the user interface.



- Click **X** to close the Concept Syntax Check window.

---

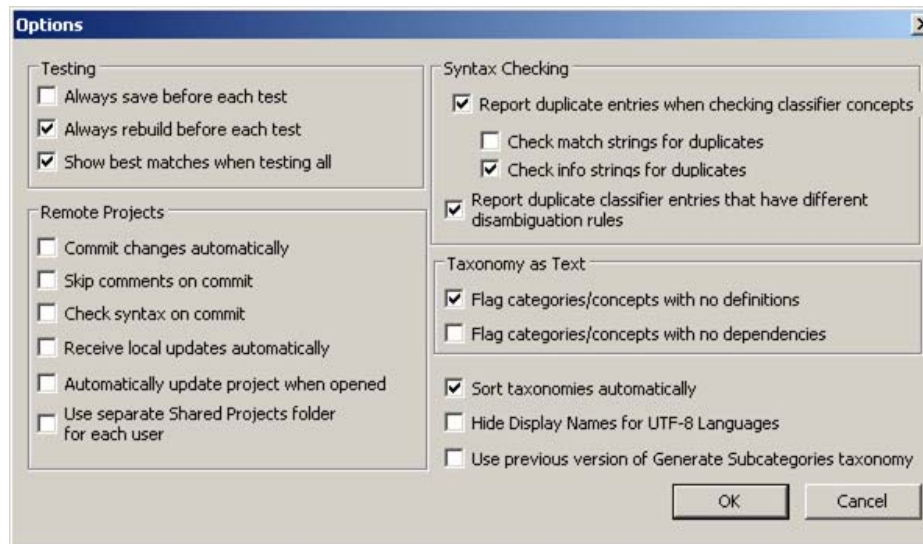
## 2.6.6 View the Taxonomy as Text

### 2.6.6.A Flag the Categories and Concepts with No Definitions

Flag categories and concepts without definitions before testing or uploading branches of your taxonomy.

To flag concepts without definitions, complete the following steps:

1. Select **Edit --> Options** and the Options window appears.
2. Click **Flag categories/concepts with no definitions**.

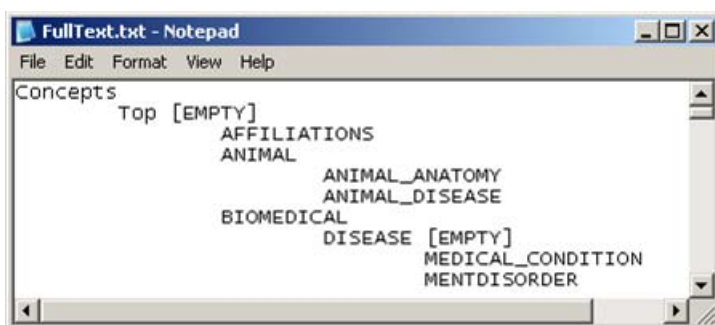


3. Click **OK**.

4. Select the Concepts (or Categorizer) node.



5. Select **View --> Taxonomy as Text**.
6. *Notepad* appears displaying the taxonomy of your project with the [EMPTY] message to the right of any nodes that have no concept definition. For example, Disease has no definition.



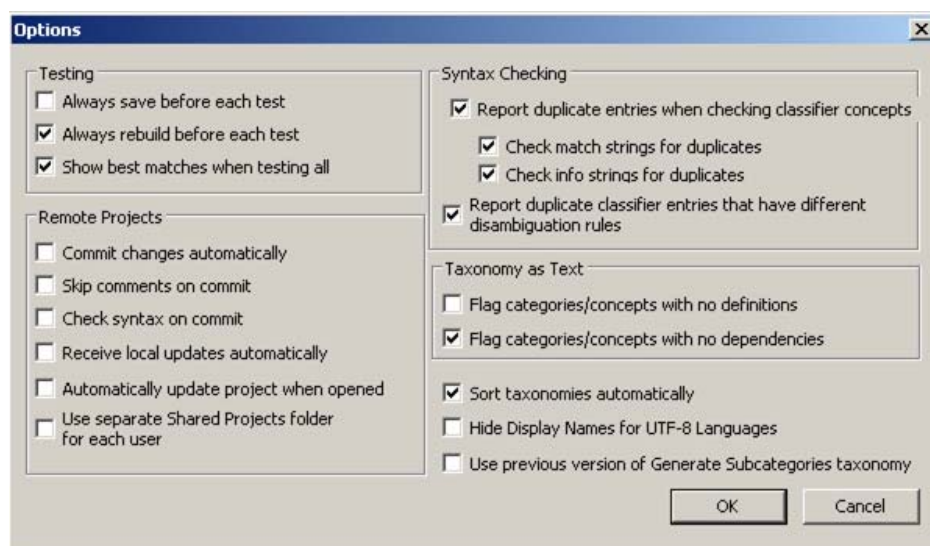
---

### 2.6.6.B Flag the Categories and Concepts with No Dependencies

Flag categories and concepts without dependencies before you delete any nodes in the taxonomy. Use this operation to prevent unintended rule changes when one category is dependent on another node for part of its rule.

To access this window and see this display, complete these steps:

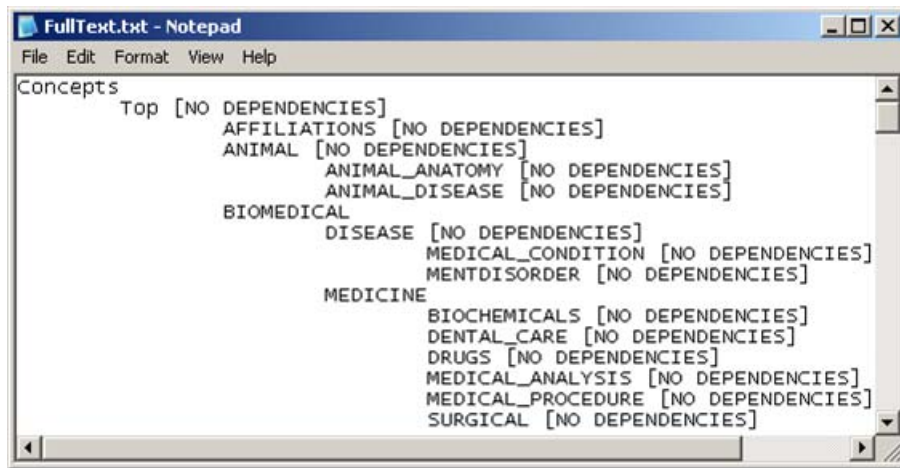
1. Select **Edit --> Options** and the Options window appears.



2. Select **Flag categories/concepts with no dependencies**.
3. Click **OK**.
4. Select **Build --> Compile Concepts** and **Build Rulebased Categorizer**, if your project contains both taxonomy branches.
5. Select **View --> Taxonomy as Text**.

---

A *Notepad* window appears and displays dependency information. This message [NO DEPENDENCIES] appears to the right of each category or concept that has no dependent relationship on another rule in the taxonomy.



```
FullText.txt - Notepad
File Edit Format View Help
Concepts
  Top [NO DEPENDENCIES]
    AFFILIATIONS [NO DEPENDENCIES]
    ANIMAL [NO DEPENDENCIES]
      ANIMAL_ANATOMY [NO DEPENDENCIES]
      ANIMAL_DISEASE [NO DEPENDENCIES]
    BIOMEDICAL
      DISEASE [NO DEPENDENCIES]
        MEDICAL_CONDITION [NO DEPENDENCIES]
        MENTDISORDER [NO DEPENDENCIES]
      MEDICINE
        BIOCHEMICALS [NO DEPENDENCIES]
        DENTAL_CARE [NO DEPENDENCIES]
        DRUGS [NO DEPENDENCIES]
        MEDICAL_ANALYSIS [NO DEPENDENCIES]
        MEDICAL_PROCEDURE [NO DEPENDENCIES]
        SURGICAL [NO DEPENDENCIES]
```

---

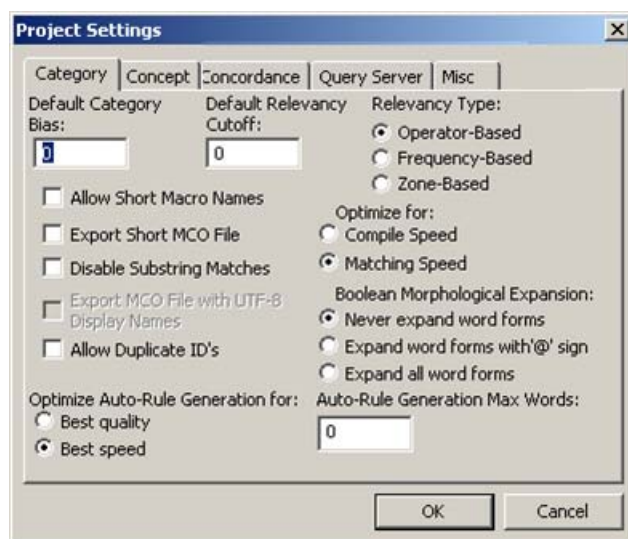
## 2.7 Specifying Project Settings

### 2.7.1 Specifying the Initial Category Project Settings

#### 2.7.1.A Specify Category Operations

Use the **Category** tab to set the project-wide settings that affect all of your categories for the selected branch of the current project. Some of these settings can be overridden in the **Data** tab for individual categories.

*Display 2-7 Default Settings for the Category Tab*



To specify settings in the **Category** tab, complete these steps:

1. Apply all of the settings in the **Category** tab that are relevant to the documents that you want to categorize. For more information, see Section 2.9.2 *The Project Settings for Categories* on page 49.

---

**Note:** If the **Export MCO file with UTF-8 Display Names** is not enabled, the project does not contain a language with UTF-8 encoding.

---

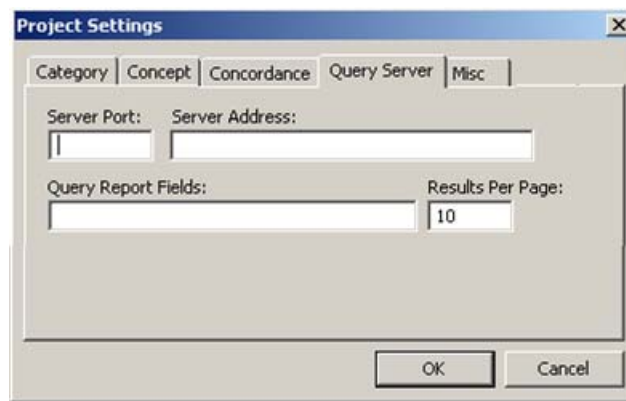


2. Click **OK**.
3. Select **Build --> Build Rulebased Categorizer** or **Build Statistical Categorizer**.
4. Select **File --> Save**.
5. Continue to define and test your categories.

### 2.7.1.B Specify Query Operations

Use these settings with Boolean category rules when you want to query an index on a remote server. Set the project-wide settings for the server in the **Query Server** tab.

*Display 2-8 Query Server Default Settings*



To specify settings in the **Query Server** tab, complete these steps:

1. Apply all of the settings in the **Query Server** tab that are relevant at this time. For more information, see Section 2.9.2.B *The Query Server Tab* on page 53.
2. Click **OK**.
3. Select **Build --> Build Rulebased Categorizer** or **Build Statistical Categorizer**.
4. Select **File --> Save**.

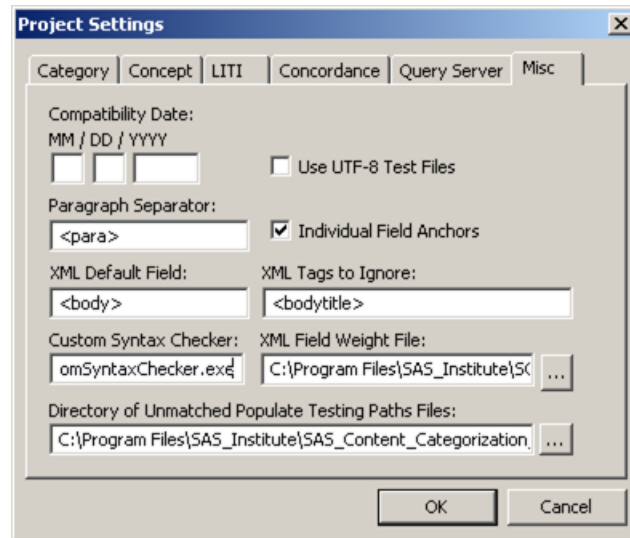
- 
5. SAS Content Categorization Studio searches the index for documents that match the categories in your project.

## 2.7.2 Specify Miscellaneous Operations

The **Misc** tab has project-wide settings that affect the application, categories, and concepts.

To specify settings in the **Misc** tab, complete these steps:

1. Select **Project --> Settings** and the **Misc** tab appears.



2. Use all of the settings that are relevant at this time. For more information, see Section 2.9.3 *The Misc(ellaneous) Tab for Categories and Concepts* on page 54.
3. Click **OK**.

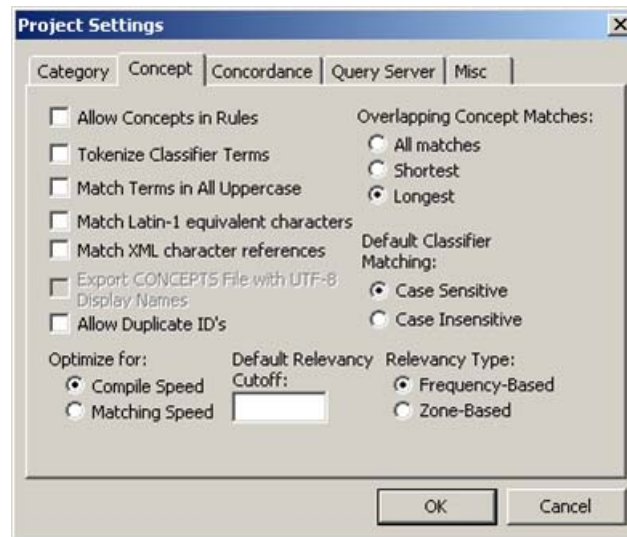
---

## 2.7.3 Specifying Concept Project Settings

### 2.7.3.A Specify Concept Operations

Use the **Concept** tab to set the project-wide settings that affect how the concept definitions that you specify are matched to input documents. These operations apply to the currently selected branch of the taxonomy. Some of these settings can be overridden in the **Data** tab for individual concepts.

*Display 2-9 Default Settings for the Concept Tab*



---

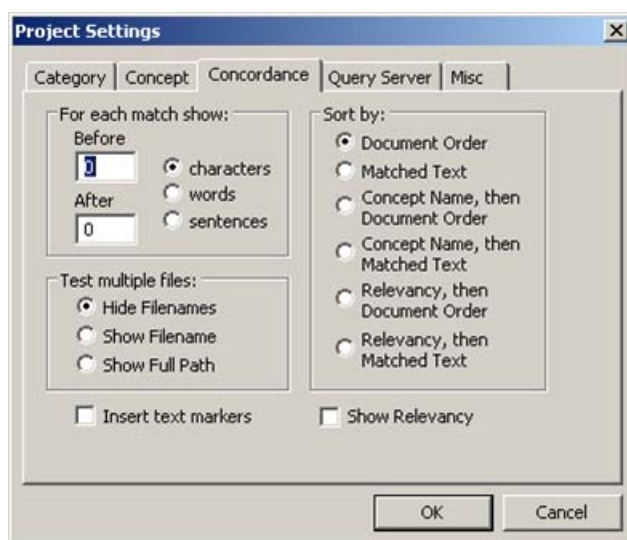
To specify settings in the **Concepts** tab, complete these steps:

1. Use all of the settings that are relevant at this time. For more information, see Section 2.9.4.A *The Concept Tab* on page 57.
2. Click **OK**.
3. Select **Build --> Compile Concepts**.
4. Select **File --> Save**.
5. Continue to write your concept definitions.

### 2.7.3.B Specify the Concordance Operations

Set the project-wide settings for the concordance operation. This operation displays the matched terms in input documents according to the specifications that you set here. Specify these settings in the **Concordance** tab.

*Display 2-10 Concordance Default Settings*



---

To specify settings in the **Concordance** tab, complete these steps:

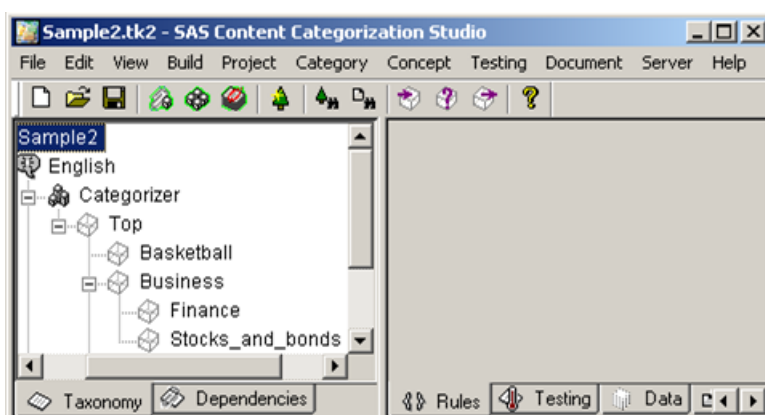
1. Select all of the settings that are relevant at this time. For more information, see Section 2.9.4.B *The Concordance View* on page 60.
2. Click **OK**.
3. Select **Build --> Compile Concepts**.
4. Select **File --> Save**.
5. Begin testing the concepts.

## 2.8 Navigating through Categories and Concepts

After you create categories and concepts, the **Taxonomy** tab displays a hierarchical view of the individual categories and concepts that comprise your taxonomy. You can use standard Windows controls to navigate through, and to manipulate, these individual categories and concepts.

See an example of a **Taxonomy** tab after some of the categories and concepts are defined. The **Taxonomy** tab below displays a taxonomy of categories.

*Display 2-11 Categories Displayed in the Taxonomy Tab*



`Top`, is the permanent name for the first node in the category or concept hierarchy in the **Taxonomy** tab. Every category below `Top`, such as

---

Basketball or Business, is a child of the Top node. These categories, in turn, can also be the parents of other subcategories or children. For example, Business is the parent of the child categories Finance and Stocks\_and\_bonds.



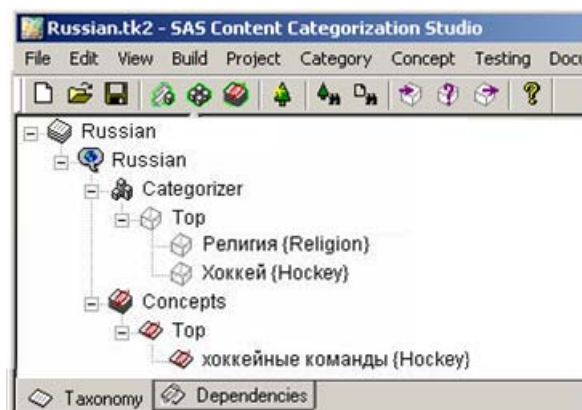
The Business category has a next to it. This sign indicates that Business has one or more subcategories that are now displayed.

## 2.9 Export a UTF-8 Binary File

If you created a taxonomy using UTF-8 display names, read this section before you upload your categories or concepts to SAS Content Categorization Server. The presence of two names (a UTF-8 name followed by the internal, Latin 1 [ASCII] name) in the taxonomy enable you to export two .mco or .concepts files.

Select **File --> Export with UTF-8 display names** to generate .mco and .concepts (binary) files. These files output the UTF-8 category or concept names and preserve the Latin-1 names internally. If you do not use this selection, only the Latin-1 names, are preserved.

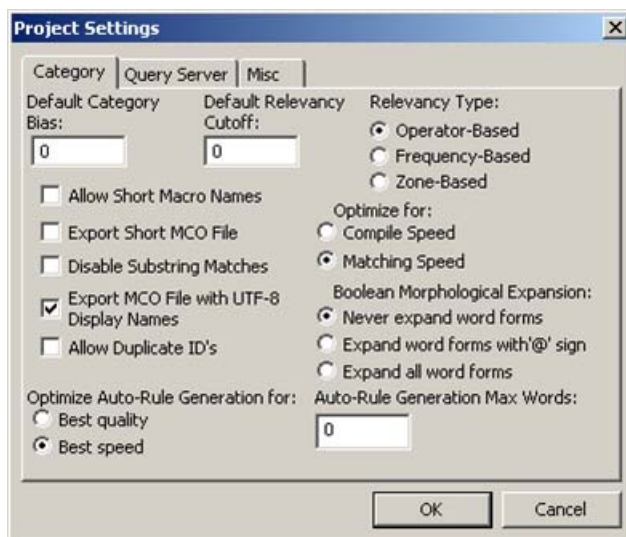
*Display 2-12 UTF-8 Display Names*



---

To perform this export operation, complete these steps:

1. Select **Project --> Settings** and the **Category** tab appears.



2. Select **Export .mco File with UTF-8 Display Names**.
3. Click **OK**.
4. (Optional) Use the **Concepts** tab, if necessary, to repeat this process.

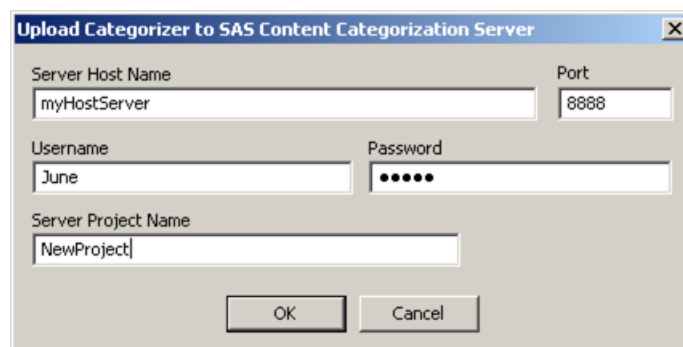
---

## 2.10 Upload the Categorizer or Concepts to SAS Content Categorization Server

After you build and test the taxonomy, use the Upload Categorizer to SAS Content Categorization Server (or Concepts) window. You can specify the requirements that are necessary to upload the categorizer (.mco file) or the concepts extractor (.concepts file) to the server in this window.

To upload the .mco file, complete these steps:

1. Select **Build --> Upload Categorizer**. The **Upload Categorizer to SAS Content Categorization Server** window appears.



The screenshot shows a dialog box titled "Upload Categorizer to SAS Content Categorization Server". It contains the following fields and controls:

- Server Host Name:** A text field containing "myHostServer".
- Port:** A text field containing "8888".
- Username:** A text field containing "June".
- Password:** A text field with masked characters (dots).
- Server Project Name:** A text field containing "NewProject".
- Buttons:** "OK" and "Cancel" buttons at the bottom.

2. Use the settings shown above and described in Section 2.11 *The Uploading the Categorizer, or Concepts, to SAS Content Categorization Server Window* on page 68 to upload your categories.
3. Click **OK**.
4. Begin applying the rules to input documents using SAS Content Categorization Server. For more information, see *SAS Content Categorization Server: User's Guide*.



---

# Part 1: Categories

---

- Chapter 4: *Categorization on page 141*
- Chapter 5: *Creating Categories on page 157*
- Chapter 6: *Using the Statistical Categorizer on page 183*
- Chapter 7: *Automatic Rule Generator Tool on page 201*
- Chapter 8: *Rule-Based Categorizers on page 211*
- Chapter 9: *Relevancy and the Settings That Affect Relevancy on page 247*
- Chapter 10: *Rule-Based Categorizer: Linguistic Terms on page 263*
- Chapter 11: *Rule-Based Categorizer: Boolean Terms on page 285*



---

# Chapter: 4

## Categorization

---

- *Overview of Categorization*
- *How to Categorize Documents*
- *Choosing a Taxonomy Type*
- *Planning Your Taxonomy*
- *Choosing a Categorizer*
- *Optimizing Precision*
- *About the Testing and Training Documents*
- *How to Build Categorizers*

### 4.1 Overview of Categorization

This chapter provides an essential high-level overview of the processes that you implement when you develop a taxonomy of defined categories.

Use this chapter to make choices about how you generate your rules. For this task, you can use one of the following technologies:

- Statistical categorizer
- Automatic rule generator tool: use this rule generator to extract terms that you can use with a rule-based categorizer
- Rule-based categorizer, whether linguistic or Boolean

---

## 4.2 How to Categorize Documents

Before you create a taxonomy, or a categorization structure, consider how to categorize your texts.

To develop a taxonomy of categories, complete these steps:

1. Analyze your documents to locate key ideas, terms, or other recognizable attributes. These are the unique identifiers for each group of texts that you are categorizing.
2. Consider end-user requirements for information access. In other words, what attributes of each group would your end users consider significant?
3. Carefully define the category names so that they represent rules that are neither too broad or too narrow. In other words, your category rules should include all of the appropriate texts and exclude the documents that are not good matches. For example, you might define the categories *Sports* and *Baseball*. However, documents that could be categorized under *Baseball* could also be categorized under *Sports*. A better set of categories might be *Football* and *Baseball*.
4. Decide to use a flat or a hierarchical taxonomy. For more information, see Section 4.3 *Choosing a Taxonomy Type* on page 143. To continue with the example provided in Step 3. above, *Football* and *Baseball* might be category names in a flat taxonomy. In a hierarchical taxonomy, however, it might be advisable to make *Football* and *Baseball* children of the parent category *Sports*.
5. If you use either a rule-based categorizer with automatically generated rules or a statistical categorizer, ensure that your categories are not too precisely defined. For example, choose categories like *Sports* and *Theater* instead of *Football* and *Baseball*. For greater precision, write your own rules. When you use the rule-based categorizer, write precise rules to match *Football* and *Baseball*.

These five steps form an important background planning component for your project.

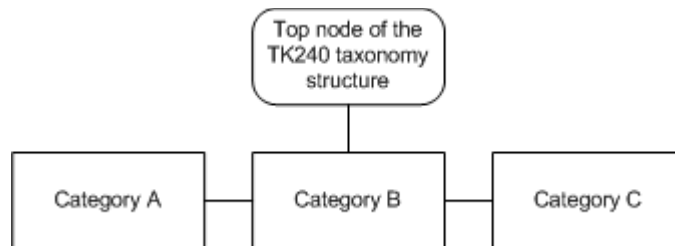
---

## 4.3 Choosing a Taxonomy Type

As you plan your taxonomy, or the overall organization of your categories, you determine whether to create a flat or a hierarchical taxonomy.

A flat taxonomy is an organizational structure where all of the categories are on the same level in the taxonomy. There are *no* subcategories, or children, for any of the categories organized into this type of structure.

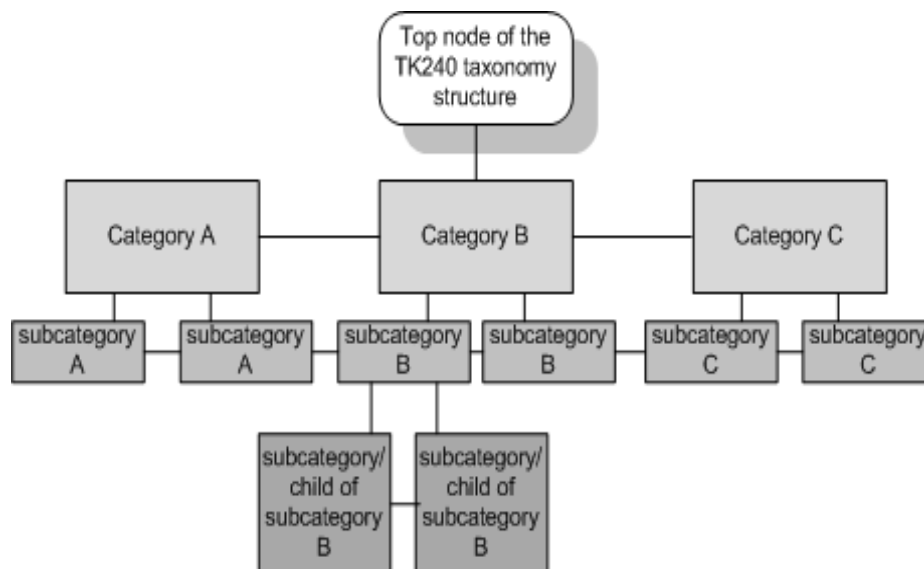
*Figure 4-1 Flat Taxonomy*



---

A hierarchical taxonomy, on the other hand, is a structure where one or more categories in the taxonomy have at least one child and where some categories could be nested. In other words, child categories can also be the parents of other children.

*Figure 4-2 Hierarchical Taxonomy*



---

## 4.4 Planning Your Taxonomy

### 4.4.1 A Sample Flat Taxonomy

See the following example of a flat taxonomy that consists of five categories (with no subcategories):

*Example 4-1: Sample Flat Taxonomy*

Top  
Politics  
Business  
Education  
Recreation  
Sports

When viewed within the SAS Content Categorization Studio **Taxonomy** tab, this example above looks similar to the example displayed below.

*Display 4-1 Flat Taxonomy*



---

## 4.4.2 A Sample Hierarchical Taxonomy

In a hierarchical taxonomy there is an interrelationship between the categories that are arranged in parent-child (category-subcategory) relationships. The example shown below provides one example of a taxonomy consisting of three top-level categories (parents) that are displayed in bold-faced type with their subcategories (children) in regular type. The children of the child subcategories appear in italic type:

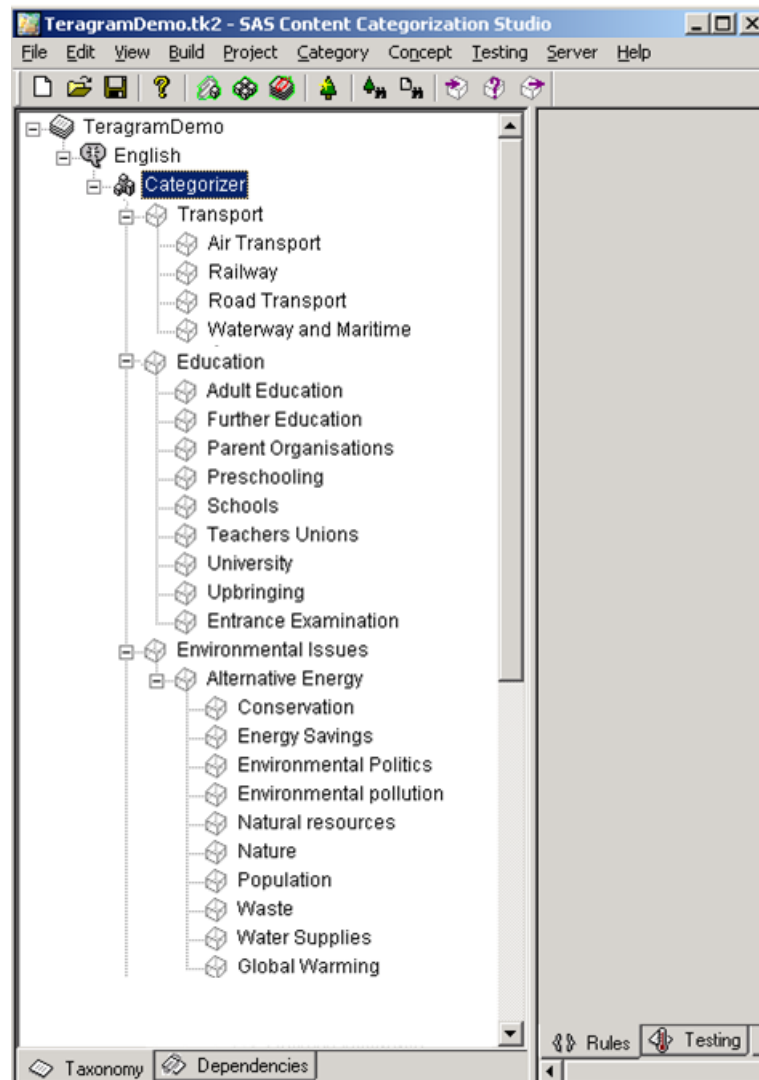
*Example 4-2: Sample Hierarchical Taxonomy*

Top  
**Transport**  
Air Transport  
Railway  
Road Transport  
Waterways and Maritime  
**Education**  
Adult Education  
Further Education  
Parent Organisations  
Preschooling  
Schools  
Teachers Unions  
University  
Upbringing  
Entrance Examination  
**Environmental Issues**  
Alternative Energy  
    *Conservation*  
    *Energy Savings*  
    *Environmental Politics*  
    *Environmental pollution*  
    *Natural resources*  
    *Nature*  
    *Population*  
    *Waste*  
    *Water Supplies*  
    *Global Warming*

When this taxonomy is displayed in the SAS Content Categorization Studio **Taxonomy** tab, it appears as shown below:



Display 4-2 Sample Taxonomy



---

### 4.4.3 Modifying the Taxonomy

After you create your initial taxonomy, build the categorizer, and create testing sets of documents, you can modify the taxonomy by testing the categories to see how well they work. This testing process enables you to see where you should make any of the following taxonomy changes:

- Add categories.
- Remove categories.
- Change the criteria for category membership.

### 4.4.4 View the Taxonomy as Text

See the taxonomy structure as text when you select **Taxonomy as Text**. Check for dependencies and missing definitions in the `FullText.txt` window that appears. You can also use this operation to export a taxonomy into a file format that can be easily read by another application.

Before you use the **Taxonomy as Text** operation, choose either or both of the selections that are available under the **Taxonomy as Text** heading in the Options window:

**Flag categories/concepts with no definitions**

See a list of all of the taxonomy nodes that have no rules or definitions.

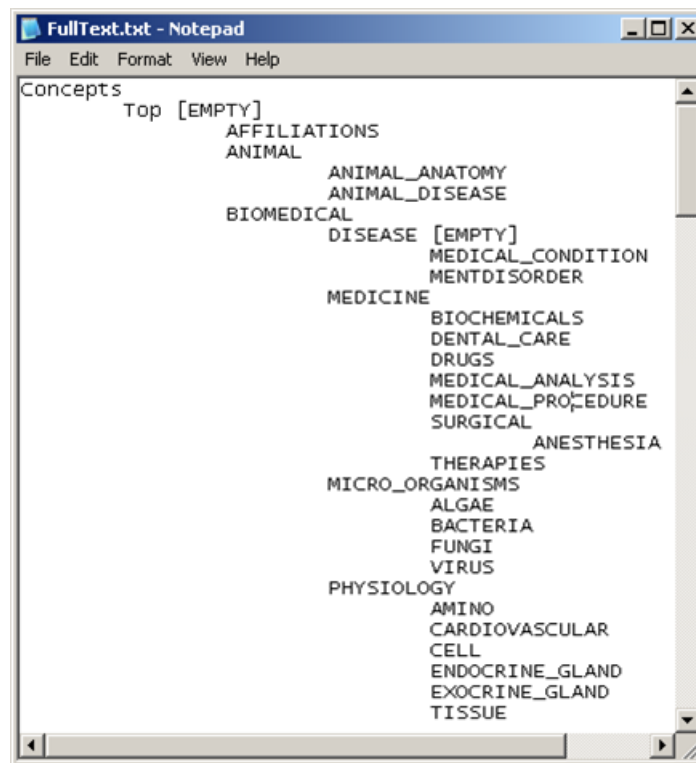
**Flag categories/concepts with no dependencies**

See a note for each node that has a rule that is not dependent on another node.

To see the taxonomy in text format, complete these steps:

1. Click the `Categorizer` or `Concepts` node in the **Taxonomy** tab.
2. Select **View --> Taxonomy as Text**. A *Notepad* window appears with the taxonomy displayed in text format.

- 
3. Analyze the text to make sure the taxonomy appears as expected.



4. Click X.

---

## 4.5 Choosing a Categorizer

### 4.5.1 The Two Basic Categorizer Types

Use SAS Content Categorization Studio, to specify a variety of rules that are dependent on the two basic rule types included with SAS Content Categorization Studio. Review the general benefits of each type of categorizer before you select one:

#### Statistical categorizer

an automated, almost out-of-the-box product that defines category rules based on a statistical analysis of input documents. After the statistical categorizer is trained, it automatically assigns each document to a category based on the information that it extracts from the document. No rules appear in the **Rules** tab.

This type of categorizer works best with categories that are not closely related. For example, if you create categories for *Business* and *Stocks*, the rule-based categorizer is the optimal choice. On the other hand, if you create *Business* and *Sports* categories the statistical categorizer works well.

#### Rule-based categorizer

enables you to specify the rules that determine each category and subcategory. Exercise maximum control over the rules that define your categories with this categorizer. Use the automatic rule generator tool to generate a list of category rule terms that you can edit and use with the rule-based categorizer.

### 4.5.2 Using the Automatic Rule Generator Tool

The automatic rule generator tool provides an intermediate step between the statistical categorizer and the rule-based categorizer. The tool defines automatically created, but humanly editable, linguistic rules. Use automatic rule generation as the first step when you write your linguistic rules. Import these rules into your **Rules** tab and edit them for precision.

---

## 4.6 Optimizing Precision

### 4.6.1 About Precision

The *precision* of a categorizer is measured by the percentage of documents that the categorizer correctly assigns to a given category.

#### Precision and Build Time

##### Rule-based categorizer

exercise maximum control when you write rules for each category, individually. Write rules based on the strings of the unique identifier terms for the linguistic rule-based categorizer. Add Boolean operators to define Boolean rules. For more information, see Chapter 8: *Rule-Based Categorizers*, Chapter 10: *Rule-Based Categorizer: Linguistic Terms*, and Chapter 11: *Rule-Based Categorizer: Boolean Terms*.

##### Automatic rule generator tool

import the rules that are automatically created by this tool and edit them for precision in the Rules window. For more information, see Chapter 7: *Automatic Rule Generator Tool*.

##### Statistical categorizer

use automated processes and a training set of documents with this categorizer. Use this categorizer to develop a taxonomy of categories that do not require precise matching. For more information, see Chapter 6: *Using the Statistical Categorizer*.

---

**Note:** The build time is less for the automatic rule generator tool and the statistical categorizer. These automated processes rely on training sets of documents. However, faster build time comes at the expense of precision and recall.

---

### 4.6.2 About Precision and Recall

Recall is a measure of the categorizer's ability to correctly assign all of the relevant documents to a matching category.

---

Precision and recall are also determined during the following project development stages:

- Take these analytics into consideration when you define individual categories and the overall taxonomy.
- Select the appropriate type of categorizer for the level of precision and recall that you want to obtain.
- Optimize the configuration settings for your categorizer to maximize precision and recall.
- The categorizer is also tested against a small collection of known documents (testing set). The accuracy of the categorizer is measured by the number of matches that the categorizer makes from the total number of test documents in a given category.
- After testing the categorizer, the rules for the selected category, or all of the categories in the taxonomy, can be edited and redefined.

## 4.7 About the Testing and Training Documents

There are two sets of documents that are required for a statistical categorizer. The testing set is also necessary for a rule-based categorizer.

### Testing Set

(used for both types of categorizers) a set of ten or more representative documents for each of the categories that comprise the taxonomy structure. These testing documents should be familiar to you. They are used to check the accuracy and precision of the categorizer. In other words, a small set of documents can help you to determine whether your categorizer is operating with the precision and the recall that you require. For example, if a single document is categorized into several categories, your category rules could be too broad. The opposite would be true if a test document is not categorized into any existing categories.

### Training set

(used only for the statistical categorizer and the automatic rule generator tool) consists of individual sets of approximately 20 documents for each category in the taxonomy structure. The training set of documents should

---

contain texts that are similar to, but different from, the testing set that is later used to test the statistical categorizer.

It is important that the training set represents the documents that you plan to categorize into the category that they are expected to match. SAS Content Categorization Studio uses the training set to develop the statistical categorizer, to generate subcategories, and to train the automatic rule generator tool. The statistical categorizer automatically analyzes documents based on the frequency of occurrence for the most meaningful terms in these documents. Each document is categorized into a category in the taxonomy structure by the statistical categorizer.

## 4.8 How to Build Categorizers

### 4.8.1 Build a Statistical Categorizer

It is easier and simpler to build a statistical categorizer than to build a rule-based categorizer. This is because the statistical categorizer automatically writes the rules for you.

To automatically build a statistical categorizer from the training set of documents, complete these steps:

1. Plan a taxonomy.
2. Create each category in the taxonomy.
3. Assemble a testing set of documents.
4. Assemble a training set of documents.
5. Use the taxonomy of training documents to automatically build the statistical categorizer.
6. Test the statistical categorizer against the testing set.
7. Revise the training set of documents, if you are not satisfied with the testing results, and repeat these steps.

A training set is a representative collection of documents that reflects the categories and subcategories that you develop. You organize the documents

---

into a directory structure that replicates your taxonomy. The requirements for a testing set are identical to those for a training set. However, the assembled documents for a given category are different in each of the two sets. This requirement ensures the accuracy of the rules.

For more information about the statistical categorizer, see Chapter 6.

### 4.8.2 Build the Automatic Rule Generator Tool

Categorization with automatically generated rules provides a mixture of automation and manual implementation. It consists of the following basic steps:

1. Plan a taxonomy for your project.
2. Create each category.
3. Assemble a training set of documents.
4. Use the training set to automatically generate linguistic rules.
5. Decide whether to export your rules to use as the basis for the rule-based categorizer.



---

## 4.8.3 Building a Rule-Based Categorizer

### 4.8.3.A Build a Rule-Based Categorizer

You can build a rule-based categorizer by explicitly tailoring a set of rules for each category in the taxonomy. This method enables you to achieve a higher degree of precision. The highest level of precision is available when you write Boolean rules.

To build a rule-based categorizer, complete these steps:

1. Plan a taxonomy for the project.
2. Define each category.
3. Write your rules.
4. Assemble a testing set of documents.
5. Test the rule-based categorizer.
6. Repeat this process as needed.

When you build a rule-based categorizer, you have the flexibility and control to meet your exact requirements for accuracy. You can use automatically generated rules or hand-written rules. You can also write linguistic rules to quickly develop a taxonomy, or Boolean rules for maximum control over categorization accuracy.

### 4.8.3.B Finding Uniquely Identifying Terms

Consider the terms that are unique to each of the categories that you want to define. Type each list into the Rules window for the selected category. To modify the rule terms, see Section 4.8.3.C *Specifying Rule Types* below.

### 4.8.3.C Specifying Rule Types

The rule-based categorizer enables you to define two types of rules, linguistic and Boolean. Linguistic rules consist of a list of terms used to categorize matching documents. For example, the following linguistic rules could be used to define the category `Government_bonds`:

```
bond price
bond
credit market
```

---

```
federal reserve bank
interest rate
maturity
short-term
treasury bill
```

A document that contains the specified percentage of these terms is categorized under `Government_bonds`. Refine linguistic rules by qualifying these rules with special symbols. For example, append the `@N` suffix to a word to apply noun stemming. Alternatively, reference an already-created concept in the taxonomy within a linguistic rule. You can also create symbolic links to other categories.

Boolean rules, on the other hand, use Boolean operators to achieve greater accuracy in the matching process than is possible using linguistic rules. Boolean rules can also include dependencies that link to concept definitions, symbolic links to other categories, and word expansion capabilities. Boolean rules can also use structured-document fields. For example, Boolean rules can use `title` and `body` fields. These fields specify where a match can occur. To differentiate between the same term used in different contexts, use disambiguation. For example, the following rule requires the words *music* or *piano*, but not *flute*, to appear in the `body` field of matched documents:

```
(AND,(OR, _body:"music", _body:"piano"),
      (NOT, _body:"flute"))
```

---

**Note:** Linguistic rules are automatically converted to Boolean rules, internally, by SAS Content Categorization Studio.

---

---

# Chapter: 5

## Creating Categories

---

- *Overview of Creating Categories*
- *Create a Category*
- *Deleting One or More Categories*
- *Specify a Custom Syntax Checker*
- *Provide Metadata for Categories*
- *Working with the Taxonomy Structure*
- *Evaluating a Referenced Category*
- *Noting an Incomplete Category*

### 5.1 Overview of Creating Categories

This chapter explains how to develop and manage categories whether they are specified with linguistic or Boolean rules. Use this chapter whether you create a new project or work with an existing project.

Use this chapter after you perform the following operations:

1. Load SAS Content Categorization Studio onto your machine.
2. Access a project that is of one of the following types:
  - a new project: For more information, see Section 2.3 *Creating a New Project* on page 101.
  - an existing project: For more information, see Section 2.7 *Specifying Project Settings* on page 130.
  - the sample project that is included with SAS Content Categorization Studio.

- 
3. Set the installation-specific operations: For more information, see Section 2.6 *Set Installation-Specific Operations* on page 113.
  4. Customize your Project Settings: For more information, see Section 2.7 *Specifying Project Settings* on page 130.
  5. Read Chapter 4: *Categorization*.

## 5.2 Create a Category

When you create a category, you add a named node to the taxonomy tree in the **Taxonomy** tab. As you create categories, you can write the rules that define these nodes.

To create a category, complete these steps:

1. In the **Taxonomy** tab, right-click on the **Top** node and select **Add Category** from the menu that appears.



2. An empty category node appears as a child of the parent category with the cursor in the text box. Type the name of the category into this box.



3. Use these two steps reiteratively until you complete your taxonomy. The categories appear in alphabetical order when you contract the taxonomy using one of the following nodes Top, Categorizer, Concepts, the language node, or the project name node.

## 5.3 Deleting One or More Categories

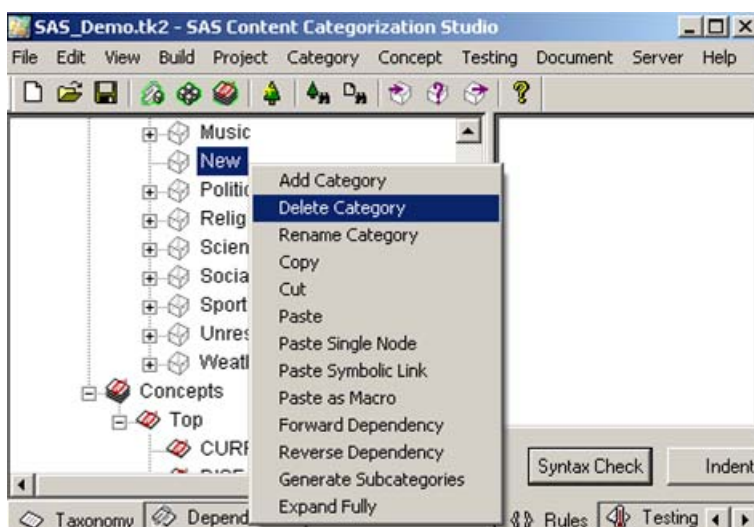
### 5.3.1 Remove One Category

Delete categories that no longer serve their intended purpose. However, you should check to see whether the selected category is a parent with children. If the category does have subcategories, you decide whether to delete the children with their parent or to move them before you delete the parent category. When subcategories exist, they are automatically deleted with their parent category, unless you moved these subcategories.

---

To delete a category, complete these steps:

1. Right-click on the category and select **Delete Category** from the drop-down menu that appears.



---

**Note:** If you use a rule-based categorizer with Boolean rules, you should check the **Dependencies** tab to ensure that there is no dependency on the category that you plan to delete. For more information, see Section 8.9 *Creating Dependencies* on page 230.

---

The SAS Content Categorization Studio confirmation window appears.



2. Click **Yes**.

- 
3. Changes appear in the **Taxonomy** tab after you click either the plus sign (+) or the minus sign (-) to the left of the parent node.

### 5.3.2 Delete Two or More Categories

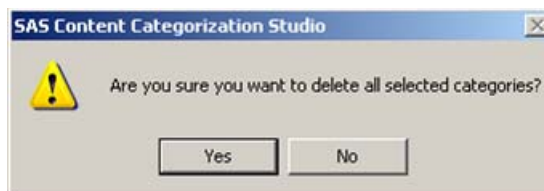
You can remove multiple nodes from the taxonomy at one time.

To delete more than one category, complete these steps:

1. Use the Shift or Ctrl key to select the categories to be deleted.



2. Select **Delete All Selected Categories** in the drop-down menu that appears. A SAS Content Categorization Studio confirmation window appears.



3. Click **Yes**.

---

## 5.4 Specify a Custom Syntax Checker

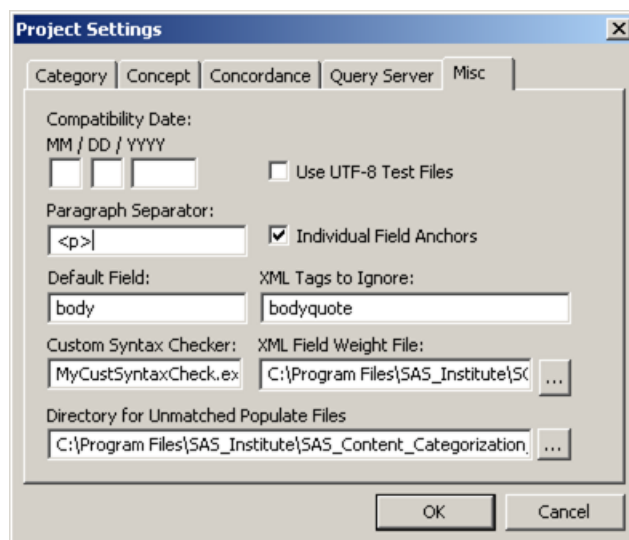
To check the grammar of the category rules in the category branch of the taxonomy, you might choose to use your own syntax checker. (This feature is rarely necessary, but it is available in cases where it is necessary to meet specific requirements.) For example, you could run

`MyCategoryRuleChecker.exe` against the input file `language.directory.xml` file and the output is put into the output file. If the grammar of the category names is OK, the output should contain a single line that says `OK`. If the status of the category name is not OK, the file should state `Error` and detail the status of the project.

After you create the custom syntax checker executable, set the path to the executable in the Project Settings - Misc window.

To set the path to the `.xml` file, complete these steps:

1. Select **Project --> Settings**.
2. The Project Settings window appears. Click the **Misc** tab.



3. Type the path to your custom syntax checker file into the **Custom Syntax Checker Executable** field.
4. Click **OK**.



5. Click **Syntax Check** in the **Rules** tab to launch the syntax check operation.

## 5.5 Provide Metadata for Categories

The **Data** tab makes it possible for you to provide associated information (metadata) with your categories. Use this optional feature when you create, edit, or rename a category.

To specify category information, complete these steps:

1. Select a category in the **Taxonomy** tab.

The screenshot shows the SAS Content Categorization Studio window. On the left, the 'Taxonomy' tab is active, displaying a tree structure under 'English' with 'Categorizer' expanded. 'Football' is selected under 'Top'. The main panel shows the 'Data' tab for the 'Football' category. Fields include: ID (1453677), Author (Ann Martin), Created (February 10, 2010), Modified (empty), Relevancy Cutoff (empty), Relevancy Bias (empty), Category Bias (empty), Match Ratio (0), and status options (Completed, Pending, Test Disabled). The Description field contains 'College Football'. The Thesaurus field is empty. The Query field is empty. The Comments field contains 'Be sure to exclude high school and pro football from this Category'. The Related Links field is empty. The Testing Path and Training Path fields are empty, each with a 'Propagate' button. The Propagate Options section has checkboxes for 'Identical Path' and 'Create Folders'.

2. Enter the identification number, if any, for this category into the **ID** field. For example, type 1453677.

- 
3. Type the name of the creator of this category into the **Author** field. For example, enter Ann Martin.
  4. Enter identifying information for this category into the **Description** field. For example, type College Football.
  5. Type any notes into the **Comments** field. For example, type Be sure to exclude high school and pro football from this category.

---

**Notes:** The **Created** date is automatically entered for you when you define a category.  
When you enter or change the rules, the **Modified** date is also automatically entered, or changed.

---

---

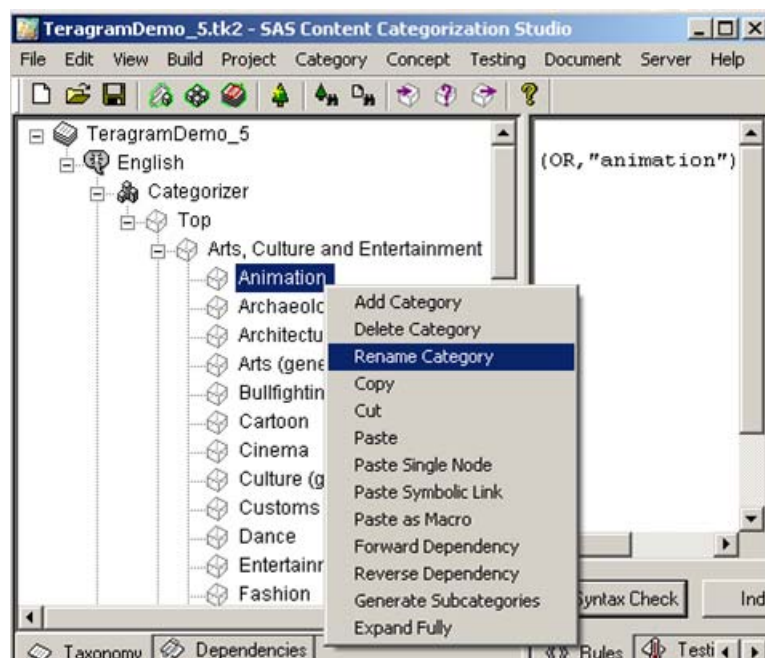
## 5.6 Working with the Taxonomy Structure

### 5.6.1 Rename a Category

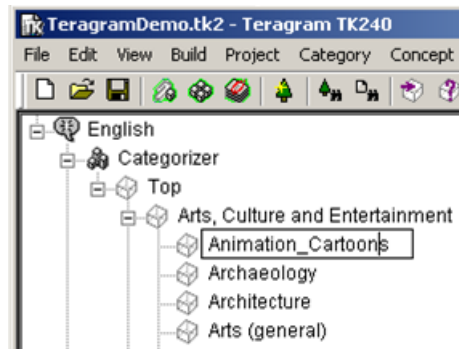
You can rename a category anytime. When you rename a category, rebuild the categorizer.

To rename a category, complete these steps:

1. In the **Taxonomy** tab, select the category to be renamed. For example, select **Animation**.
2. Right-click on the category node and select **Rename Category** from the drop-down menu that appears.



3. Type the new name for the category into the box that appears.



4. (Optional) To see the categories reorganized into an alphabetical list, click the minus (-) sign to the left of the parent category.

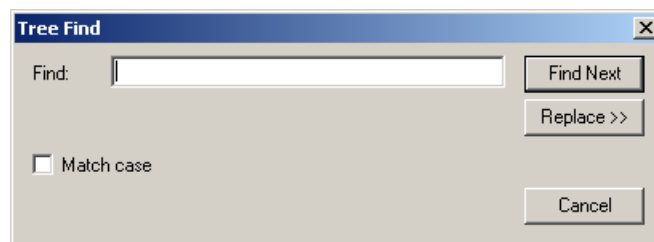
## 5.6.2 Finding and Replacing Category Names

### 5.6.2.A Find Text in the Taxonomy Tree

The **Tree Find** operation locates categories by name. This operation works best with large, hierarchical taxonomy structures where all of the categories in the taxonomy cannot be displayed at one time.

To find a category and change its name, complete these steps:

1. Select **Edit --> Tree Find**. The **Tree Find** window appears.



2. Type the name, or the first letters of the category name that you want to locate, into the **Find** field.
3. Click **Find Next**.

- 
4. (Optional) Click **Replace**. The Tree Replace window appears. Use the Tree Replace window to find and replace category names. For more information, see Section 5.6.2.B *Replace Text in the Taxonomy Tree* on page 167.
  5. (Optional) Select **Match Case** to locate case-sensitive matches.
  6. (Optional) Use Step 3 reiteratively.

The SAS Content Categorization Studio confirmation window appears after the last instance of matching text is located.



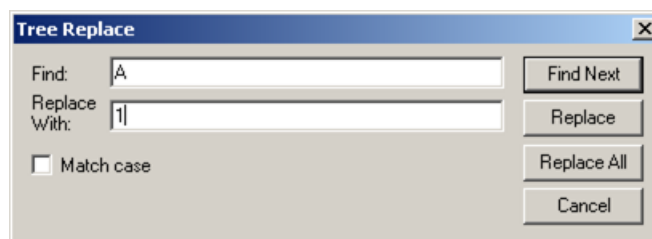
7. Click **OK**.
8. Click **Cancel** in the Tree Find window.

### 5.6.2.B Replace Text in the Taxonomy Tree

This operation works best in large, hierarchical taxonomy structures where all of the categories cannot be displayed at one time. Use the **Tree Replace** operation to replace text in the **Taxonomy** tab.

To access the **Tree Replace** window, complete these steps:

1. Select **Edit --> Tree Replace**. The Tree Replace window appears.



2. To locate a term, type the name, or the first letters of the category name, into the **Find** field.

- 
3. Type the term that should be substituted for each instance of a matched term into the **Replace With** field.
  4. (Optional) Select **Match Case** to find case-sensitive matches.
  5. Click **Find Next**.
  6. Click **Replace**.
  7. (Optional) Click **Replace All** to automatically replace all instances of found text.

---

**Warning:** Use the **Replace All** operation with care. This operation cannot be undone. It is typically used to replace limited numbers of category names in small taxonomy structures where the whole name is identified and replaced.

---

8. (Optional) Use Step 5 and Step 6 reiteratively. The SAS Content Categorization Studio confirmation window appears after the last instance of matching text is located.



9. Click **OK**.
10. Click **Cancel** in the Tree Replace window.

---

## 5.6.3 Creating Categories Using the Copy Operation

### 5.6.3.A Copy and Paste One Category

You can create one or more categories by copying and pasting them into another part of the taxonomy. When you copy categories, unlike moving them, you create a duplicate. For example, you can copy a parent and paste it into the taxonomy as the child of another parent. The copied category is identical to its source category, because it has the same set of rules and the same associated data. Only the full path is different. For example, the original category name might be `Top/Music/Jazz` while the name of the copied category could be `Top/Culture (general)/Jazz`.

After you copy and paste a category, modify its metadata and rule, to make it different.

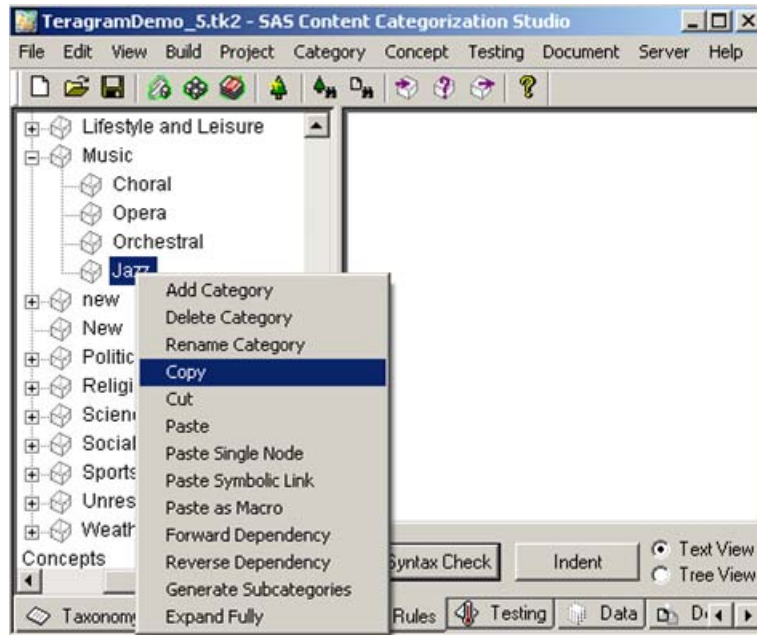
---

**Note:** You cannot copy a category into the concepts branch of the taxonomy tree.

---

To copy and paste one category, complete these steps:

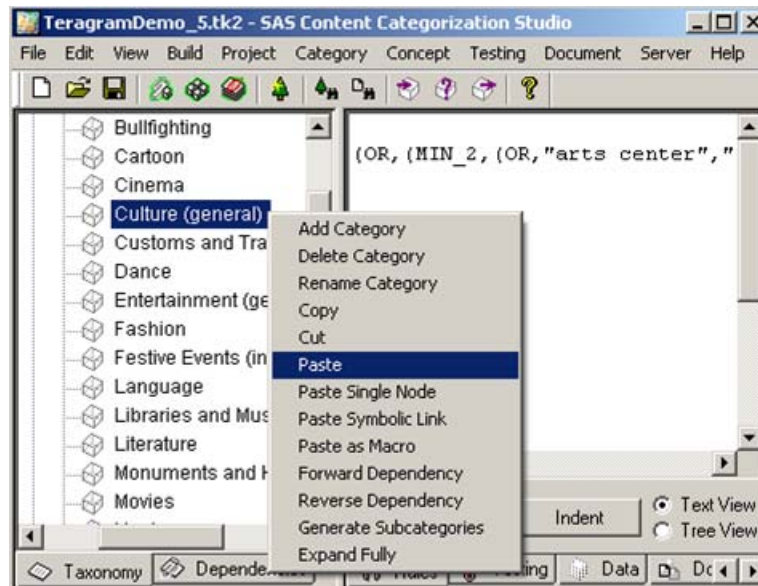
1. Right-click on a category in the **Taxonomy** tab. For example, select Jazz.



2. Select **Copy** from the drop-down list that appears.



3. In the **Taxonomy** tab, right-click on the category that you want to be the parent of the copied category. For example, select *Culture (general)*.



4. Select **Paste** from the drop-down list that appears. The copied category is pasted below the parent category. For example, *Jazz* might be pasted below *Culture (general)*.



---

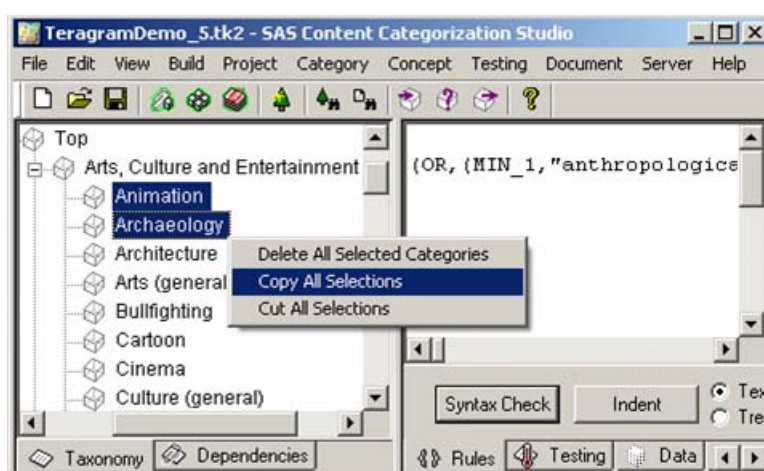
### 5.6.3.B Copy and Paste More than One Category

Create duplicate categories to quickly build a taxonomy with multiple, similar categories. You can then edit your categories to differentiate them. Use the **Copy All Selections** operation to copy and paste two or more categories into the **Taxonomy** tab.

After you copy and paste two or more categories, modify their metadata and rules, to differentiate these categories.

To copy and paste more than one category into the **Taxonomy** tab, complete these steps:

1. Click either the Shift or the Ctrl key, and select two or more categories in the **Taxonomy** tab. For example, select the Animation and Archaeology categories.

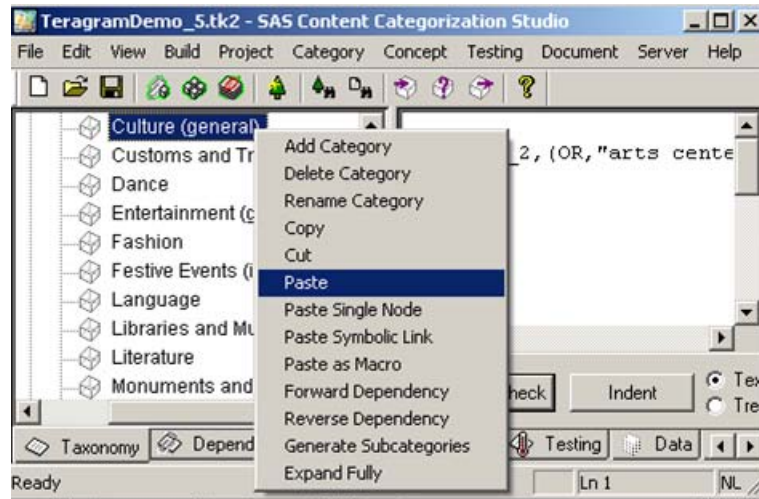


---

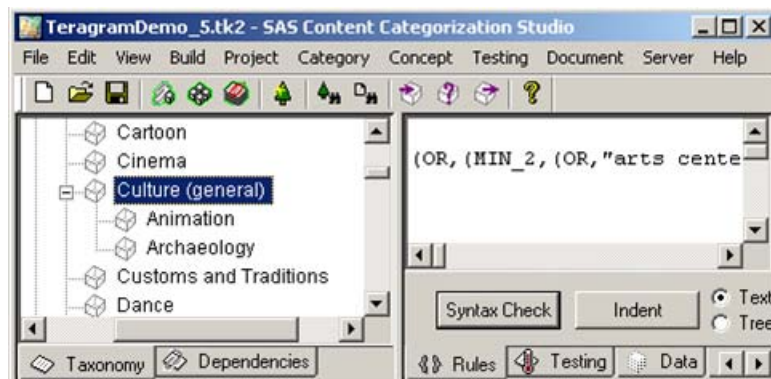
**Hint:** These three drop-down operations only appear when you select more than one category.

---

2. Right-click on another category. For example, select Culture (general).



3. Select **Paste** from the drop-down menu that appears. The two nodes that you copied and pasted appear in the taxonomy structure.



---

## 5.6.4 Moving One or More Categories

### 5.6.4.A Move One Category

The **Cut** and **Paste** commands enable you to delete one or more categories in one part of the category taxonomy tree and paste them to another section.

---

**Note:** Categories cannot be moved to the concepts section of the taxonomy tree.

---

The following three points relate to rule-based categorizers only:

- Boolean categories can become children of linguistic categories, and linguistic categories can become children of Boolean categories.

---

**Hint:** This applies to copying as well as to moving.

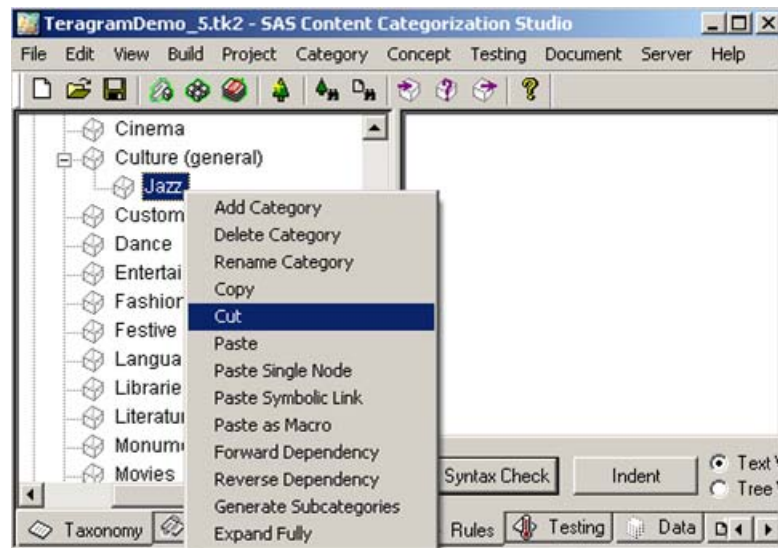
---

- The moved category keeps its own set of rules and metadata.
- If the moved category (for a rule-based categorizer only) is referenced by a macro in another category, the macro is not changed. This means that the macro might point to a non-existent category. For this reason, before you move a category, check the **Dependencies** tab to see these dependencies before you perform this operation. For more information, see Section 11.12 *Dependencies between Categories or Categories and Classifier Concepts* on page 337.

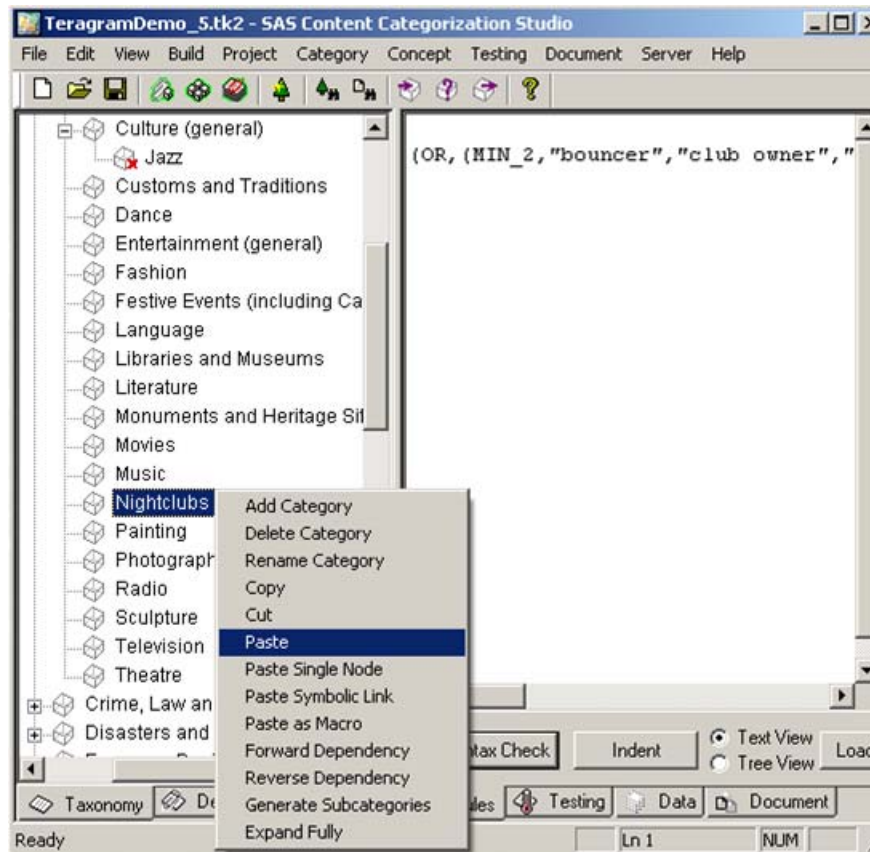
---

To move one category, complete these steps:

1. Right-click on the category that you want to move in the **Taxonomy** tab. For example, select `Jazz`.



2. Select **Cut** from the drop-down menu that appears and a red X appears on the cut category. It stays until the move operation is complete.
3. Select the category that you want to make the *parent* of the category to be moved. If the category is moved to a top level position, select **Top**.
4. Select **Paste** from the drop-down menu that appears.



**Note:** All subcategories of the moved parent categories are also relocated. For example, the child category `Jazz` is relocated.

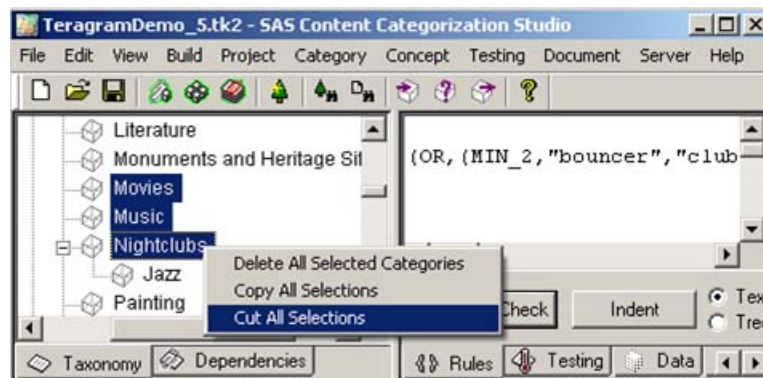
5. The moved category appears in the **Taxonomy** tab below its parent. For example, Jazz appears below Nightclubs.



#### 5.6.4.B Move Two or More Categories

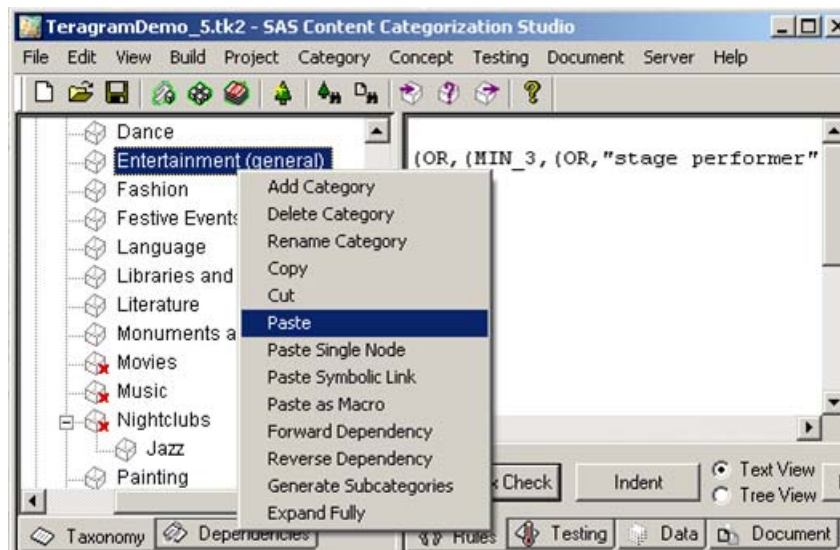
To move more than one category with the **Cut** and **Paste** operations, complete these steps:

1. Press either the Shift or Ctrl key and select two or more categories in the **Taxonomy** tab.





- 
2. Right-click on the highlighted nodes and select the **Cut All Selections** operation that appears in the drop-down menu. The deleted categories are marked with a red X.



---

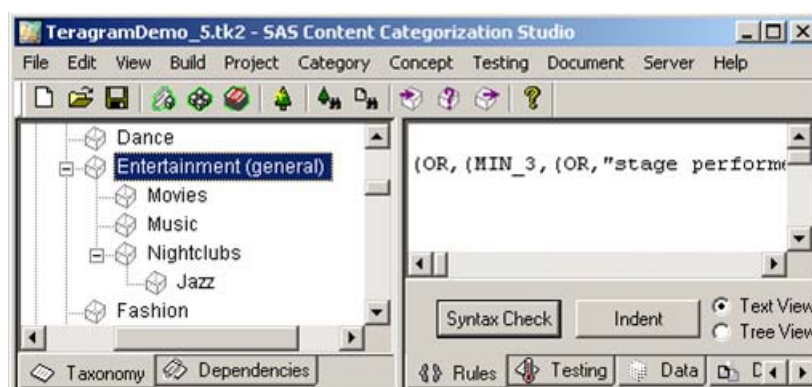
**Note:** Any subcategories of the moved parent categories are also relocated. For example, the child category `Jazz` is relocated.

---

3. Right-click on the category that becomes the parent of the marked categories. For example, right-click on `Entertainment (general)`. Select **Paste** from the drop-down menu that appears.



- 
4. The moved categories appear as children of the selected category.

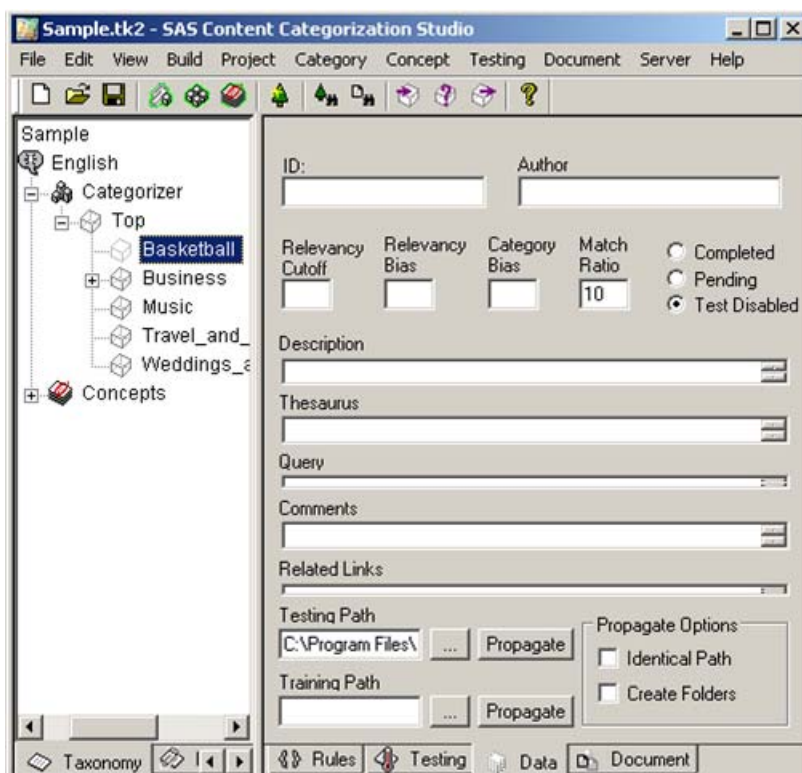


## 5.7 Evaluating a Referenced Category

Select **Test Disabled** in the **Data** tab to evaluate a referenced category. (You can also use this feature with concepts.) When you make this selection, no test results are generated for this node. This operation is often used to define helper categories (or concepts) with common rules that apply to a number of categories. However, these common rules are not exposed to users.

For example, an electronics retailer might define a taxonomy that includes TV brands. Helper categories could be used to define features. In this case, the rules that define the features are hidden, but at work.

When you select this radio button, the node for the selected category appears in lighter gray than the nodes for the other categories in the **Taxonomy** tab. This color change enables you to easily see the categories that are disabled.

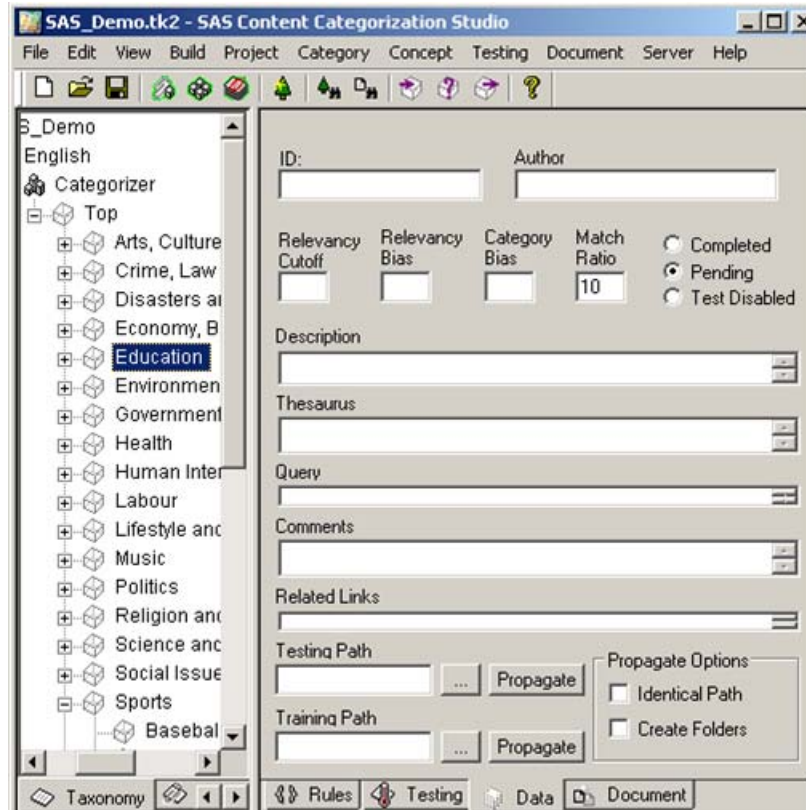


---

## 5.8 Noting an Incomplete Category

Use the **Pending** radio button in the **Data** tab to flag a category that is incomplete for informational purposes, only. For example, use this flag to mark a category where you started to define the metadata, but did not complete this process. The category rule is included in the .mco file.

*Display 5-1 Selected Category*





---

## Chapter: 6

# Using the Statistical Categorizer

---

- *Overview of the Statistical Categorizer*
- *Benefits of the Statistical Categorizer*
- *Determining Category Membership*
- *Quick Start Guide for the Statistical Categorizer*
- *Training the Statistical Categorizer*
- *Building and Saving the Categorizer*
- *Testing the Statistical Categorizer*
- *Revising the Statistical Categorizer*

### 6.1 Overview of the Statistical Categorizer

The statistical categorizer enables you to automatically categorize large numbers of documents into a limited number of broad categories. For example, automate categorization rules for categories such as transportation, books, vacations, and so on. This categorizer works best with taxonomies where the distinctions between the categories are obvious, and not when the categories are closely related, or overlapping. For example, *Stocks* and *mutual funds* provide an example of potentially overlapping categories. For this reason, although you can use this categorizer to define rules for hierarchical taxonomies, the examples shown are only for flat taxonomies.

Unlike the rule-based categorizer where you write rules that define the category, the statistical categorizer automatically defines the rules based on a training set of documents. The training set automatically configures the statistical categorizer to perform a statistical analysis of the uniquely identifying terms in each training document.

The statistical categorizer is trained using the meaningful terms that appear most frequently in the training documents for the category. The statistical

---

categorizer categorizes input documents based on the rules that it automatically developed from these training documents.

## 6.2 Benefits of the Statistical Categorizer

When you choose to use the statistical categorizer, you gain the following benefits:

- easy-to-use setup
- rapid deployment
- automatic rule generation that saves you the time of writing rules

## 6.3 Determining Category Membership

SAS Content Categorization Studio deploys advanced linguistic technologies to automatically generate category rules. These rules are based on a linguistic analysis of all of the documents in the training set. The statistical categorizer excludes any terms that match another rule in order to derive a list of unique identifiers for each category.

The individual category rules are based on the training documents that are provided for both the specified category and for the entire taxonomy. In other words, automatic rule generation takes place in relationship to all of the categories that comprise the taxonomy.

In this type of taxonomy, a document that is a member of one category is excluded from all of the other categories. For this reason, any changes made to the training documents for one category affect other category rules.

It is important to take the following steps before you automatically generate the rules for your taxonomy:

First, create *all* of the categories for your taxonomy structure.

Second, assemble *all* of the documents for the training set. This includes the documents for each of the categories in the taxonomy.

---

**Hints:** The category rules for the statistical categorizer, unlike the rule-based categorizer, are not displayed when you click on the **Rules** tab.  
If you view a tested document in the **Document** tab, the matched terms are not highlighted.

---

## 6.4 Quick Start Guide for the Statistical Categorizer

To build and deploy the statistical categorizer, complete these steps:

1. Create a new project. For more information, see Section 2.3 *Creating a New Project* on page 101.
2. (Optional) Reset the following project operations using the Options window. For more information, see Section 2.6 *Set Installation-Specific Operations* on page 113:
  - **Always save before each test**
  - **Always rebuild before each test**
  - **Sort taxonomies automatically**
3. Specify a taxonomy of categories. For more information, see Section 5.2 *Create a Category* on page 158 and Section 5.6 *Working with the Taxonomy Structure* on page 165.
4. Assemble the training set of documents. For more information, see Section 6.5.2 *Assemble a Training Set of Documents* on page 188.
5. Create a directory structure that replicates the taxonomy for the training documents. For more information, see Section 6.5.3 *Set Training Paths to the Training Directory* on page 189.
6. Place the training set of documents into the directory structure. For more information, see Section 6.5.4 *Placing the Training Files into the Training Directory* on page 192.

- 
7. Set the training and testing paths for the categories in the **Data** tab. For more information, see Section 6.7 *Testing the Statistical Categorizer* on page 193.
  8. Build the statistical categorizer. For more information, see Section 6.6.1 *Build the Statistical Categorizer* on page 192.
  9. Test the statistical categorizer. For more information, see Section 6.7 *Testing the Statistical Categorizer* on page 193.
  10. Make any necessary revisions. For more information, see Section 6.8 *Revising the Statistical Categorizer* on page 200.



---

## 6.5 Training the Statistical Categorizer

### 6.5.1 Preparing to Train the Categorizer

The statistical categorizer is automatically trained by SAS Content Categorization Studio. This operation uses a training set of documents that saves you the time of developing rules for broadly defined categories.

A *training set of documents* is defined as a group of texts (usually 20) that are ideal matches for the category. For example, if you want to specify *Baseball* as a category, choose documents that contain the unique identifying terms that you expect the rule to match. For example, select articles on *Baseball* that include terms such as baseball, bat, catcher, and so on. Choose these texts instead of general *Sports* documents.

Before you can train the statistical categorizer, create a project and name all of the categories in the taxonomy. For more information, see Chapter 2 and Chapter 5. Unlike the rule-based categorizer, you cannot create and test each category as you add it to the taxonomy. The statistical categorizer uses all of the documents in the training set as it defines each category.

When you want to set up a directory structure for a training set of documents, complete the processes explained in the following sections:

- *Assemble a Training Set of Documents* on page 188
- *Set Training Paths to the Training Directory* on page 189
- *Placing the Training Files into the Training Directory* on page 192

This process is similar to the steps that are used to create a testing set of documents for the rule-based categorizer. The differences between the two processes are explained in Section 6.5.2 *Assemble a Training Set of Documents* below:

---

## 6.5.2 Assemble a Training Set of Documents

The training and testing sets of documents are assembled in similar ways, but their purposes make the composition of each group different. The training set contains documents with the unique terms that should be identified in the testing documents. The testing set of documents is a group of texts that you expect to match the category rule. However, these documents are not selected for the unique identifiers that they contain for the selected category.

The major differences between the training and testing sets of documents are listed below:

1. The training set of documents trains the statistical categorizer to match the key words identified in this group of texts with those in input texts. Accuracy in selecting appropriate documents for each category is critical to building a precise categorizer. For this reason, the emphasis is placed on assembling appropriate representations for each category in the taxonomy structure, and not on assembling a range of documents.

Assemble approximately 20 documents that you are familiar with that are also ideal candidates for each of the categories in your taxonomy. These documents should include the types of texts that you expect the end user to query. For example, include `.html`, `.XML`, `.SGML`, and `.txt` documents. These texts should also be of varying levels of categorization complexity.

2. Some of the testing documents should represent ideal matches for the tested category, and some should not match.
3. The documents in both the testing and training sets for a category are different to ensure valid test results. In other words, no documents can be part of both the training and the testing sets.

---

### 6.5.3 Set Training Paths to the Training Directory

The entire training set of documents is stored in a directory tree structure that mimics the taxonomy. This matching helps to ensure the accuracy of automatic rule generation.

You have two choices when you develop a training directory. You can either create the directory tree manually, or you can enable SAS Content Categorization Studio to develop the directory tree structure.

In either case the name of the root *training* directory is different from the root *testing* directory.

---

**Note:** To return precise results, name both directories differently and place unique sets of texts in each.

---

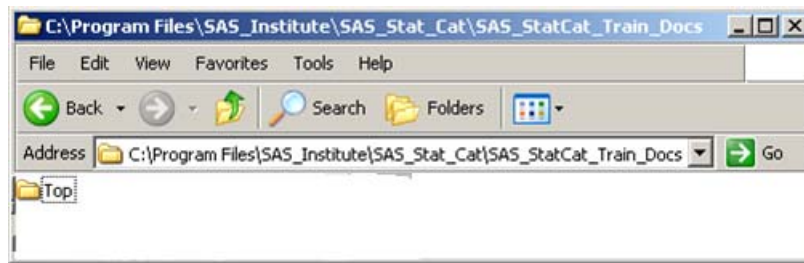
*Display 6-1 Testing and Training Directories.*



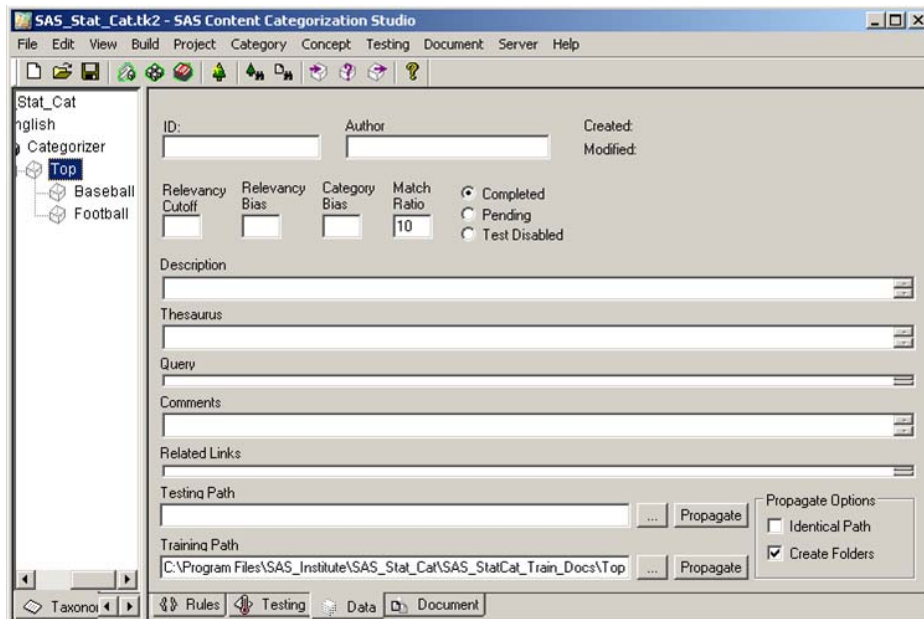
To create the training directory structure, complete these steps:


1. Create a new folder and give the training root directory an appropriate name. For example, create a folder named *SAS\_StatCat\_Train\_Docs*.

2. Inside this folder create a **Top** directory.



3. Click the **Data** tab in the SAS Content Categorization Studio user interface.

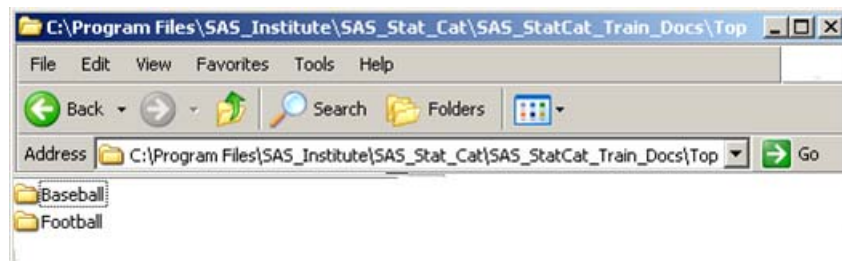


4. Click  to set the **Training Path** to the **Top** folder.
5. Select **Create Folders**.
6. Click **Propagate**.

- 
7. A SAS Content Categorization Studio confirmation window appears.



8. Click **OK**.
9. Move the training files into the training folders. For more information, see Section 6.5.4 *Placing the Training Files into the Training Directory* on page 192.



---

**Note:** If you change the name of one of your categories, also rename the matching training directory. For more information, see Section 5.3 *Deleting One or More Categories* on page 159.

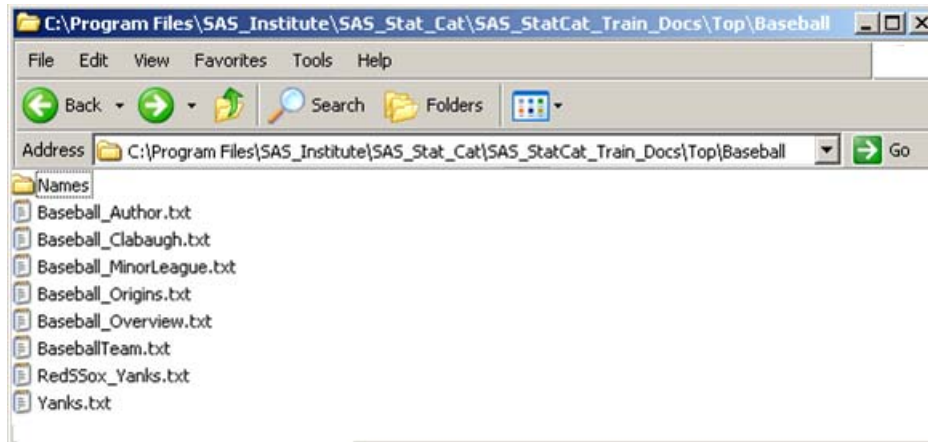
---

---

## 6.5.4 Placing the Training Files into the Training Directory

After you create the directory tree, you can manually paste all of the training documents into these folders.

*Display 6-2 Training Files*



## 6.6 Building and Saving the Categorizer

### 6.6.1 Build the Statistical Categorizer

After you create a taxonomy of training files and set the paths to their folders, build the statistical categorizer (.st.cat file) before you can test its precision. If you make changes to the training documents or the taxonomy, rebuild the statistical categorizer.

This section explains how to manually build the statistical categorizer. (Use these steps to rebuild the statistical categorizer as you make changes to the project.) This section assumes that you did not enable **Always rebuild before each test** in the Options window.

---

To build the statistical categorizer, complete these steps:

1. Double-click on the categorizer node in the **Taxonomy** tab and select **Build --> Build Statistical Categorizer** in the drop-down menu that appears. A SAS Content Categorization Studio confirmation window appears.



2. Click **OK**.

After you build a categorizer, this categorizer becomes the *active categorizer*. Any subsequent testing uses this categorizer by default.

3. (Optional) To confirm that the statistical categorizer is the selected categorizer, select the **Build** menu. You should see a check mark next to **Build Statistical Categorizer**.

## 6.6.2 Saving the Project

To save the changes that you make as you create your project, select **File --> Save**. To automatically save before each test, select **Always save before each test** in the Options window.

## 6.7 Testing the Statistical Categorizer

### 6.7.1 Before You Test the Statistical Categorizer

After you build the statistical categorizer, you can determine its precision and recall by running a full set of tests. The testing process shows whether any of your categories are too broad or too narrow. You can adjust the category rules by re-assembling the training set of documents as necessary.

---

Before you test the statistical categorizer, complete these steps:

1. Define all of the categories in the taxonomy by training your categorizer. For more information, see Section 6.5 *Training the Statistical Categorizer* on page 187.
2. Build and save the taxonomy. For more information, see Section 6.6 *Building and Saving the Categorizer* on page 192.
3. Assemble a testing set of documents and place them in the testing directory by applying the same steps that you used to gather your training set to the testing set. For more information, see Section 6.5 *Training the Statistical Categorizer* on page 187. However, you should select a wider range of testing, than training, documents.
4. Select **Build --> Build Statistical Categorizer**, unless **Always rebuild before each test** is selected in the Options window.

### 6.7.2 Batch Test the Statistical Categorizer

This section provides an overview of the testing steps that are explained in greater detail in Chapter 13 for rule-based categorizers.

To test the statistical categorizer, complete these steps:

1. Select **Build --> Build Statistical Categorizer**.
2. Click the **Testing** tab and select **Test files for this category**.





---

**Hint:** The testing documents are automatically loaded into the **Testing** tab when you set the **Testing Path** in the **Data** tab.

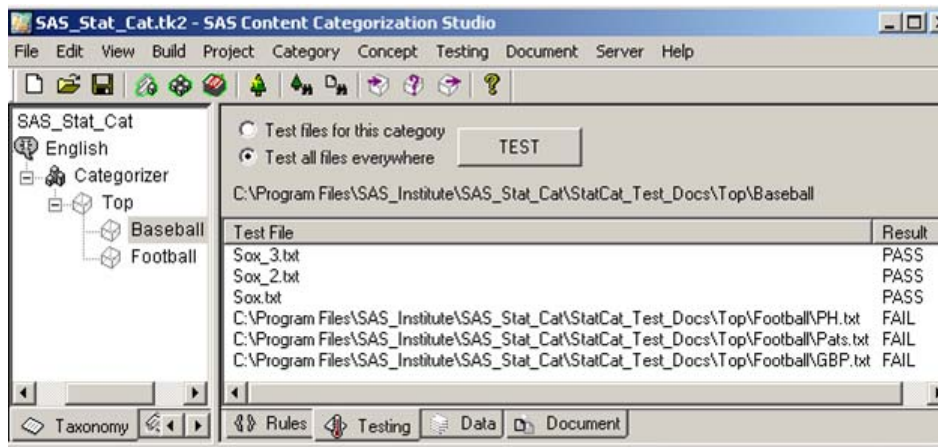
---

3. Click **TEST** in the **Testing** tab and the testing results are displayed in the **Result** column.



4. Use the **Result** column to see the documents that pass and those that fail.

5. To test the selected category against all of the testing documents in this testing directory, select **Test all files everywhere**. The testing results appear in the **Testing** tab. The testing files that are located outside of the testing directory are displayed with their full path.



6. Import failing documents at any point in the process to check the accuracy of your categorizer. Failing test documents are texts that should *not* pass the rule requirements for a category. For example, documents for *Philadelphia Eagles* should not match the *Bald Eagles* category. If you create a folder of failing test documents, you can test the *Philadelphia Eagles* texts to make sure that they do not pass. For more information, see Section 15.3 *Import Failing Documents* on page 402.

### 6.7.3 Test One Document

After you test the testing files that you assembled for one category, you can test each of your testing documents individually against one or all of the categories. Use this operation to see the relevancy score for each of your documents against multiple categories. For more information, see Chapter 14: *Testing with One Document*.

To test one document, complete these steps:

1. Select the category node that you want to test in the **Taxonomy** tab. For example, select **Baseball**.



2. Double-click on one test file in the **Testing** tab to access the selected text into the **Document** tab.
3. The **Document** tab appears to display the PASS or FAIL test results for the tested document.



---

**Note:** The ◀ and ▶ buttons beneath the PASS message do not work with the statistical categorizer. When you use this categorizer, you cannot see the matched terms in the **Document** tab.

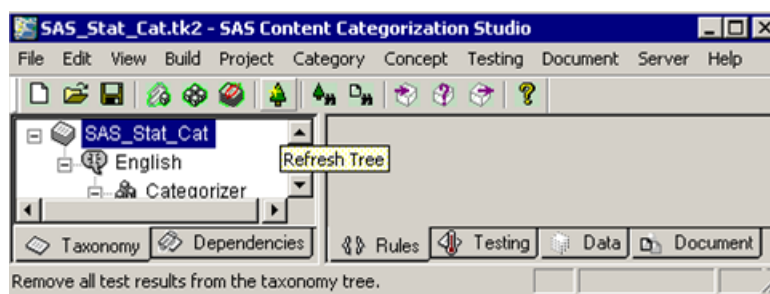
---

4. Select **All categories** in the **Document** tab. PASS and FAIL messages for the results appear in the **Taxonomy** tab for each category and the Best Matches window also appears. (By default, the **Show best matches when testing all** check box is selected in the Options window. If this operation is not selected, choose this operation to see the Best Matches window.)



5. Use the Best Matches window to see the relevancy results for all of the matching categories. For example, the `Baseball` category is listed with a relevancy score of 221.

6. (Optional) Click **Refresh Tree** to remove the testing results.



### 6.7.4 Run a Full Test of the Categorizer

After you test the categorizer, you can run a full test to gain more information about the test results.

To run a full test report, complete these steps:

1. Double-click on the **Categorizer** node in the **Taxonomy** tab and select **Testing --> Full Test Report**.

The Category Test Report window appears displaying the test results for all of the categories.

Path	All Docs	In-Cat	Total	In-Cat %	Neg	N-Tot	Neg %	Prec %	Popul...	Pop Rel
Top	0	0	0	0	0	0	0	0	0	0
Top/Baseball	8	8	8	100	0	0	0	100	0	0
Top/Football	7	7	7	100	0	0	0	100	0	0

For more information about the results displayed in the Category Test Report window, see Section 2.12.12 *The Full Test Report Window* on page 87.

2. (Optional) Click **View as Text** to see this report as a text document in *Notepad*.
3. Click **OK**.

---

## 6.8 Revising the Statistical Categorizer

If you are not satisfied with the test results obtained by the statistical categorizer, use the following operations. (Alternatively, you can choose to use either the automatic rule generator tool or the rule-based categorizer.)

- Revise the training set of documents so that they better represent the categories.
- Rebuild the statistical categorizer (`.st.cat` file) using the revised training set.
- Test the revised statistical categorizer and examine the results for the entire taxonomy.

---

**Note:** A change to a single category affects the way that the statistical categorizer builds the entire taxonomy. Retest the entire taxonomy if you make a change to one or more categories by adding, deleting, or changing any of the training documents.

---

---

# Chapter: 7

## Automatic Rule Generator Tool

---

- *Overview of the Automatic Rule Generator*
- *Benefits of the Automatic Rule Generator Tool*
- *Understanding Category Membership*
- *Quick Start Guide for the Automatic Rule Generator Tool*
- *Automatically Generating Rules*
- *Exporting Rules*
- *Clearing the Automatically Generated Rules*

### 7.1 Overview of the Automatic Rule Generator

The automatic rule generator tool is an optional software solution that expedites the category rule writing process. This tool develops a set of editable linguistic rules from a training set of documents. The selected advanced Linguistic technologies are automatically deployed to identify a set of unique terms to identify each category in the taxonomy from these training documents. For this reason, this tool works like the statistical categorizer. The automatic rule generator tool automatically develops rules based on a training set of documents that you supply.

As a tool, this solution provides an intermediate working rule for categories that maximizes the benefits of both the statistical and rule-based categorizers. For example, the automatic rule generator tool returns the most meaningful terms using statistical analysis. You can edit these terms in order to refine your rules using the rule-based categorizer. This tool is not designed to be used as a categorizer, but only as an intermediate step in the categorization process.

---

Use the automatic rule generator tool for the following reasons:

- Quickly and automatically generate a list of linguistic terms based on an analysis of the training documents that you provide for each category in the taxonomy.
- Develop automatic rules that can be easily exported for broadly defined categories such as business and basketball. Precisely differentiated categories such as *lakes* and *ponds* are not good candidates for this tool.
- When you are unsure of how to develop linguistic rules for a particular taxonomy, use this tool to generate unique terms for each category. You can then edit these rules.

After the automatic rules are generated by the categories, you can choose whether to export each rule into the **Rules** tab. Use the **Rules** tab to select and edit the terms that form the basis of a category rule used by the rule-based categorizer.

In summation, the automatic rule generator tool facilitates the process of writing rules for the rule-based categorizer. Use this tool to have SAS Content Categorization Studio automatically develop a set of initial category-defining terms.

## 7.2 Benefits of the Automatic Rule Generator Tool

The automatic rule generator tool offers you several benefits including the following:

### Easy-to-use set up

Similar to the statistical categorizer, the automatic rule generator tool uses a training set of documents that has a taxonomy structure that imitates the structure of the category taxonomy. This tool makes it easy to develop a rule that defines each category.

### Rapid deployment



---

Unlike the statistical categorizer, the automatic rule generator tool enables the linguistic rule-based categorizer to be deployed almost immediately after the rules are exported to the **Rules** tab.

A greater degree of precision

Use this tool as an initial step in the process of building a rule-based categorizer. You can modify these rules for greater precision.

## 7.3 Understanding Category Membership

SAS Content Categorization Studio uses the training set of documents to automatically generate a set of linguistic terms for all of the categories in the taxonomy. These sets of terms can form linguistic rules that identify category members from input documents.

Any changes that are made to the training documents might affect all of the other categories in the taxonomy. The automatic rule generator tool considers all of the terms from all of the training documents before selecting the unique terms for each category.

You do not build a categorizer using this tool. Export the terms into the **Rules** tab where you can edit them to work with the rule-based categorizer.

## 7.4 Quick Start Guide for the Automatic Rule Generator Tool

To generate rules for your categories using the automatic rule generator tool, follow the steps below that reference relevant, earlier sections of this manual. This process assumes that you have created a project, built a taxonomy, and made selections in the Options and Project Settings windows.

1. Create a directory of training documents for the automatic rule generator tool using the following four steps. For more information, see Section 6.5 *Training the Statistical Categorizer* on page 187.

- 
- a. Assemble approximately 20 documents that you are familiar with and that you consider to be ideal candidates for each of the categories in your taxonomy. These documents should include the types of texts that you expect the end user to query such as .html, .xml, .sgml, .txt. These texts should also have varying levels of categorization complexity.
  - b. Create a directory structure to hold the training set of documents that is identical to the taxonomy structure. Set the **Training Path** in the **Data** tab to this directory. For more information, see Section 6.5.3 *Set Training Paths to the Training Directory* on page 189.
  - c. Import your training files into the directory structure that matches your taxonomy. For more information, see Section 6.5.4 *Placing the Training Files into the Training Directory* on page 192.

---

**Note:** Effective use of the automatic rule generator tool requires that you assemble a training set of documents for *each* of the categories in the taxonomy. If a training directory is not defined and populated for each category, inaccurate rules are developed.

---

2. Generate automatic rules. For more information, see Section 7.5.2 *Automatically Generate Rules* on page 205.
3. Export the automatically generated rules. For more information, see Section 7.6 *Exporting Rules* on page 206.
4. Make any necessary edits. For more information, see Section 8.5.2 *Write Rules* on page 222.
5. Test the rules. For more information, see *Part 2: Testing*.

---

## 7.5 Automatically Generating Rules

### 7.5.1 Prepare to Automatically Generate Rules

The automatic rule generator tool, like the statistical categorizer, automatically creates a set of rules based on the taxonomy of training documents that you define. For more information, see Section 6.5.3 *Set Training Paths to the Training Directory* on page 189.

After you assemble the training documents into a directory structure and set the paths to this taxonomy, generate your rules. For more information, see Section 7.5.2 *Automatically Generate Rules* below.

### 7.5.2 Automatically Generate Rules

The automatic rule generator tool generates rules, or lists of identifying terms for each category, based on the taxonomy of training documents that you develop.

To generate automatically defined category rules, complete these steps:

1. Select **Category --> Generate Rules Automatically**.
2. The **Automatic Rule** tab appears to the right of the **Document** tab. Select the **Document** tab to see the terms that define this rule.



3. Review the terms that comprise the rules. If you are satisfied with the selected terms, you can export all of the rules into the

---

appropriate **Rules** tab for each category. If not, consider retraining the categorizer or exporting rules individually. For more information, see Section 7.6 *Exporting Rules* below.

---

**Note:** The **Automatic Rule** tab is read-only. You cannot edit the automatically generated rules without exporting them into the **Rules** tab.

---

## 7.6 Exporting Rules

### 7.6.1 Determining When and How to Export Rules

To edit automatically created rules, you export them from the **Automatic Rules** tab into the **Rules** tab. Perform this operation for two purposes:

First, export unique, identifying category terms to make rule-writing easier. The automatic rule generator tool extracts a list of significant terms for each category from the training set of documents that you provide. After you export the rule terms, edit them in the **Rules** tab.

Second, you should use the export and edit operations instead of making changes to the training set of documents. If you make changes to the training set, these changes might affect all of the rules in the taxonomy. When you edit each automatically created rule, only the affected category is changed.

There are two different operations that you can use to perform the export process:

First, select **Category --> Export All Generated Rules** to send all of the rules for all of the categories at one time. This is a timesaving operation to use the first time you generate automatic rules.

---

**Hint:** **Export All Generated Rules** is available only after you generate automatic rules.

---

---

Second, click **Export** in the **Automatic Rule** tab to move the linguistic terms from this tab into the **Rules** tab. When you use the **Export** button, repeat this process for every category. Use the export operation in cases where you do not want to overwrite any existing rules.

## 7.6.2 Option 1: Export All Generated Rules

Use this operation the first time you automatically generate rules for a new project. This saves you the time and labor of performing the export process category-by-category. For more information, see Section 7.6.3 *Option 2: Export the Generated Rules for One Category* on page 208.

---

**Note:** If you write some category rules, and you decide to export all of the rules at one time, the export operation deletes these rules. These rules are replaced with automatically generated linguistic rules.

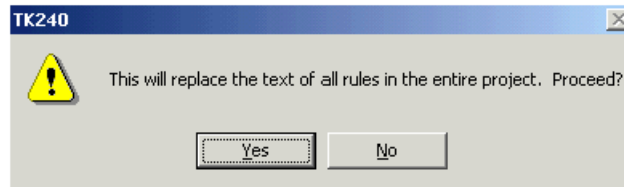
---

To export all of the generated rules, complete these steps:

1. Select any category in the **Taxonomy** tab and click the **Automatic Rule** tab. The automatically generated rules for the selected category are displayed.



- 
2. Select **Category --> Export All Generated Rules**. A SAS Content Categorization Studio confirmation window appears.



3. Click **Yes**.
4. Select a category in the **Taxonomy** tab and select the **Rules** tab to display the new rule.



5. To edit these terms, see Section 8.5.2 *Write Rules* on page 222.

### 7.6.3 Option 2: Export the Generated Rules for One Category

You should export one rule at a time when you want to rewrite or edit some, but not all, of your existing category rules. This operation enables you to preserve some of your category rules.

To export the generated rules for one category, complete these steps:

1. Select a category in the **Taxonomy** tab and access the **Automatic Rule** tab.



2. Click **Export**. A SAS Content Categorization Studio confirmation window appears.



3. Click **Yes**.
4. Select the **Rules** tab where you edit these terms. For more information, see Section 8.5.2 *Write Rules* on page 222.



---

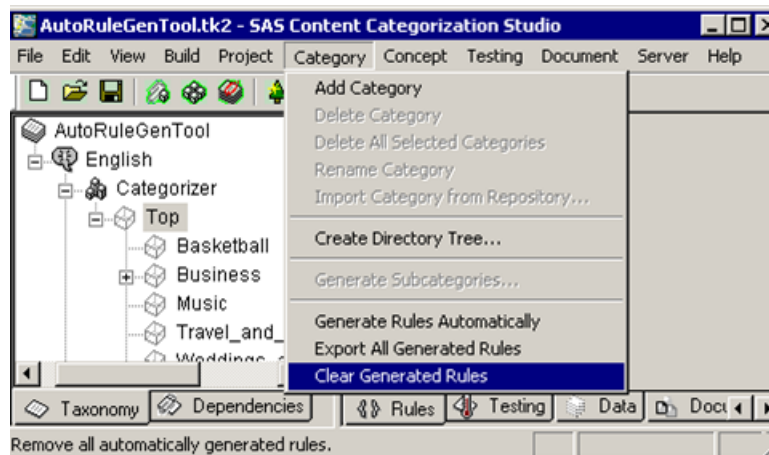
**Note:** You can test the imported rules using the test processes for the rule-based categorizers. For more information, see *Part 2: Testing*.

---

## 7.7 Clearing the Automatically Generated Rules

Select **Category --> Clear Generated Rules** to remove the generated rules from the project. However, this operation is not available in the **Category** menu unless you used the **Generate Rules Automatically** operation.

*Display 7-1 Clear Generated Rules Operation*



When you select the **Clear Generated Rules** operation, the Automatic Rule tab disappears from the lower right side of the user interface. This means that any automatically generated rules that have not been exported are lost. For this reason, review your category rules before you perform this operation.



---

# Chapter: 8

## Rule-Based Categorizers

---

- *Overview of Rule-Based Categorizers*
- *Benefits and Features*
- *Quick Start Guide*
- *Preparing to Write Your Rules*
- *Developing Category Rules*
- *Check the Syntax of a Boolean Rule*
- *Differentiating Symbolic Links from Dependencies*
- *Create Symbolic Links*
- *Creating Dependencies*
- *Building the Rule-Based Categorizer*
- *Automatically Save the Changes*

### 8.1 Overview of Rule-Based Categorizers

Humanly defined rules, based on linguistic terms, capture the unique identifiers that are found in the documents that match each category. These words are similar to the list of terms that are automatically derived from training sets of documents by the statistical categorizer or automatic rule generator tool. However, humanly defined rules can be more precise than automatically generated rules.

Humanly defined rules used by the rule-based categorizers provide more precision and recall than their automated counterparts. The rule-based categorizers use rules that you, or other subject matter experts write, to precisely define category membership for some documents, while excluding texts that contain inappropriate content. These linguistic terms can be modified by special symbols, Boolean operators, and other modifiers.

---

There are two basic types of rule-based categorizers. Each of these categorizers is specified by a different type of rule that can also be qualified:

#### Rule-based categorizer using linguistic rules

Linguistic rules provide an intermediate step between the complexity of category rules that are automatically defined by the automatic rule generator tool and Boolean rules. Boolean rules provide the greatest precision. For more information, see Chapter 10: *Rule-Based Categorizer: Linguistic Terms*.

You can define linguistic rules when you qualify automatically generated rules with special symbols, weights, and other values. You can also write a list of terms that defines each category. Add qualifiers, if you choose.

There are two types of linguistic rules:

##### Unqualified linguistic rules

A category rule is defined by a list of linguistic terms with no modifiers. This rule is similar to the list of terms generated by the automatic rule generator tool. For more information, see Chapter 7: *Automatic Rule Generator Tool*.

##### Qualified linguistic rules

This list of linguistic terms is modified by special symbols, weights, and other values. Use these values to determine the best matches for your categories.

#### Rule-based categorizer using Boolean rules

The Boolean rule-based categorizer is the optimal solution. This categorizer uses rules that include unique, identifying words that are modified by Boolean terms to precisely define category membership. For more information, see Chapter 11: *Rule-Based Categorizer: Boolean Terms*.

There are two types of Boolean rules:

##### Boolean rules

Linguistic terms are modified by Boolean operators. These operators can determine matches based on location and the presence, or absence, or specific terms.

##### Qualified Boolean rules

---

Boolean rules are modified using special symbols, structured-text field names, or other modifiers.

Although you can define linguistic or Boolean rules and use categories defined by each in the same taxonomy, you cannot mix these two rule types to define one category. You can create category rules that use linguistic rules for one or more categories and category rules that use Boolean rules for the other categories in the same taxonomy.

Use the following overview of the processes involved in building a taxonomy of categories:

1. **Build your taxonomy one category at a time:** Unlike the statistical categorizer and the automatic rule generator tool, it is not necessary to complete your taxonomy before you write category rules. When you use the rule-based categorizer you can define your categories one at a time. This process enables you to develop deep and narrow rules for each category.
2. **Test each category as it is built and before you add a new category:** You can test each category individually as you build your taxonomy. Gain an in-depth view of the testing results and identify challenges early in the building process.

---

**Note:** SAS Content Categorization Studio analyzes each category rule in the context of the entire taxonomy to return the best match.

---

3. **Redefine your rules as necessary:** If the testing results do not produce the expected returns, you can repeat the process. Redefine your category rules, rebuild the categorizer, and test until you obtain the results that you require.

---

## 8.2 Benefits and Features

When you choose to use the rule-based categorizer you gain the benefits of:

Humanly created rules

Write your own rules to obtain the precision that you require.

Narrow category definitions

Define narrow category rules for the purposes of minimizing duplicate category membership and to gain a greater degree of recall. For example, define categories that differentiate between *Adult Education* and *Preschool Education*.

Precision

Control the precision, or the ability of the rule-based categorizer to correctly assign documents to each category, when you determine category membership using hand-written rules. For example, choose to correctly categorize documents that contain the word *train* into either *Transportation* or *Body-building* categories.

Relevancy types for both linguistic and Boolean rules

Use a combination of selections set in the Project Settings - Category window and the **Data** tab to determine relevancy. Rule modifiers, and the available settings, can also affect relevancy.

Server Query operation for Boolean rules

Use a Boolean category rule to query an index with the **Server Query** operation. This operation is available for Boolean rules only. If it is used with linguistic rules, these rules are automatically converted to Boolean rules.

Use the overview provided below to gain a comparative overview of the components that are available for category rules:

Table 8-1: Alphabetical Listing of Category Features

Rule-Based Categorizer Feature	Linguistic Terms	Boolean Rule	Both
Boolean Expressions		X	
Boolean Morphological Expansion		X	
Default Category Bias	X		
Category Bias (Data tab)			X
Dependencies			X
Default Category Bias (Category tab)			X
Default Relevancy Cutoff (Category tab)			X
Expand Forms (Rules tab)		X	
Frequency-Based Ranking	X		
Operator-Based Ranking			X
Zone-Based Ranking			X
Indent (Rules tab)		X	
Load Text (Rules tab)			X
Match Ratio (Data tab)	X		
Relevancy Bias (Category tab)	X		
Relevancy Cutoff (Data tab)			X
Relevancy Type (Category tab)			X
Relevancy Cutoff (Category tab)			X
Server Query operation		X	
Special symbols			X
Stemming (word form expansion)			X
Structured text fields		X	

Table 8-1: Alphabetical Listing of Category Features (Continued)

Rule-Based Categorizer Feature	Linguistic Terms	Boolean Rule	Both
Symbolic Links			X
Syntax Check (Rules tab)			X
Text View mode (Rules tab)			X
Tree View mode (Rules tab)		X	

## 8.3 Quick Start Guide

To build and deploy the rule-based categorizer, complete these steps:

1. Create a new project. For more information, see Section 2.3 *Creating a New Project* on page 101.
2. Specify the settings that apply not only to this project, but to all projects created with this installation. For more information, see Section 2.6 *Set Installation-Specific Operations* on page 113.
3. Select project settings. For more information, see Section 2.7 *Specifying Project Settings* on page 130.
4. Add new categories. For more information, see Section 2.3 *Creating a New Project* on page 101.
5. Assemble a set of testing documents into a directory structure that mimics your taxonomy. For more information, see Chapter 12: *Assembling Testing Sets*.
6. Test your rules as you build them. For more information, see Chapter 13: *Batch Testing*, Chapter 14: *Testing with One Document*, and Chapter 15: *Other Testing Operations*.

---

## 8.4 Preparing to Write Your Rules

### 8.4.1 Understanding Rules and Category Membership

Category membership for a rule-based categorizer works in many of the same ways that category membership for the statistical categorizer and the automatic rule generator tool works. The categorizer that you develop should match documents that meet the membership criteria for one category while it excludes texts that meet the criteria for other categories.

Linguistic and Boolean rules use different sets of parameters to define category rules. This remains true even when the linguistic rules are internally converted to Boolean rules by the application.

The following matching features are specific to linguistic rules:

#### Match ratio

This is the percentage of matching terms that are necessary to locate in an input document in order to return a match. The default setting is 10% in the **Match Ratio** field of the **Data** tab. You can reset this specification for each individual category in the Data window. Expand category membership by lowering the match ratio, or limit category membership by increasing this number. (This setting is also used internally by SAS Content Categorization Studio to convert linguistic rules to Boolean rules.)

#### Special symbols

These symbols override the match ratio setting and determine the matching documents. For more information, see Table 10-2 on page 278.

The following matching features are specific to Boolean rules:

#### Boolean operators

Use Boolean operators to precisely define how matched terms appear in an input document. For example, the `DIST` operator specifies the distance between matched terms. If matched terms exceed this distance, no match is returned.

#### Structured Text fields

---

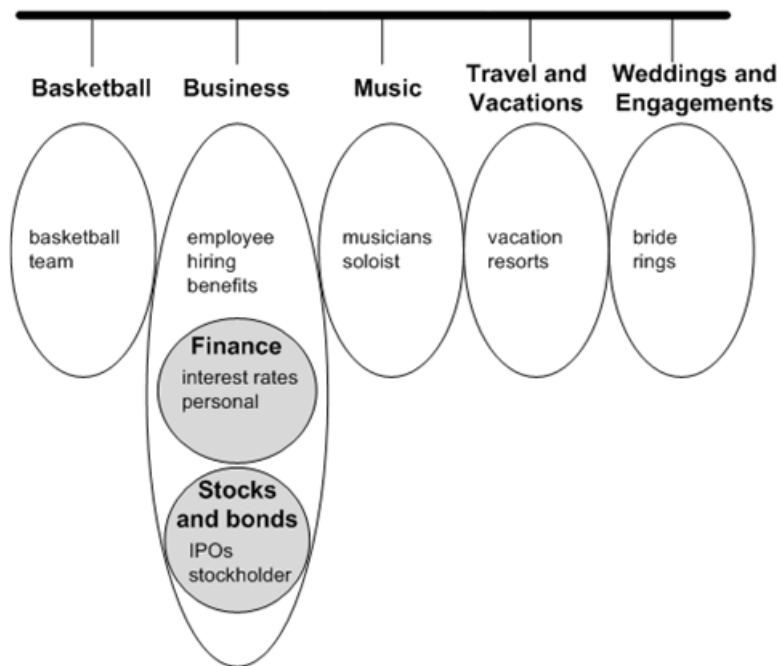
Limit the location of matched terms to specific fields in an XML document.

When documents are matched to categories, SAS Content Categorization Studio analyzes, and if necessary applies, the rules for *all* of the categories in the taxonomy. For this reason, it is important to consider the taxonomy in its entirety when you define a rule for one category.

### 8.4.2 An Example of Category Rules

A sample taxonomy with some linguistic category rules is displayed below. Underneath the category and subcategory names, white ovals with sample rules appear. The gray circles display a list of linguistic terms that define child rules.

*Figure 8-1 Sample Taxonomy*





---

None of the identifier terms for any of the categories are the same. This includes children. For example, the subcategories *Finance* and *Stocks and bonds* do not share any terms with each other or with their parent category *Business*. If two or more categories shared identifier terms, a document that contained these terms might be categorized into each of these categories. For this reason, you might want to qualify your linguistic terms or write Boolean rules. To share identifier terms, consider creating a dependency or a symbolic link. For more information, see Section 8.8 *Create Symbolic Links* on page 226 and Section 8.9 *Creating Dependencies* on page 230.

The goal when writing rules is to create a list of terms that are unique to the documents that are categorized into the selected category. Although numerous terms can identify membership in one category, many of these terms might also be used to identify membership in another category. For example, see *Stocks and bonds*. For this reason the word *money* is not specified in the example shown above.

There is no specific number of linguistic terms that should form the basis of category membership. For some categories, one or two terms might be sufficient. For example, the term *HINI* could be sufficient for a category of the same name. For other categories, it is advisable to create a list of 20 or more terms that define category membership. For example, a list of U.S. states would require 50 entries. When you write category rules that use a large number of linguistic terms, consider the effects of the match ratio setting on the number of terms to be matched.

## 8.5 Developing Category Rules

### 8.5.1 Select a Rule Writing Operation

These are the ways to develop category rules:

Use the automatic rule generator tool

After you define all of your categories, use the automatic rule generator tool to develop a list of rules for each category. For more information, see Section 7.5.2 *Automatically Generate Rules* on page 205.

Hand-write the rules for each category

---

When you write your rules, you have several choices:

First, type your rule syntax into the **Rules** tab.

*Display 8-1 Export an Automatic Rule.*



Second, click **Load Text** in the **Document** tab:

To use this import operation, complete these steps:

1. Click **Load Text** in the **Rules** tab to import the rules that you wrote in another document. For example, you can import rules from *Notepad* into the **Rules** tab.

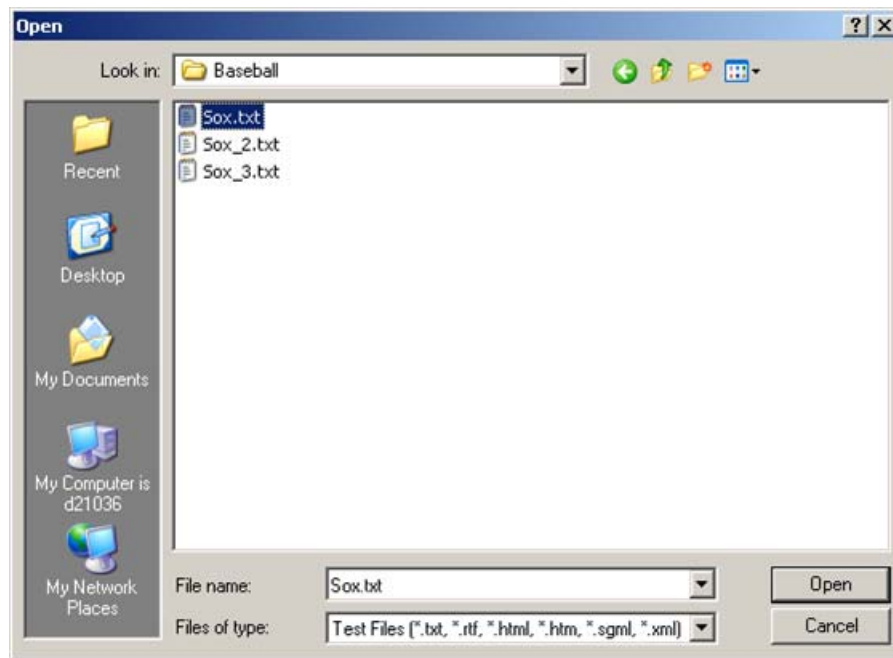


---

**Note:** The **Load Text** button is available only in the **Text View** mode of the **Rules** tab.

---

The Open window appears.



2. Choose the file that specifies the rules for this category. For example, select Sox.txt.
3. Click **Open** and the new rule is loaded into the **Rules** tab.



- 
4. Edit the automatically created linguistic rules by hand. For more information, see and modify the information in Section 8.5.2 *Write Rules* on page 222.
  5. (Optional) Write Boolean rules. For more information, see Chapter 11: *Rule-Based Categorizer: Boolean Terms*.

Combine the hand-written and the automatic rule building features of .SAS Content Categorization Studio

Write a Boolean rule for a parent category from the automatically generated rules of its child categories when you click **Create Rule Text from Children**. You perform this operation whether these rules are linguistic, Boolean, or both. The Boolean rule that is automatically created uses OR operators to join individual rules. For more information, see Section 11.10 *Automating Parent and Child Rule Generation* on page 333.

## 8.5.2 Write Rules

The **Rules** tab enables you to write your linguistic or Boolean rules in a blank window. You can use the same text editing commands in the **Rules** tab that you use when you develop a list in a word processing program. The linguistic rules that you write in the **Rules** tab are also similar in appearance and content to the linguistic rules generated by the automatic rule generator tool. For more information, see Section 7.5.2 *Automatically Generate Rules* on page 205.

To write, or edit, a category rule, complete these steps:

1. Select a category.
2. Click the **Rules** tab.
3. Identify the unique terms that identify category members. You can choose to use the set of testing documents that you assembled for each category, when it is created, in order to identify these terms.
4. Place your cursor in the **Rules** tab and type a list of the words that uniquely define your category. This is a linguistic rule. One way to write a Boolean rule is to add Boolean operators to modify these unique identifiers. For more information, see Chapter 10: *Rule-Based Categorizer: Linguistic Terms* or Chapter 11: *Rule-Based Categorizer: Boolean Terms*.

---

Use the **Text View** mode to write both linguistic and Boolean rules. The **Tree View** mode works with Boolean rules, only. For more information, see Section 11.9.1 *Edit Rules in the Tree View Mode* on page 322.

The list of unique identifiers could look similar to the example shown below if you are writing a linguistic rule.



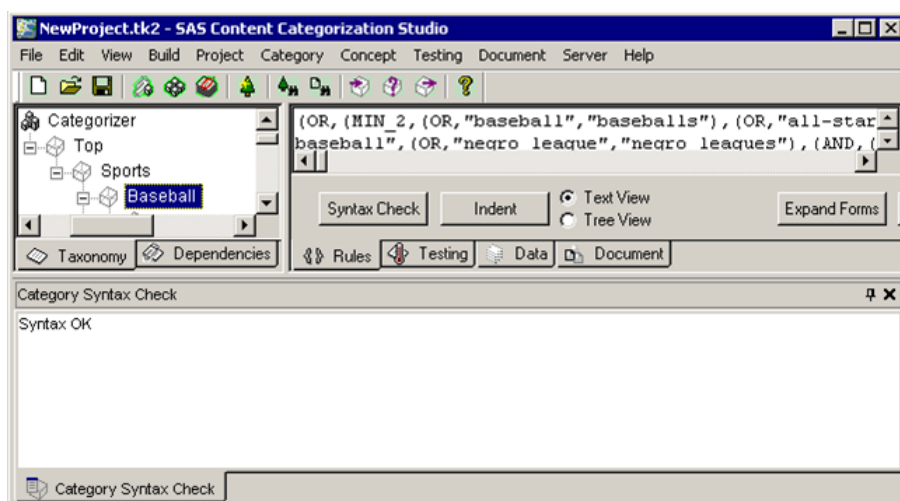
---

## 8.6 Check the Syntax of a Boolean Rule

Click **Syntax Check** in the **Rules** tab to test the grammar of a Boolean category rule in text view mode only. Syntax checking is automatic in the Tree View mode (used only with Boolean rules). If you click **Tree View**, and the Boolean rule syntax is incorrect, a SAS Content Categorization Studio status window appears and advises you of this error.

To check the syntax, click **Syntax Check**. The **Category Syntax Check** tab appears with a *Syntax OK* message.

*Display 8-2 Category Syntax Check Window*



Alternatively, if the rule requires a grammar change, the **Category Syntax Check** tab displays a status message. Use information in this message to make the required syntax changes.

---

**Hint:** When you rebuild the categorizer, syntax checking is automatically performed.

---

---

## 8.7 Differentiating Symbolic Links from Dependencies

Symbolic links are not categories and they differ from dependencies for the following reasons:

- The rule for a symbolic link is the *source* category rule that points to the *target* category rule. Dependencies, on the other hand, reference another category or a classifier concept. They use the referenced rule for *part* of their own rule. Symbolic links have no rule of their own. They are placeholder categories that reference the *source* category for their entire rule.
- Symbolic links are only pointers to another category. They do not function like categories because they do not have their own category rule.
- Symbolic links can be made between categories, only. A symbolic link for a category cannot be pasted into the concept portion of the taxonomy.
- Dependencies can be nested, while symbolic links cannot reference categories that in turn reference other categories.
- The **Dependencies** tab enables you to check dependencies. Symbolic links are not displayed in the **Dependencies** tab. If you delete the *source* category for the symbolic link, the *target* category remains in the taxonomy tree as a useless node. This is also true if there are multiple target categories.

For more information, see Section 8.9 *Creating Dependencies* on page 230.

---

## 8.8 Create Symbolic Links

### 8.8.1 About Symbolic Links

Symbolic links are navigational tools that are defined as placeholder or pointer subcategories. These links are similar to the links used by UNIX. Symbolic links can be used in a hierarchical taxonomy for categories with either Boolean or linguistic rules. These links refer to the *source* category that contains the rule. For this reason, symbolic links appear as child categories in a taxonomy.

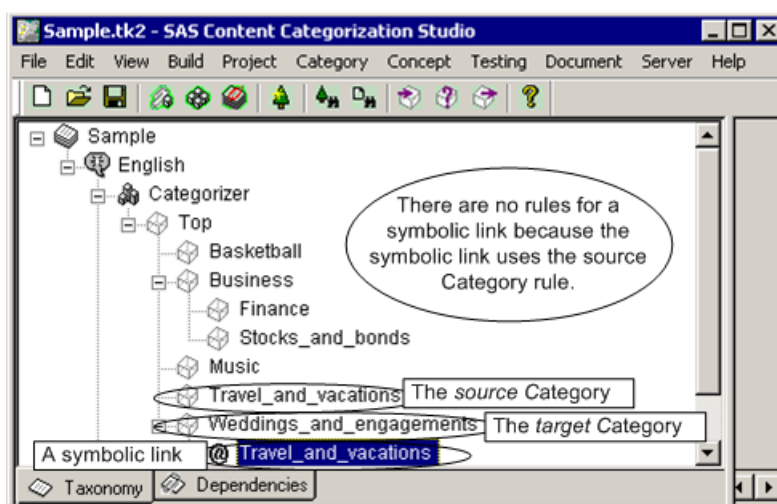
Input documents are not matched to symbolic links. Instead, the categories that appear as symbolic links serve as a reference to the *target* category. Matching documents are assigned to the category that is the target of the rule that is pointed to by the symbolic link.

Unlike regular categories and subcategories, symbolic links contain no rules and categorize no documents. The *source* category with its relevant category rules is linked to the *target* category. The symbolic link appears as a child of the *target* category. For example, the *source* category `Travel_and_vacations` might be linked to the *target* category `Weddings_and_engagements`. In this case, the symbolic link appears in the Taxonomy window as a child of the *target* category.

A symbolic link is only a link. All of the texts that match the *target* category are categorized into the *source* category. This is true no matter how many target categories are created. In other words, you could define `Travel_and_vacations` as a symbolic link for several parent categories.



Figure 8-2 Symbolic Link



## 8.8.2 Benefits of Symbolic Links

Symbolic links enable you to perform the following operations:

- Define multiple instances of symbolic links that all point to one *source* category. For example, you can define a *source* category named *Corporate Fraud*. Create a symbolic link to this source category. All of the documents that match the symbolic link are returned as a match on the *source* category.
- When you create one *source* category with a number of pointers, you need to edit only the *source* category rule. All of the symbolic links are affected.
- The testing operation tests only the *source* category rule.
- The *target* subcategories do not have **Rules**, **Testing**, **Data**, or **Document** tabs. To prevent confusion, these tabs can be used only with the *source* category.

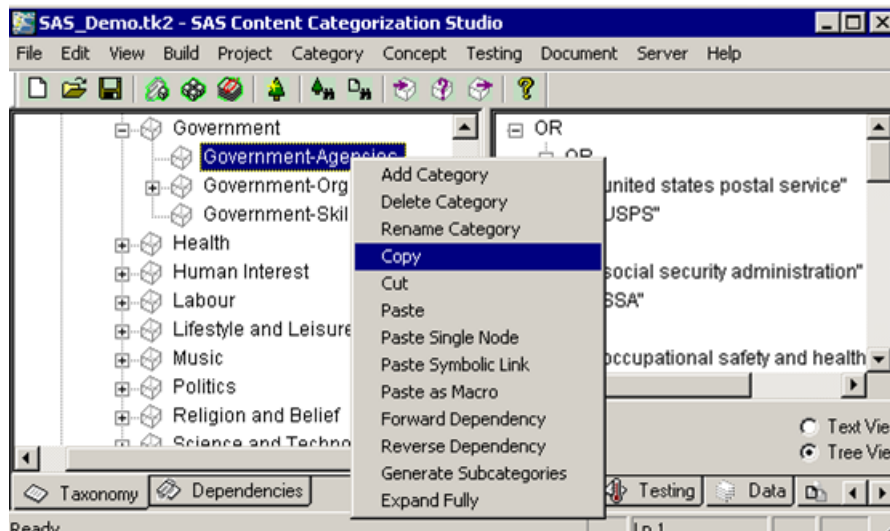
---

### 8.8.3 Define a Symbolic Link

Define symbolic links between categories when these nodes exist within a single language branch of the taxonomy. A category rule in one language branch cannot reference a category in a different taxonomy branch.

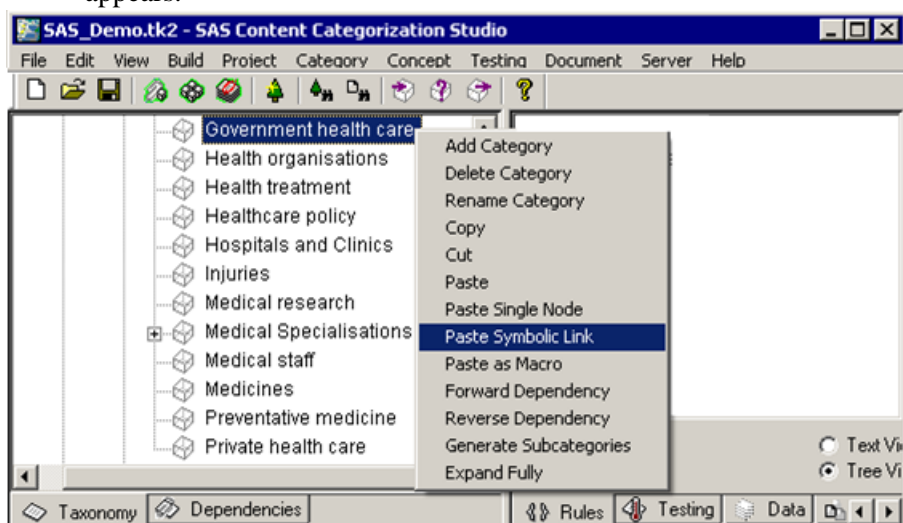
To create a symbolic link between two categories, complete these steps:

1. Right-click on the category that you plan to make the *source* category in the **Taxonomy** tab. For example, select Government-Agencies.



2. Select **Copy** from the drop-down menu that appears.

3. Right-click on the target category. For example, select **Government health care**. Select **Paste Symbolic Link** from the menu that appears.



4. Click the plus sign (+) that appears to the left of the target category to see the symbolic link. This link is represented as a child category with an at sign (@) to the left of its name.



5. Select **Build --> Build Rulebased Categorizer**.

---

## 8.9 Creating Dependencies

### 8.9.1 How Dependencies Work

Dependencies are defined when one taxonomy node references the entire rule or definition of another node in the same language branch of a taxonomy. In contrast, symbolic links use the referenced rule as if it is their whole rule. Dependencies enable you to reference an entire rule, or definition, as a building block, or a component of the rule. For example, define a long classifier rule, reference that rule without rewriting it, and add to the rule.

Dependencies can also be defined in either a flat or a hierarchical taxonomy. Unlike symbolic links, there is no symbol that appears in the Taxonomy pane to show that dependencies exist. For this reason, check the **Dependencies** tab before you delete a category or a classifier concept. If you remove a category that contains part, or all, of a rule for a dependent category, unexpected results might occur.

You can create dependencies between the following types of nodes:

- To define a dependency between two Boolean rules use a macro. For more information, see Section 11.12.2 *Paste a Macro* on page 337.
- Reference a concept within the definition of a grammar concept. For more information, see Section 19.5.4 *Defining Dependencies in Grammar Rules* on page 487.
- A classifier concept can be the source for either a linguistic or a Boolean rule. For more information, see Section 10.10 *Define Dependencies* on page 281 and Section 11.12 *Dependencies between Categories or Categories and Classifier Concepts* on page 337.

The referenced rule or definition forms *part* of the rule, but not the *whole* rule. For this reason, ensure that all of the parts of the source rule are appropriate to the development of the new category rule.

Use any of the following operations before you delete a taxonomy node in a project with dependencies. These operations make it possible for you to see any dependencies before they are eliminated.

---

**View --> Taxonomy as Text**

Use this operation to see the taxonomy nodes that are not dependent on another node for their definitions. For more information, see Section 4.4.4 *View the Taxonomy as Text* on page 148

**Dependencies tab**

You can also choose to see the nodes in the Dependencies window. Choose this operation to see forward and reverse dependencies. For more information, see Section 8.9.4.B *Checking Dependencies before Deletions and Edits* on page 238.

## 8.9.2 Benefits of Dependencies

Dependencies provide the following benefits:

**Easy-to-build category rules**

Use the unique terms defined in classifier concept definitions, or linguistic rules, to define category membership. For example, if you specified a Classical Music classifier concept definition, these unique terms might also apply to a Music category. When you create a dependency between these two nodes, the classifier terms are automatically incorporated into the Music rule.

**Shorter Category rules**

Reference one or more classifier concepts instead of writing a long list of terms for a category rule.

**Rule editing is simplified**

Edit once and affect multiple rules. In the *Music* example above, edit the definition for the *Classical Music* concept and you can also change the rule for the *Music* category. For this reason, dependencies also simplify the process of changing large, complex rules.

**Accurate rules are simplified**

When you edit once, you minimize the possibility of making errors.

---

## 8.9.3 Creating Dependencies between Categories and Concepts

### 8.9.3.A Special Considerations

Both linguistic and Boolean rules can reference concepts as dependencies. For this reason, this type of dependency is discussed in this chapter.

There are special considerations for categories that are dependent on concepts:

- Although you can automatically create dependencies between categories, you enable the inclusion of concept definitions in category rules. Use the **Concept** tab in the Project Settings window to define a dependency on a concept definition within a category rule. For more information, see Section 8.9.3.B *Specify Project Settings with Dependencies* below.
- You do not need to specify the full pathname of the classifier concept. Unlike categories, concept names are unique across the concepts namespace for each language.
- The match ratio setting, used for linguistic terms only, reads each matched concept as one matched term in the specified category rule.
- If you create dependencies with Boolean rules that reference classifier concepts and plan to use the **Server Query** operation with your index, specify lowercase characters for the referenced concepts. The **Server Query** operation is not case sensitive at this time. For more information, see Section 11.14 *Query an Index* on page 342.

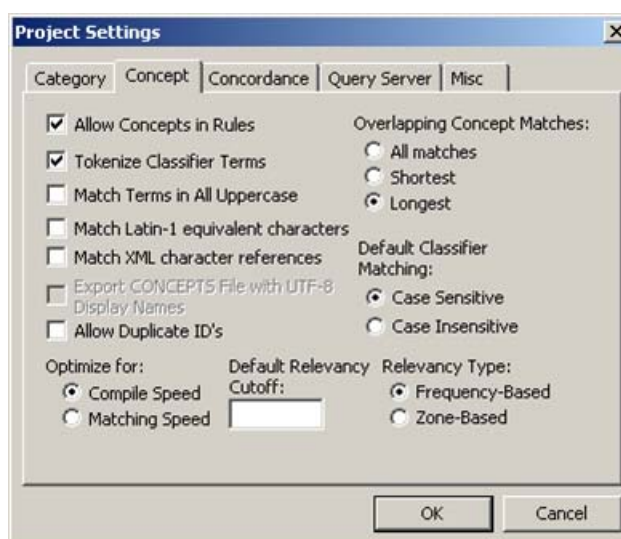
### 8.9.3.B Specify Project Settings with Dependencies

Unlike the process of creating dependencies between categories, use the **Concept** tab in the Project Settings window to enable categories to reference classifier concept definitions.

To enable classifier terms to be matched by SAS Content Categorization Studio when a classifier concept is referenced by either a linguistic or a Boolean category rule, complete these steps:

1. Select **Project --> Settings**.

- 
2. Click the **Concept** tab in the Project Settings window that appears.



3. Select **Allow Concepts in Rules**.

---

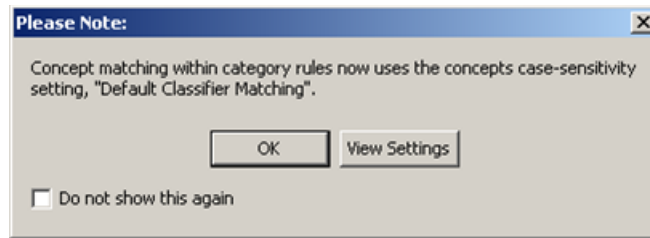
**Note:** If you do not select **Allow Concepts in Rules**, unexpected behaviors might occur.

---

4. (Default setting) Leave **Case Sensitive** selected, under the **Default Classifier Matching** heading, to restrict matching to terms that meet the upper- and lowercase specification. For example, *bush* is matched in the string *bush growing at the side of the road*. In this example, the word *Bush* in the string *President Bush* does not appear as a match.
5. (Optional) Select **Case Insensitive** to specify that anytime the input term is located, whether there is an exact case match or not, it is returned as a match. In this example, the word *bush* matches both “President *Bush*” and “*bush* growing at the side of the road.”
6. (Optional) If you access an old project in a newer version of SAS Content Categorization Studio, the Please Note window appears. This

---

window warns users that are running earlier versions of SAS Content Categorization Studio that matching is now case sensitive.



7. (Optional) Click **View Settings** to see these specifications.
8. (Optional) Select **Do not show this again** to prevent the Please Note window from reappearing.
9. Click **OK**.
10. Click **OK** in the **Concept** tab.
11. Select **Build --> Build Rulebased Categorizer**.



---

### 8.9.3.C Write the Concept Reference Syntax

Category rules that depend on a classifier concept definition use brackets ([ ]) around the name of the source concept to reference this concept. For example, to reference the RESORTS classifier concept use the following syntax in a linguistic rule:

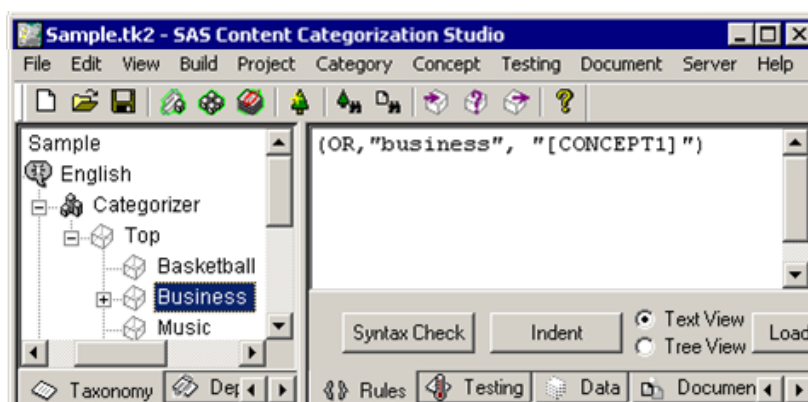
```
[RESORTS]
```

If the target category is specified by a Boolean rule, place quotation marks (") around the bracketed concept. For example, you can specify this rule:

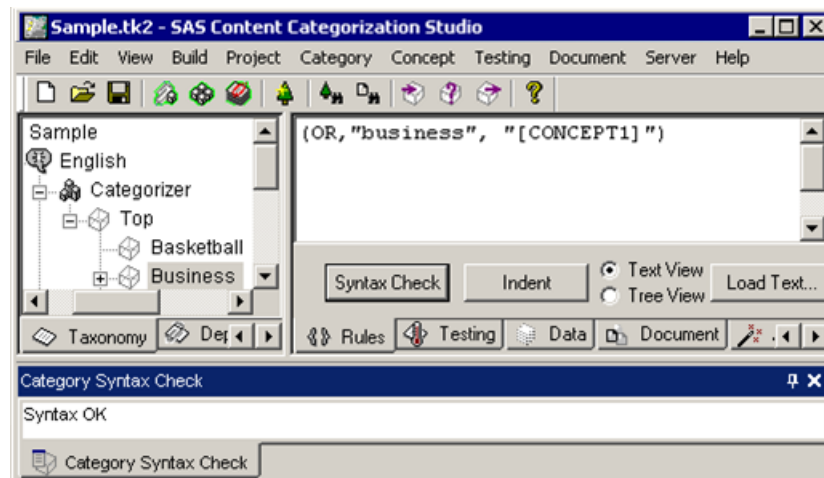
```
(OR, "vacation", "[RESORTS]")
```

To write a rule that depends on a classifier concept, complete these steps:

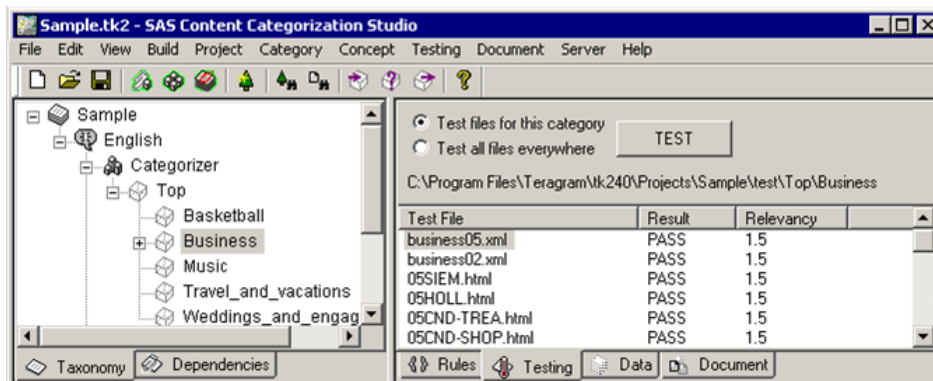
1. Compile the concepts. For more information, see Section 17.7 *Compile Concepts* on page 439.
2. Write a category rule in the **Rules** tab. Specify a classifier concept dependency using the syntax explained above.



3. Click **Syntax Check** and the **Category Syntax Check** tab appears.

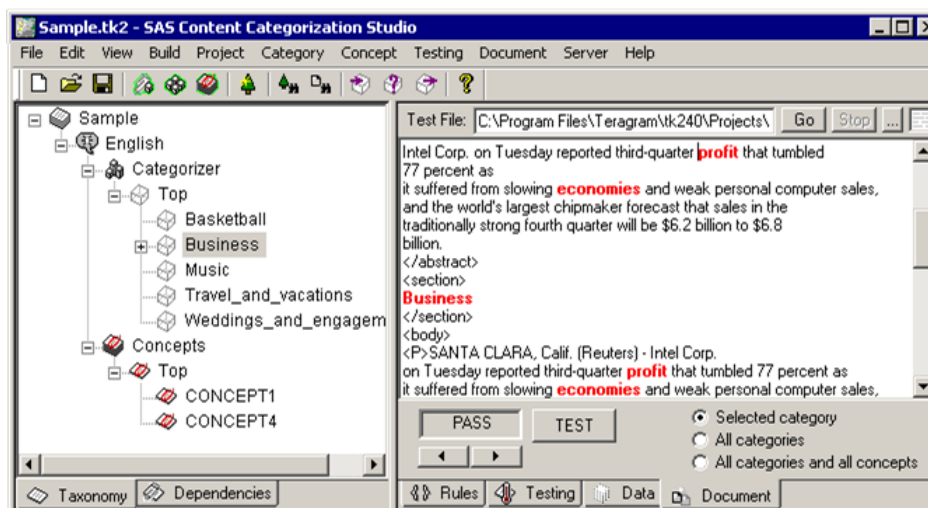


4. If the Category Syntax Check window states **Syntax OK**, select **Build --> Build RuleBased Categorizer**. If the syntax contains an error, edit the category rule.
5. To test the rule, click **TEST** in the **Testing** tab.



6. Double-click on one test document in the **Testing** tab. The text with its test results appears in the **Document** tab. The matched terms in the

input document that are specified in both the classifier concept and the category rule are highlighted in red.



## 8.9.4 Checking Dependencies Before Editing or Deleting a Category or Concept

### 8.9.4.A Knowing When to Check Dependencies

When you check dependencies before you make changes to the taxonomy, you prevent any unintended changes to these rules and definitions. Check dependencies after you perform the following operations:

- Delete a category or concept node.
- Rename a node.
- Move a node.
- Edit a rule or definition.

---

#### 8.9.4.B Checking Dependencies before Deletions and Edits

To check the Dependencies window for interdependent relationships, use either of the following operations:

- Right-click on a node in the **Taxonomy** tab and select either **Forward Dependency** or **Reverse Dependency** in the menu that appears.

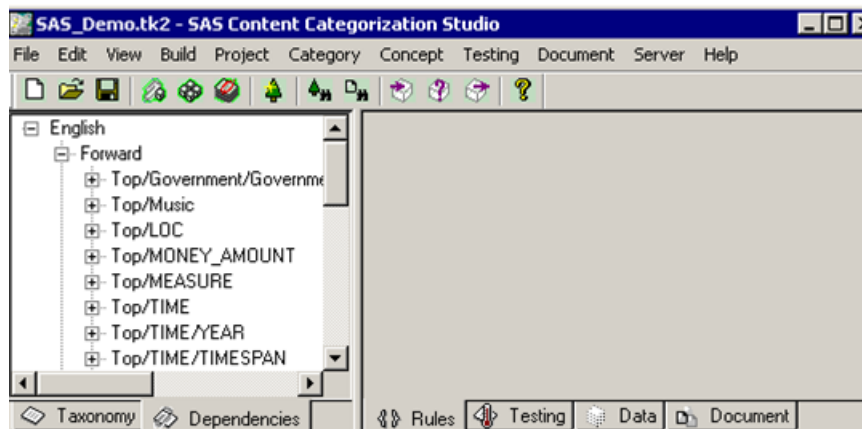
*Display 8-3 Forward Dependency Operation*



- Select the **Dependencies** tab and see the dependencies listed below the Forward and Reverse nodes.

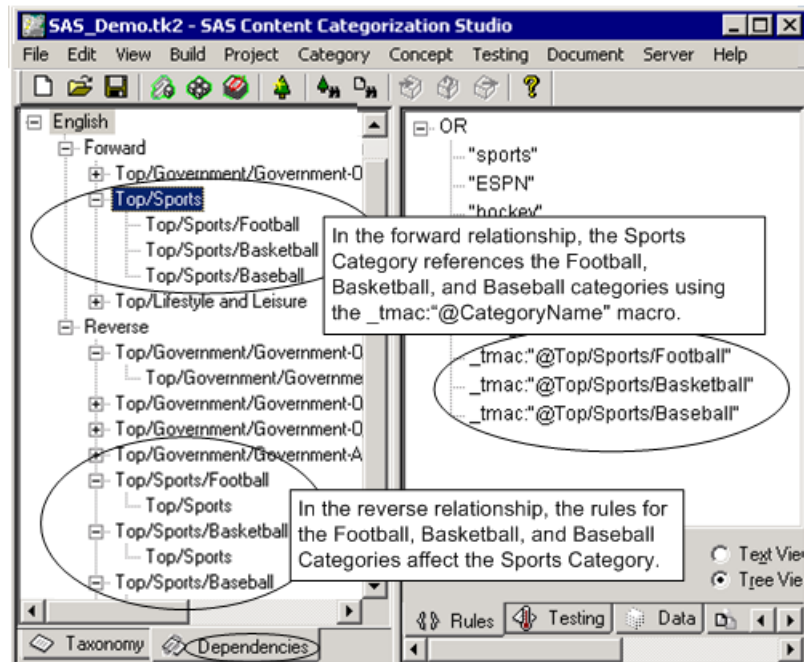
---

#### Display 8-4 Dependencies Tab



A forward dependency means that the target category uses the rule of the source category as part of its rule. A reverse dependency, on the other hand, means that the source category is referenced by the target category. A change or deletion of the rule, name, or location of a source category affects the rule of the referencing category.

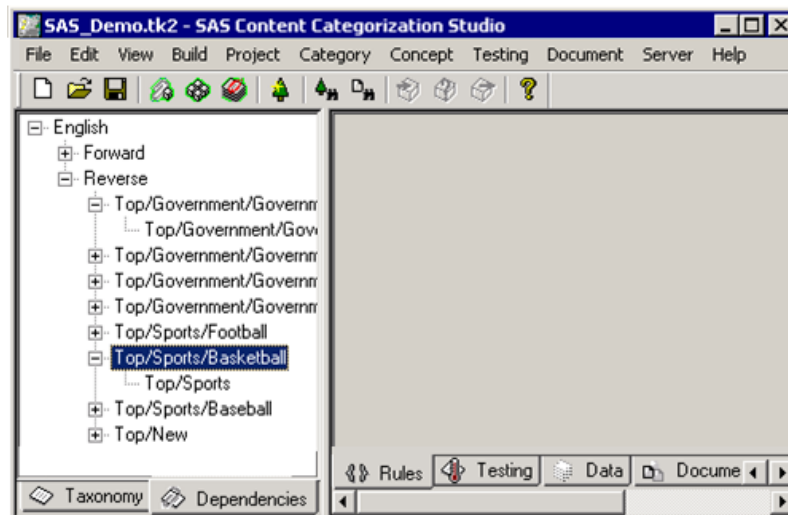
Figure 8-3 Forward and Reverse Dependencies



---

To see the relationships between the selected category and other nodes, click the plus (+) sign to the left of that node.

*Display 8-5 Reverse Dependency*



---

**Hint:** When you highlight a source, or a target, category in the Dependencies window and click the **Taxonomy** tab the same category is highlighted.

---

---

## 8.10 Building the Rule-Based Categorizer

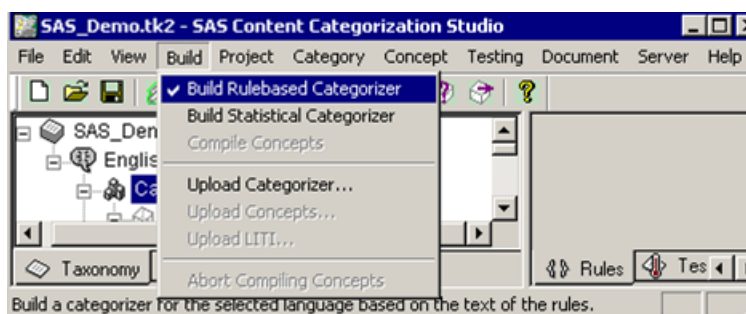
### 8.10.1 Manually Build the Categorizer

Before you test your project, build the rule-based categorizer. When you perform the build operation, a binary (.mco) file is created. This file is required for both test and upload purposes.

The build operation also runs a simple check on the rules. For this reason, building the rule-based categorizer is a necessary prerequisite to testing. It is also a useful feature when you make minor changes to a category, but you do not have time to run a full test on the categorizer.

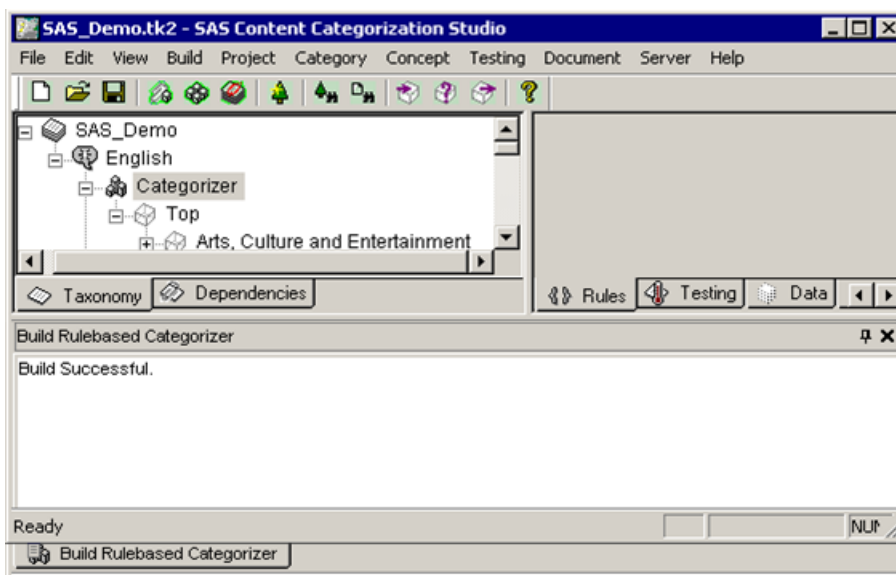
To build the rule-based categorizer, complete these steps:

1. Select either the language, or the categorizer, node in the Taxonomy window.
2. Select **Build --> Build Rulebased Categorizer**.





- 
3. SAS Content Categorization Studio builds the categorizer. If the build is successful, the Build Rulebased Categorizer window that appears at the bottom of the user interface displays the `Syntax OK` message.



---

**Tip:** If the syntax, or build, is not `OK`, the Category Syntax Check window provides the details necessary to make the required changes.

---

4. To close the Build Rulebased Categorizer window, click **X**.

---

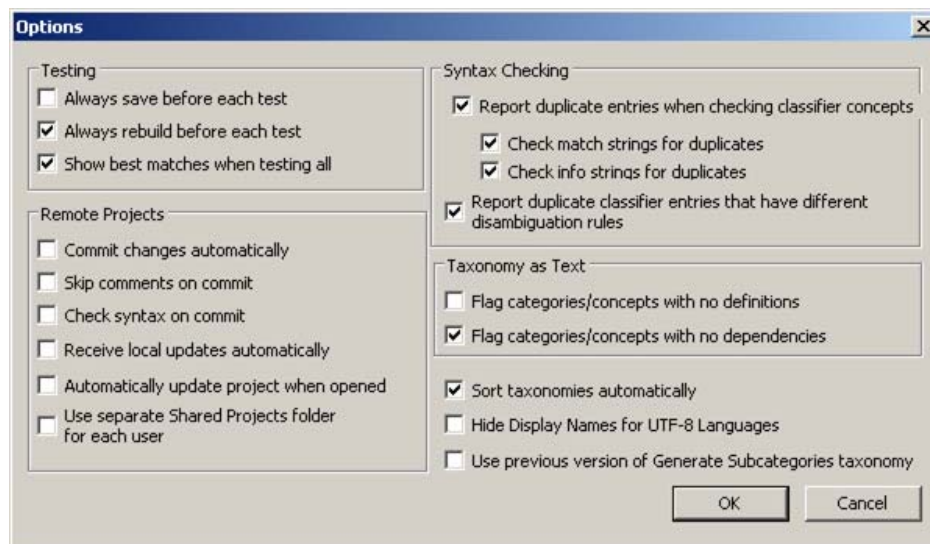
## 8.10.2 Automatically Rebuild the Rule-Based Categorizer

Rebuild the rule-based categorizer (.mco file) after you make changes to any of the category rules. The rebuild operation is necessary after you perform any of the following operations:

- Change the category rules.
- Modify the taxonomy or its nodes in any way.
- Change one or more of the category rules.

To automatically rebuild your categorizer, complete these steps:

1. Select **Edit --> Options**.
2. Select **Always rebuild before each test**.



3. Click **OK**.

When you enable this operation, your categorizer (.mco file) is automatically rebuilt before each test.

---

**Hint:** When you choose to build the categorizer, SAS Content Categorization Studio also runs a syntax check on the selected category rule.

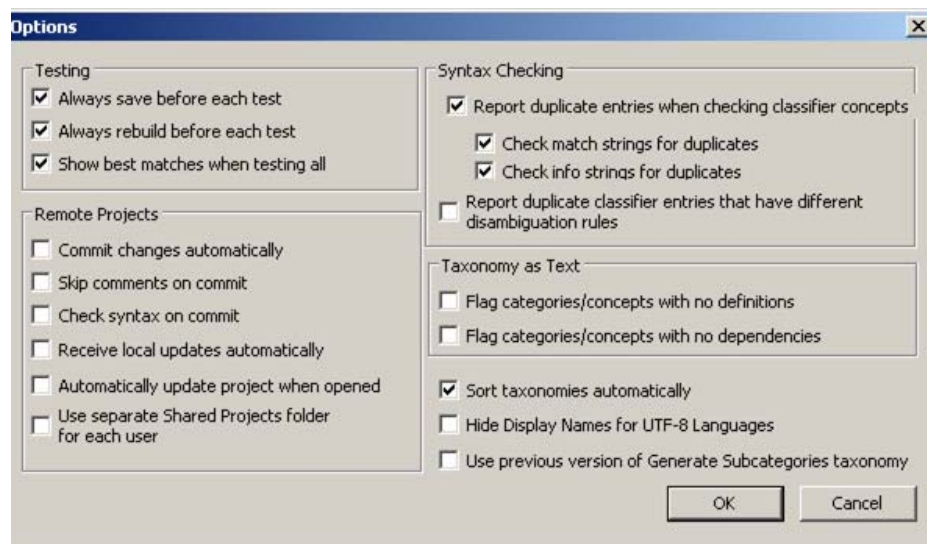
---

## 8.11 Automatically Save the Changes

Whether you choose to automatically rebuild the categorizer (.mco file), you should save the changes that you make before you test. This automatic operation saves the changes before testing begins.

To automatically save the changes to your project, complete these steps:

1. Select **Edit --> Options**.



2. Select **Always save before each test**.
3. Click **OK**.



---

# Chapter: 9

## Relevancy and the Settings That Affect Relevancy

---

- *Overview of Relevancy*
- *Determining What Relevancy Type to Use*
- *How to Set Relevancy Cutoff Settings*
- *About Relevancy and Category Bias Settings*

### 9.1 Overview of Relevancy

Relevancy is a specification that is used by SAS Content Categorization Studio to determine the best match when an input document matches more than one category. Relevancy is determined only after a document matches one, or more, category rules. For linguistic rules, this means that the Match Ratio specification is also met.

There are three types of relevancies that you can set in the **Category** tab of the Project Settings window:

#### **Frequency-based**

A document is scored by the sum of the instances of matching terms that occur in the document.

#### **Zone-based**

Rule matches are weighted according to the section of the input document where they are located.

#### **Operator-based**

The Boolean operators that are used for precision and recall purposes are weighted in addition to the matched terms. This is true for Boolean rules and for linguistic rules. Linguistic rules become Boolean rules in an internal operation that is not visible to the user.

---

---

There are several settings and specifications that can also affect relevancy. Some of these are specific to the type of rule that you write, and others apply to both types of rules. All of the settings, operators, and symbols apply to each of the three frequency types.

## 9.2 Determining What Relevancy Type to Use

### 9.2.1 How Frequency-Based Relevancy Works

Frequency-based relevancy is a count of the total number of instances of matching terms that are located in an input document. This algorithm is one measure of the best category match for an input document.

The equation for frequency-based relevancy is shown below:

$$\text{relevancy} = \text{frequency}_{\text{children}}$$

where  $\text{frequency}_{\text{children}}$  is defined as:

**frequency:** the total number of occurrences of matching terms in the document

**children:** the number of times each matching child (term) occurs in the text

Some Boolean operators limit the locations where matches can occur in a document. However, frequency-based relevancy totals all of the instances of matched terms, regardless of their location and the Boolean operators specified.

### 9.2.2 How Zone-Based Relevancy Works

Zone-based relevancy computes category matches based on the number of matched terms and their location within the document. Zone-based relevancy penalizes clusters of words in less relevant document sections. For this reason, this type of relevancy is often used for news articles.

The algorithm for zone-based relevancy works by dividing a document into three equal sections and computing the relevancy score for the matches that are located in each of these zones:

1. The matching rules in the first section receive the highest weighting.
2. Results that occur in the second zone get the second highest weighting.
3. Matches in the third section are assigned the lowest weight.

The relevancy score is based on the length of the document and the number of rule matches that occur within each zone. For this reason, if both a long and short document have the same number of matching terms, the shorter text is more relevant. The zones in small documents are equally weighted.

The scores for each of the three document sections are then combined into a single number for the entire document that ranges between 0 and 10. 0 is the least relevant and 10 is the most relevant.

The base algorithm for zone-based relevancy is shown below:

$$\text{RELEVANCY} = \text{ALPHA} * (\text{WG} * \text{RG} + \text{W1} * \text{R1} + \text{W2} * \text{R2} + \text{W3} * \text{R3}) + (1 - \text{ALPHA}) * (\text{MAX}(\text{R1}, \text{R2}, \text{R3}))$$

The components of this rule are explained in the table below:

Table 9-1: Rule Terms

Rule Term	Description
RELEVANCY	The total relevancy score.
ALPHA	The constant between 0 and 1 that balances the relevancy score for all three zones with the score for the zone with the highest relevancy. This parameter prevents a document from being assigned a relevancy score that is inappropriately low if all of the matches are in zones two or three. In this case, the relevancy score for this document is low, but appropriate.
<b>Note:</b> This algorithm assigns a higher weight to rule matches located within the document title. The algorithm also penalizes tight clusters of matching terms that occur in zones two and three. Word clusters often distort the overall relevance of a document.	
WG	The global weight that applies to the entire document.
RG	This relevancy score for the entire document is computed by a heuristic that takes into account the number of matches and the length of the document. For example, 10 matches in a document of 200 words means that this text has a higher relevancy score than 10 matches in a 2,000 word document.

Table 9-1: Rule Terms (Continued)

Rule Term	Description
Wx	The weight assigned to a specific zone.
Rx	The relevancy score for a zone.
MAX ( R1 , R2 , R3 )	The highest of the relevancy scores for the three zones. For example, if the relevancy for zone one is two, four for zone two, and 1.5 for zone three, MAX ( R1 , R2 , R3 ) is four.

### 9.2.3 How Operator-Based Relevancy Works

Relevancy criteria assigns higher relevancy weights to Boolean category rules that have the most coverage and to stronger Boolean operators. The first example below shows how two rules that both use the OR operator are applied against two different documents. The example below shows that the OR operator is ranked higher than the AND operator:

*Example 9-1: Relevancy Weights Example Using the OR Operator*

Category A uses the OR operator with one term:

( OR , "a" )

Category B uses the OR operator with two terms:

( OR , "a" , "b" )

If *Document 1* contains one occurrence of term a, both of the categories are matched. If *Document 1* does not contain the term b, category A is more relevant. However, if *Document 2* contains one occurrence of term a and one occurrence of term b, category B is considered a better match.

*Example 9-2: Relevancy Weights Example Using Different Operators*

Category B uses the operator OR to modify the a and b terms in the rule:

( OR , "a" , "b" )

Category C uses the operator AND to modify the a and b terms in the category rule:

( AND , "a" , "b" )

Using the example above, Document 2 contains one occurrence of each of the a and b terms. Therefore, both categories are matched. However, category B is



---

ranked higher than category C because category B has an OR operator in its category rule.

## 9.2.4 How Boolean Operators Affect Relevancy Weights

Relevancy for Boolean rules depends on the weight assigned to the various Boolean operators and the presence of matching terms in the input document. See the table below:

Table 9-2: Relevancy Weight Computations

Operator	Relevancy Formula	Special Case of
AND	$\Sigma \text{weight}_{\text{children}}/(\text{nb}_{\text{children}}+1)$	
DIST	$\Sigma \text{weight}_{\text{children}}/(\text{nb}_{\text{children}}+1)$	AND
END	$\Sigma \text{weight}_{\text{children}}/(\text{nb}_{\text{children}}+1)$	AND
MAXOC	$\Sigma \text{weight}_{\text{children}}/(\text{nb}_{\text{children}}+1)$	AND
MAXPAR	$\Sigma \text{weight}_{\text{children}}/(\text{nb}_{\text{children}}+1)$	AND
MAXSENT	$\Sigma \text{weight}_{\text{children}}/(\text{nb}_{\text{children}}+1)$	AND
MIN	$((\Sigma \text{weight}_{\text{children}} - \text{nb}_{\text{min}})/\text{nb}_{\text{children}})+1$	OR
MINOC	$\Sigma \text{weight}_{\text{children}}/(\text{nb}_{\text{children}}+1)$	AND
NOT	1	
NOTIN	$\text{weight}_{\text{child}}$	
NOTINPAR	$\text{weight}_{\text{child}}$	
NOTINSENT	$\text{weight}_{\text{child}}$	

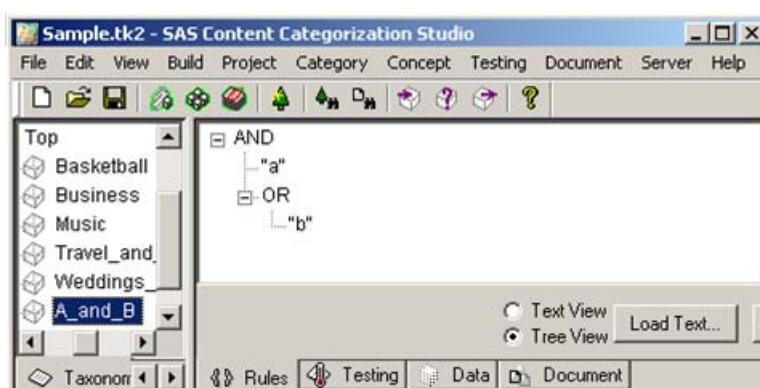
Table 9-2: Relevancy Weight Computations (Continued)

Operator	Relevancy Formula	Special Case of
OR	$((\sum \text{weight}_{\text{children}} - 1) / \text{nb}_{\text{children}}) + 1$	
ORD	$\sum \text{weight}_{\text{children}} / (\text{nb}_{\text{children}} + 1)$	AND
PAR	$\sum \text{weight}_{\text{children}} / (\text{nb}_{\text{children}} + 1)$	AND
PARPOS	$\sum \text{weight}_{\text{children}} / (\text{nb}_{\text{children}} + 1)$	AND
SENT	$\sum \text{weight}_{\text{children}} / (\text{nb}_{\text{children}} + 1)$	AND
START	$\sum \text{weight}_{\text{children}} / (\text{nb}_{\text{children}} + 1)$	AND

$\text{nb}_{\text{children}}$  is defined as the total number of sub-nodes under the operator. For example, see the rule (AND, "a", (OR, "b")). AND has two sub-nodes, while OR has only one sub-node.

The weights for each matched, terminal child are 1. This number is added to the count for the number of children. The propagated weight for each matched non-terminal child is added to the count for the number of children. For example (OR, "b") is a non-terminal child of AND in the rule (AND, "a", (OR, "b")).

Display 9-1 Child Node for the Operators AND and OR



## 9.2.5 How Stemming Affects Relevancy

Special symbols can affect frequency-based relevancy. For example, if you prefix an at sign (@), with or without the letters N or V, stemming has the potential to increase the number of matches. Stemming adds more terms to the rule and increases the number of possible matches.

## 9.3 How to Set Relevancy Cutoff Settings

### 9.3.1 Analyzing Relevancy Cutoff

The relevancy cutoff is the minimum relevancy value. Unless an input document meets the relevancy cutoff for a category, it is not considered a category match. This is true even if the document matches the category rule.

When you specify the **Default Relevancy Cutoff** field in the **Category** tab, this setting is used for all of the categories in the taxonomy. You can also set the **Relevancy Cutoff** field in the **Data** tab to specify a different relevancy cutoff value for a specific category. Texts that contain a relevancy value that is less than the number specified for either cutoff value are excluded from category membership.

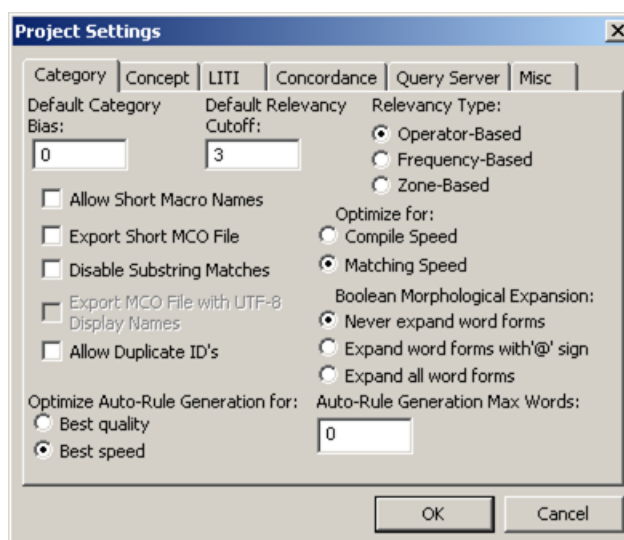
---

Documents that match the rule, but fall below the relevancy cutoff setting, are marked with an asterisk (\*). This specification appears after the `PASS` message in the **Testing** tab. The asterisk indicates that these documents pass conditionally.

### 9.3.2 Specify Relevancy Cutoff Values

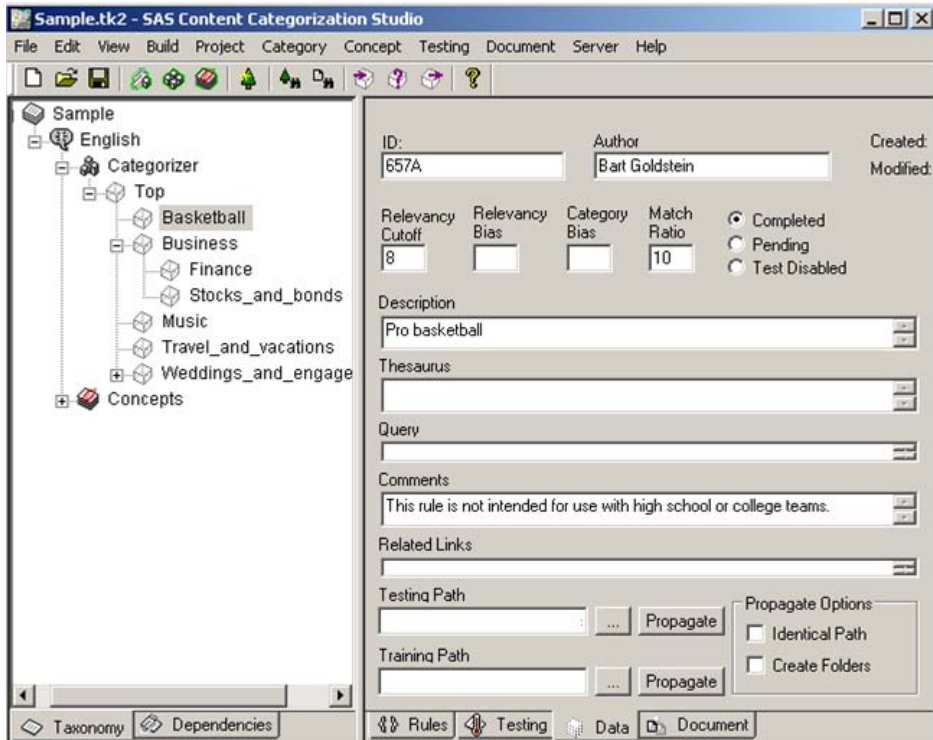
To specify the relevancy cutoff for either the taxonomy of categories or a single category, complete these steps:

1. Select **Project --> Settings**.
2. The **Category** tab appears in the Project Settings window. Enter a number such as 3 into the **Default Relevancy Cutoff** field. This specification applies to the **Relevancy Type** that you select. By default, **Operator-Based** is selected.



After you set the **Default Relevancy Cutoff** value, you can change the **Relevancy Cutoff** setting for one, or more, categories in the **Data** tab.

3. Select a category in the **Taxonomy** tab and click the **Data** tab to set the **Relevancy Cutoff** value. For example, specify 8.




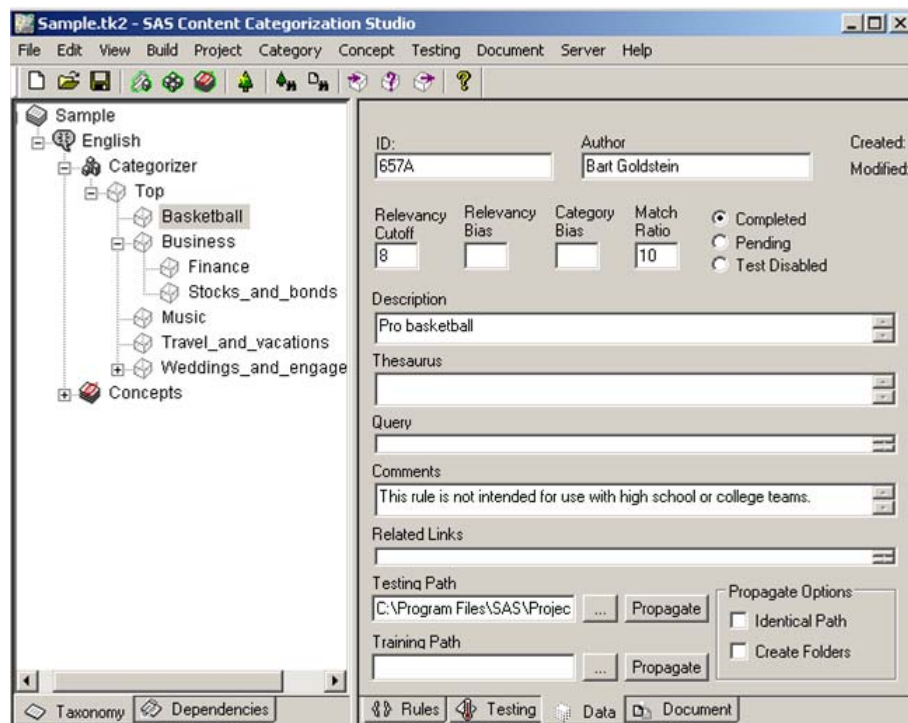
---

### 9.3.3 Test to Compute an Approximate Default Relevancy Cutoff Setting

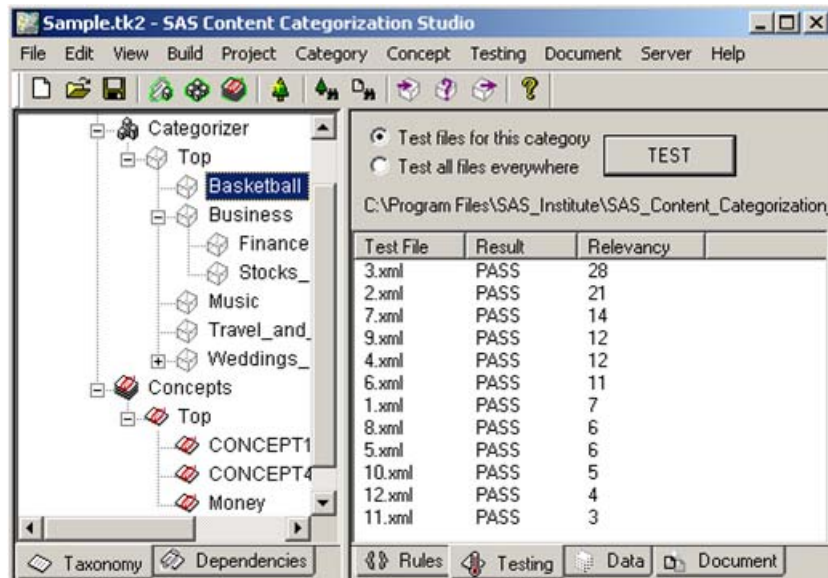
Often the testing process is required in order to specify an optimal relevancy cutoff setting. For this reason, you can use the testing process to determine these numbers.

To test a taxonomy for the purposes of obtaining the optimal relevancy settings, complete the following steps:

1. Using the directions in Chapter 12: *Assembling Testing Sets*, assemble a testing set of documents for the selected category. For example, assemble documents for the `Travel_and_vacations` category. Click  and use the Open window that appears to set the **Testing Path** in the **Data** tab.



2. Click **Propagate**.
3. Click the **Testing** tab.



4. (Optional) If the default selection **Test Files for this category** is not selected, click this radio button.
5. Click **TEST** and the testing results appear in the **Testing** tab.
6. Use the following columns in the **Testing** tab to determine how to set the **Default Relevancy Cutoff** setting:

#### Test File

The name and type of the test file appears below this heading.

#### Result

The **PASS**, **FAIL**, and **PASS\*** (conditional passing for documents that match the rule, but fail to meet the relevancy threshold) messages appear.

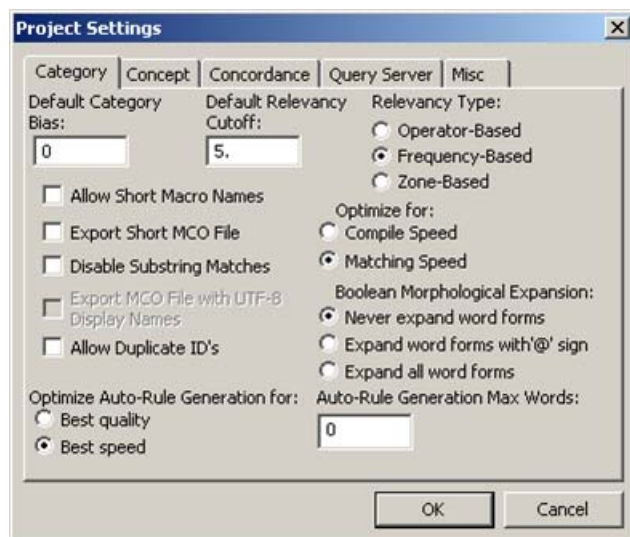
When a document passes the match ratio specification, but falls below the **Default Relevancy Cutoff**, the document has an asterisk (\*) after the word **PASS**.

#### Relevancy

---

This is the relevancy score for each tested category.

7. Use the relevancy scores to consider an appropriate number for the **Default Relevancy Cutoff** field in the **Category** tab. Use this data to also decide whether you should specify a number in the **Relevancy Cutoff** field of the **Data** tab.
8. Enter the number that you derived from the testing results into the **Default Relevancy Cutoff** field of the **Category** tab.



9. Click **OK**.
10. (Optional) Reset the **Relevancy Cutoff** setting in the **Data** tab for one, or more, categories.
11. Select **Build --> Build Rule-based Categorizer**.
12. Test the category to see how this setting affects category matching.



---

## 9.4 About Relevancy and Category Bias Settings

### 9.4.1 How Relevancy and Category Bias Settings Are Determined

You can change the way that relevancy is determined for matching documents when you specify the **Category Bias** and **Relevancy Bias** settings, or all three settings. Use the **Relevancy Bias**, **Default Category Bias**, and **Category Bias** fields for these two purposes, among other possible uses:

First, boost the relevancy of one category in relationship to all of the other categories in the taxonomy using the **Relevancy Bias** field in the **Data** tab. For example, you can specify a single term for some category rules where one term unambiguously identifies the category, such as the term SARS. The relevancy score for this category match is lower than the score for category rules where multiple terms are matched in an input document.

Second, choose to boost the relevancy of all of the categories in the taxonomy *and* the relevancy of one of these categories. For example, boost the relevancy scores of all of the categories in the taxonomy into the range used by third-party software. Within this higher range, boost the score of one category such as H1N1. To perform these operations, use the **Default Category Bias** field in the **Category** tab and the **Category Bias** field in the **Data** tab.

The equation for these relevancy settings is specified below:

$$(\text{defcatbias} * \text{catbias}) + (\text{relevancy} * \text{relbias}) = \text{new\_relevancy}$$

The default values for these settings are specified below:

**Default Category Bias:** 0

**Category Bias:** 0

**Relevancy Bias:** 1

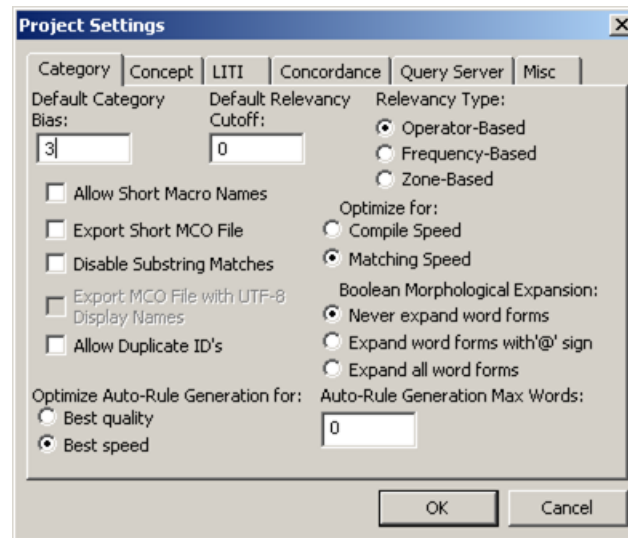
If you change the value in the **Default Category Bias** field in order to boost the relevancy for your taxonomy, set the **Category Bias** setting to 1. Take this step for each category in the taxonomy. You can also set the category bias to another number for any category whose relevancy you want to boost on an individual basis.

---

## 9.4.2 Setting the Default Category Bias

To increase the relevancy bias of all of the categories that comprise the taxonomy, use the **Default Category Bias** field in the **Category** tab. When you reset this number, you boost the relevancy for each category across the entire taxonomy into the range used by a third-party software product. The default setting is 0.

*Display 9-2 Default Category Bias Field*



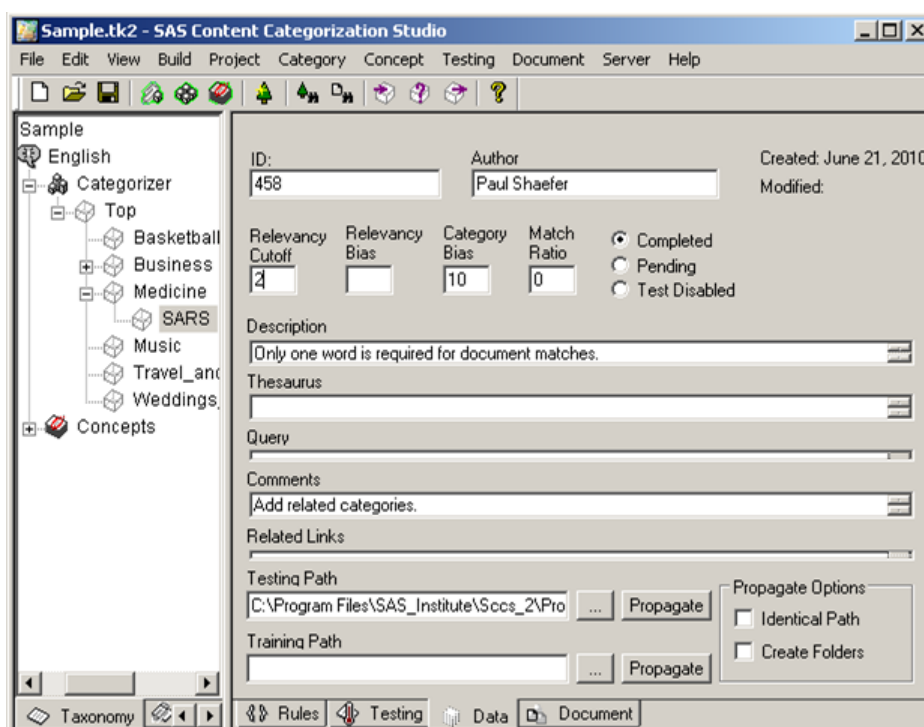
---

### 9.4.3 Set Category and Relevancy Bias

The **Category Bias** and **Relevancy Bias** fields are set by default to 0. For this reason, these two fields appear blank in the **Data** tab. If you specify a number for **Default Category Bias**, this setting is not effective unless you also specify 1, or a higher number, in the **Category Bias** field. Similarly, unless you change the default setting 0 in the **Category Bias** field to at least 1, the change in the **Default Category Bias** field is not effective.

To set the **Category Bias**, complete these steps. (Make the appropriate changes if you are setting **Relevancy Bias** instead.)

1. Type a number into the **Default Category Bias** field in the **Category** tab.
2. Select a category in the **Taxonomy** tab and click the **Data** tab.



3. Type a number into the **Category Bias** field. For example, enter 10.

- 
4. (For the **Category Bias** field, only) Repeat Step 3 above for every category in the taxonomy. Enter a lower number than the number that you select for the category whose relevancy you want to boost in relation to the rest of the taxonomy. For example, type 1 into the **Category Bias** field for every other category.

---

# Chapter: 10

## Rule-Based Categorizer: Linguistic Terms

---

- *Overview of the Rule-Based Categorizer*
- *Benefits of Linguistic Rules*
- *Quick Start Guide*
- *About the Three Types of Linguistic Rules*
- *Writing Rules in the Rules Window*
- *Weight Linguistic Rules*
- *Specifying the Match Ratio*
- *Selecting Special Symbols*
- *Create Symbolic Links*
- *Define Dependencies*

### 10.1 Overview of the Rule-Based Categorizer

Linguistic terms are the unique identifying words that you use to define a category. These strings should express the ideas that differentiate each category in the taxonomy. To qualify these terms, add special symbols.

Linguistic rules are explained in this chapter. Before you define Boolean rules, you should understand how to write linguistic rules. The terms that are used to define linguistic rules form the basis of Boolean rules. For more information, see Chapter 11: *Rule-Based Categorizer: Boolean Terms*.

---

---

## 10.2 Benefits of Linguistic Rules

Use the linguistic rule-based categorizer to obtain the benefits of humanly defined rules:

- Write linguistic rules in less time than it takes to develop Boolean rules.
- Use the list format of the linguistic rule to write initial rules. Modify the linguistic terms when you add Boolean operators.
- Define weighted linguistic rules by specifying weights for each term. You can also specify a threshold value for category membership.
- Modify linguistic rules with special symbols, word form expansion, and other settings to exclude, include, or to prioritize category membership.
- Use the **Match Ratio** field in the **Data** tab to specify the percentage of matched terms that make an input document a match on a category rule.
- Dependencies between categories with linguistic rules and classifier concepts enable you to reference a classifier definition as part of a category rule.
- Set a minimum threshold for the count of the matched terms and their occurrences. To set this number, use the **Default Relevancy Cutoff** field in the **Category** tab of the Project Settings window. Use this setting with frequency-based ranking, only.
- Boost the relevancy of the categories into the range that is used by a third-party software product. Specify this number in the **Default Category Bias** operation in the **Category** tab of the Project Settings window.
- Use the **Category Bias** and **Relevancy Bias** fields in the **Data** tab to weight and rank individual categories within the taxonomy. Use the **Category Bias** operation to weight a category more heavily by *adding* to the relevancy score. Relevancy bias enables you to adjust matching in order to limit duplicate category membership and to ensure that some categories are ranked higher than others and are therefore more relevant. The number that you enter in the **Relevancy Bias** field in the Data window is used to *multiply* the relevancy score to rank the selected category higher.

- 
- Dependencies between categories with linguistic rules and classifier concepts enable you to reference a classifier definition as part of a category rule.

## 10.3 Quick Start Guide

To build and deploy the linguistic categorizer, use some or all of the steps outlined below:

1. Specify installation-specific operations using the Options window:
  - Select **Testing** and choose:
    - Always save before each test**  
Automatically save the changes before testing.
    - Always rebuild before each test**  
Automatically rebuild the project before testing.
    - Show best matches when testing all**  
See the best rule matches in the Best Matches window when you use the **Document** tab.
  - Select **Sort taxonomies automatically** to alphabetize the nodes in the **Taxonomy** tab.
  - Under **Taxonomy as Text** select:
    - Flag categories/concepts with no definitions**  
Write definitions for each of these nodes before you apply the categorizer or concepts extractor.
    - Flag categories/concepts with no dependencies**  
Mark the taxonomy nodes that have rules that use another category or concept rule. (You can safely delete these nodes.)
2. (Optional) If you are writing weighted linguistic rules, some of the following steps do not apply. For more information, see Section 10.6 *Weight Linguistic Rules* on page 273.
3. Specify project-specific settings using the **Category** tab:
  - Default Category Bias**

---

Assign more weight to categories to boost them into the range used by third-party software.

**Default Relevancy Cutoff**

Specify the minimum relevancy that makes a document a match for this category.

**Relevancy Type**

Change the default setting **Operator-based** matching to **Frequency-based**, or to **Zone-based**. For more information, see Section 9.2 *Determining What Relevancy Type to Use* on page 248.

**Disable Substring Matches**

Choose this operation, unless you want to enable partial term matches. For more information, see Section 2.9.2.A *The Category Tab* on page 49.

**Export Short MCO File**

Produce a \*.short.mco file. This is a categorization binary file where the category names that are returned are the short names, instead of the full pathnames.

**Duplicate ID's**

Enable this operation if you want two or more categories to share the same identification number.

4. Specify project-specific settings using the Project Settings - Misc window:

**Compatibility Date**


If you are running an older version of SAS Content Categorization Server, enter the date of this version. SAS Content Categorization Studio generates a binary file (.mco or .concepts) that is compatible with the older version of SAS Content Categorization Server. This date makes the .mco or .concepts file compatible with the older version of SAS Content Categorization Server until you have time to install the updated application.

**Use UTF-8 Test Files**

Select this operation if your test files are in UTF-8 format, even if the language of the categorizer is not in UTF-8 format.



---

Click  to set the location for the **Directory for Unmatched Populate Files**. SAS Content Categorization Studio places all of the unmatched testing files into this location.

5. Use the **Data** tab to specify metadata for each category:

**Match Ratio**

Specify the percentage of terms that make a document a match for this category rule. The default is 10%. For more information, see Section 10.7 *Specifying the Match Ratio* on page 276.

**Default Category Bias or Default Relevancy Cutoff**

Use the first setting if you are using the results with third-party software and want to boost the results into a range used by this software.

**Category Bias, Relevancy Bias, or Relevancy Cutoff**

These settings affect the rankings of matched categories in the taxonomy and specify the relevancy that is necessary for category matches. For more information, see Section 9.4 *About Relevancy and Category Bias Settings* on page 259.

6. Use special symbols to qualify the rules. When you qualify your rules with special symbols, you affect the use of the various matched terms in a document. These symbols can also affect the match ratio and other settings. For example, when you enable word stemming more matches occur. For more information, see Section 10.8 *Selecting Special Symbols* on page 278.
7. Select a category in the **Taxonomy** tab and use the **Rules** tab to access the Rules window in the default text view mode.
8. Write a list of words that uniquely identify the selected category in the **Rules** tab. For more information, see Section 10.5.2 *Write a Linguistic Rule* on page 271.
9. Specify a new match ratio setting. For more information, see Section 10.7 *Specifying the Match Ratio* on page 276.
10. Assign special symbols to words in the rules. For more information, see Section 10.8 *Selecting Special Symbols* on page 278.

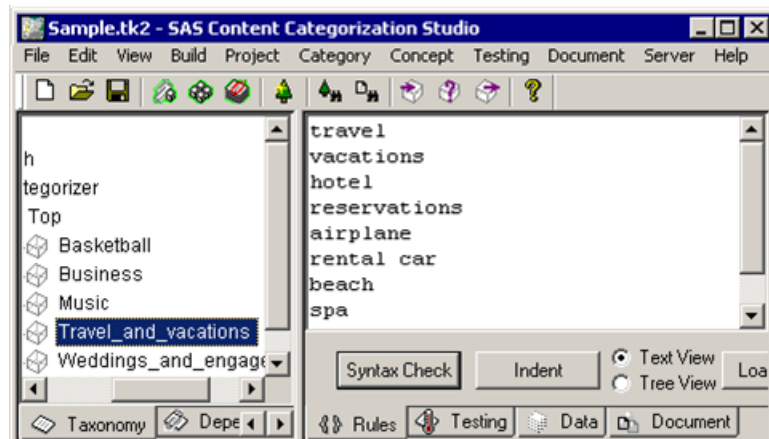
- 
11. Select a relevancy type. For more information, see Section 9.2 *Determining What Relevancy Type to Use* on page 248.
  12. Boost the relevancy of one or more categories in the taxonomy. For more information, see Section 9.4.3 *Set Category and Relevancy Bias* on page 261.
  13. Create symbolic links where the *target* category uses the whole rule of the *source* category. For more information, see Section 10.9 *Create Symbolic Links* on page 281.
  14. Define dependencies. For more information, see Section 10.10 *Define Dependencies* on page 281.
  15. Test the taxonomy. For more information, see *Part 2: Testing*.
  16. Make any necessary changes.
  17. Retest.
  18. Upload the categorizer. For more information, see Section 2.10 *Upload the Categorizer or Concepts to SAS Content Categorization Server* on page 138.

---

## 10.4 About the Three Types of Linguistic Rules

Linguistic rules are the key words that uniquely describe a specific category. Before you develop your rules, identify these key words. These terms identify members of one category and exclude these texts from matching another category.

*Display 10-1 Linguistic Category Rule*



There are three basic types of linguistic rules:

### Unqualified linguistic rules

Write a list of words that are not case-sensitive. Unqualified linguistic rules also do not use any special symbols.

### Qualified linguistic rules

Modify the unique terms by adding Special Symbols to the beginning or end of a term. For more information, see Section 10.8 *Selecting Special Symbols* on page 278.

---

**Note:** Use the Project Settings and Data windows to make additional category membership modifications.

---

### Weighted linguistic rules

Write a list of terms, add a comma (,) followed by a number to assign weight to each instance of a matched term that is located in a specific

---

document. When you assign weights to rule terms, you can also specify a threshold weight that determines the cutoff for category membership based on the value of the matched terms. If the total weight of the matched terms equals, or exceeds, the threshold weight, the document is considered a member of the selected category. For more information, see Section 10.6 *Weight Linguistic Rules* on page 273.

## 10.5 Writing Rules in the Rules Window

### 10.5.1 Overview of the Components of the Rules Window

Whether you choose to edit your automatically generated rules, or to create new rules, work in the Rules window.

*Display 10-2 Rules Window*



---

Use the following components in the **Rules** tab to write, or edit, your linguistic rules:

#### Text View

(Default selection for both the Boolean and linguistic rules and the only selection available for writing linguistic rules) Linguistic rules are written as separate lines of text, one new line for each term.

#### Load Text

Use this operation to load the rules, written in another program, into this window. For example, use *Notepad* to write your rules and then upload them. For more information, see Section 8.5.2 *Write Rules* on page 222.

## 10.5.2 Write a Linguistic Rule

Write a basic, or unqualified, linguistic category rule. This is a rule that you can later qualify using special symbols or weights.

To write a basic linguistic rule, complete these steps:

1. Select a category node in the **Taxonomy** tab.



2. Click the **Rules** tab and it appears in **Text View**, the default setting.
3. Type a list of unique identifying terms in the Rules window. All, or some, of these terms should be common to the documents that become

---

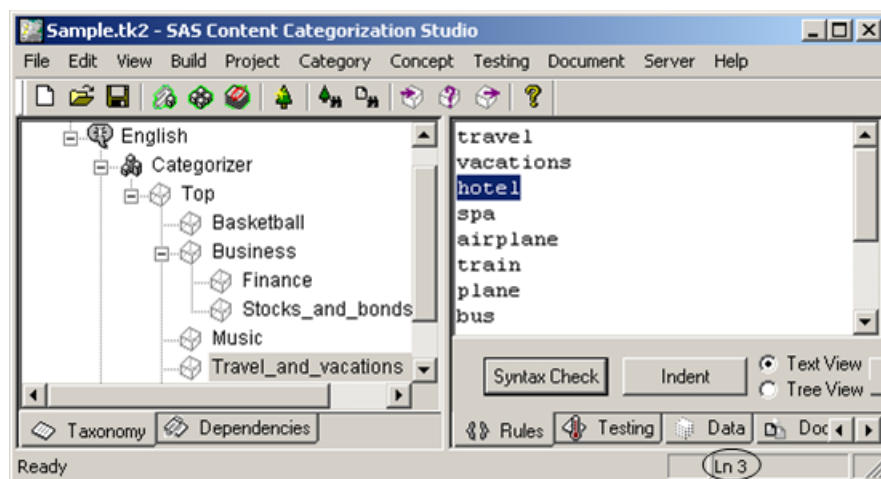
members of the selected category. They should also be unique to this group of texts. See the rule example shown below:

*Example 10-1: An Unqualified Linguistic Rule for Finance*

```
money  
bank  
loan  
credit  
application  
borrow  
lend  
credit line
```

Enter each term on a separate line. Since category rules are not case sensitive, it does not matter what combinations of uppercase and lowercase text you use.

The line number for the entered text is visible on the right-hand side of the user interface. The letters Ln are followed by the number of the selected line.



4. Select **Build --> Build Rulebased Categorizer**.
5. Select **File --> Save**.

---

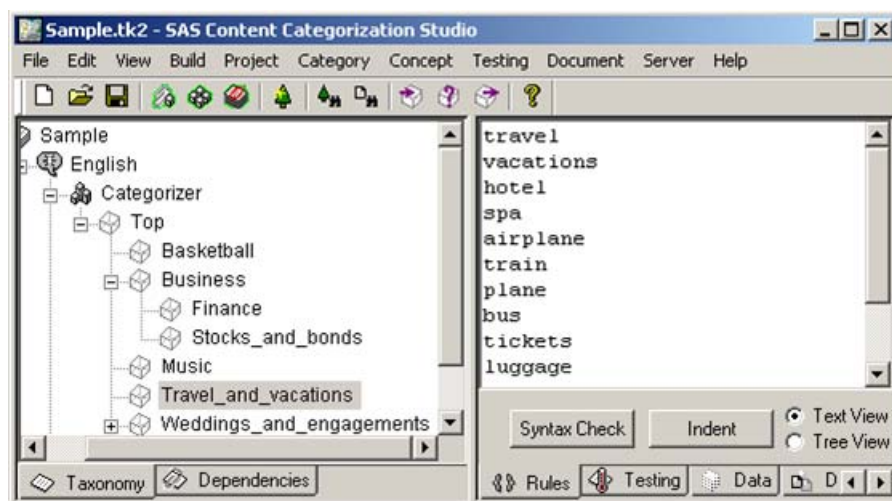
## 10.6 Weight Linguistic Rules

Write weighted category rules for some or all of the categories in your taxonomy. In this case, weight is used to determine category membership. A weighted category rule specifies the weight assigned to each occurrence of a term. This rule also specifies the threshold that determines the sum of the weights necessary for category membership.

This type of category rule does not use any of the special symbols, relevancy, bias, or the match ratio specifications that are used with other forms of linguistic rules. When you weight a category, unless the rule terms occur with sufficient total frequency, the threshold weight is not met.

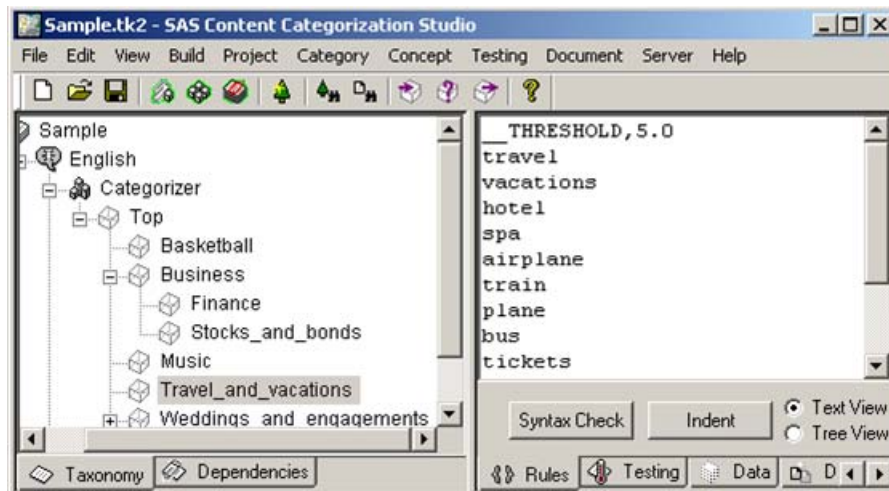
To weight a linguistic rule, complete these steps:

1. Select a category in the taxonomy.



2. Click the **Rules** tab and enter a list of the terms that comprise the category rule.

- 
3. Type a threshold weight into the first line of the category rule. Use the following syntax: `__THRESHOLD, <threshold_weight>`.



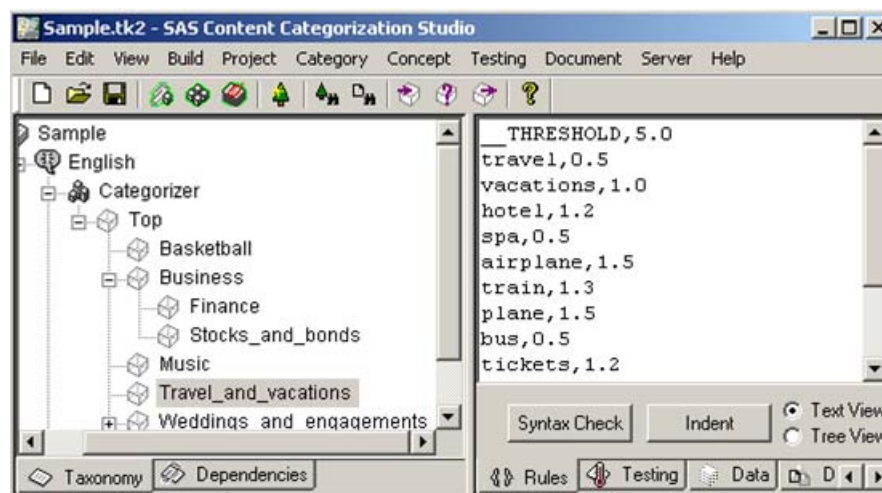
---

**Note:** There are two underscores (\_\_) before the word THRESHOLD. This term THRESHOLD is specified in all uppercase letters. Spaces do not appear before, or after, the comma (,) for either the word THRESHOLD or the weight for each term.

---



- 
4. Type a comma (,), to the right of each term in the rule, followed by a number that indicates how each match on this term is weighted. Use the following syntax, <rule term>,<term\_weight>. Do not type any spaces between the string, comma, or the weight.



---

**Hint:** When you specify a threshold, enter a weight for each term or an error message is returned in the Category Syntax Check window.

---

5. Select **Build --> Build Rule-Based Categorizer**.

---

## 10.7 Specifying the Match Ratio

### 10.7.1 How Match Ratio Works

It is not necessary for each term, or string of words, to appear in every document that is a member of a linguistic rule. However, a certain percentage of the unique terms that comprise the category rule are required or the document might not be correctly categorized.

Use the **Match Ratio** field in the **Data** tab to specify the percentage of terms that qualify the document as a category member. The default setting is 10%.

The match ratio setting works with linguistic rules. Match ratio is also affected by special symbols, specifically \*, \*\*, --, and +. For more information, see Section 10.8 *Selecting Special Symbols* on page 278.

### 10.7.2 Optimizing the Match Ratio Setting

Optimize the match ratio by making any of the following changes:

#### Match ratio percentage

In general, the larger the percentage of terms to be matched in the category rule, the narrower the rule is. If you specify a percentage that is lower than the 10% default setting, the category rule is wider and duplicate category membership becomes more probable. For example, 5% enables a match on fewer terms to return a match for the category. For more information, see Section 10.7 *Specifying the Match Ratio* on page 276.

#### Number of terms in the Category rule

The number of terms in the category rule can be changed to redefine the breadth of category membership. For example, if a category rule consists of 20 terms and the match ratio setting is 10%, two of the 20 terms make the input document a match. If, instead, you define a rule with 10 terms, a match on one term returns a match on the input document.

---

To understand the relationship between the number of terms and the match ratio setting as they apply to incoming documents, see the following table:

Table 10-1: Match Ratio and Category Membership

Terms	Match Ratio Percentage	Category Rule	Category Membership
More	Larger	Broader	Multiple
Fewer	Smaller	Narrower	Fewer
More	Small	Broad	Several possible
Fewer	Large	Narrow	Few

### Special symbols

Special symbols affect the Boolean syntax that replaces linguistic rules internally when these rules are exported.

In some cases these special symbols override the match ratio setting. For this reason, you should carefully consider the terms that uniquely include or exclude documents from category membership. For more information, see Section 10.8 *Selecting Special Symbols* on page 278.

### Category Bias setting

Frequency-based relevancy scores for categories that are defined by one term only, can be boosted when you use this setting. This specification affects matching and for this reason, this setting could override the results obtained by the match ratio setting. Alternatively, you could specify the **Relevancy Bias** setting in the **Data** tab.

---

## 10.8 Selecting Special Symbols

Qualify a linguistic rule with special symbols, described in the table below, for the following purposes:

- Apply stemming.
- Expand the word forms that are matched.
- Determine what terms make a document a match for the specified category.
- Ensure that a match on the rule is *not* a match for the category that it defines.

Special symbols for linguistic and Boolean rules differ. Some special symbols affect the match ratio setting and relevancy. For more information, see Table 11-3 on page 303.

Table 10-2: Special Symbols Used in Linguistic Rules

Symbol	Type	Description
@	Suffix	Apply stemming to the word that precedes this symbol to expand the category rule so that it includes all forms of this word. For example, specify <code>price@</code> and the category rule expands to include <i>price</i> , <i>prices</i> , and <i>pricing</i> . The word, as well as all of its variants, count once if there is a match toward the match ratio. After the match ratio setting is met, each instance of a matching term and each stemming match count once toward frequency-based relevancy.
@N	Suffix	Expand the category to include all of the noun forms of the word that precedes this symbol. If the preceding word is not a noun, no stemming is applied. The word, as well as each of its matched variants, count once toward the match ratio specification and once toward frequency-based relevancy, after the match ratio is met.
@V	Suffix	Expand the category to include all of the verb forms of the word that precedes this symbol. If this term is not a verb, no stemming is applied. The word, as well as all of its variants, count once if there is a match toward the match ratio. Each word and stemming instance also count once toward frequency-based relevancy. This is true only after the match ratio is met.

Table 10-2: Special Symbols Used in Linguistic Rules

*	Prefix	Assign this term more classificatory weight (more relevancy) than other, unmarked words in the list. The single asterisk counts <i>twice</i> toward the match ratio, but only once toward relevancy.  This example uses a match ratio setting of 20%. If the term that is prefixed by * is matched, this term is worth 50% (10% of the 20% necessary) of the match ratio. It is then multiplied by 2, or 20%.
**	Prefix	Counts four times toward the match ratio, but only once toward relevancy.  Continue with the example of a match ratio setting of 20%. If the term that is prefixed by * is matched, the matching term is worth 50% (10% of the 20% necessary) of the match ratio multiplied by 4. 40% is double the 20% requirement for the match ratio setting.
-	Prefix	Counts against the match.
_L	Suffix	Use the underscore character (_) followed by an uppercase L to represent a literal. Append the at sign (@) to the end of a word, and the word is not expanded because it is treated as a literal.
_C	Suffix	Override case-insensitive using case-sensitive matching.
--	Prefix	Augment the single hyphen (-) symbol. The presence of these symbols causes the rule <i>not</i> to match. In other words, when this term is present and the match ratio setting is met, there is no category match for this document. (Frequency-based relevancy is irrelevant in this case.)
+	Prefix	Use this symbol to prevent a match if this term is not present in the document. This symbol also suppresses stemming, overrides the match ratio setting, and counts once toward frequency-based relevancy.
!	Suffix	Select <b>Expand all word forms</b> in the <b>Category</b> tab of the Project Settings window. All of the words in the category rule, except those that are followed by an exclamation point, are stemmed.

The following rule provides an example of the use of special symbols:

*Example 10-2: Qualified Linguistic Rules*

```
travel@
*vacations
**hotel
```

---

```
boat@N
+reservations
airplane_C
rental cars_L
beach!
-spa
walk@V
--wedding
```

The terms that define the category rule have various levels of matching weight. If these terms are located in incoming documents, they affect the category matches:

- All forms of the word *travel* count toward the match ratio.
- A match on the word *vacations* counts twice toward the match ratio and once toward relevancy.
- If the word *hotel* is present in the document, it counts four times toward the match ratio, but only once toward relevancy.
- All expanded noun forms of the word *boat*, count once toward the match ratio and once toward relevancy.
- Unless the word *reservations* is present in the document, it is not a category match. Stemming is suppressed, the match ratio is overridden, and a match on this term counts once toward frequency-based relevancy.
- The word *airplane* is a match if it is located in matching case.
- The term *rental cars* is a match, but any other forms of the word do not match.
- The word *beach* is *not* stemmed. Stemming is suppressed even if you select **Expand all word forms** in the **Category** tab.
- The presence of the term *spa* means that this document might not be a good match for this category. If the number of matching terms equals the match ratio setting, this document is disqualified as a match.
- All of the verb forms of the word *walk*, located in an input document, count toward both the match ratio and frequency-based relevancy.
- The term *wedding* cannot appear in any of the documents that match this category.

To test these special symbols, see *Part 2: Testing*.

---

## 10.9 Create Symbolic Links

As part of rule development for categories that use linguistic terms, you can choose to use symbolic links that are pointers to other categories. Documents are categorized into the source category, because they match its rule.

Use symbolic links when it is beneficial for multiple (target) category rules to reference one (source) category rule. For example, use the terms that define the category `Team` as part of the rules for the categories Baseball, Football, and Soccer. Categorize the matching documents under the Baseball, Football, and Soccer categories. For more information, see Section 8.8 *Create Symbolic Links* on page 226.

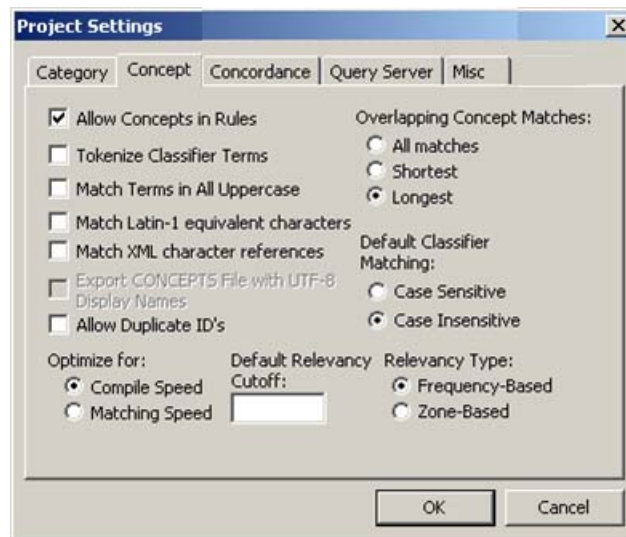
## 10.10 Define Dependencies

Dependencies enable a linguistic rule to use an entire classifier definition as part of its rule. This operation saves you time, and assures a greater degree of accuracy. When you edit the classifier concept terms, you also edit the dependent linguistic rule. For more information, see Section 8.9 *Creating Dependencies* on page 230.

To create dependencies between categories defined by linguistic rules and classifier concepts, complete these steps:

1. Specify a linguistic rule to define a category.
2. When you define the classifier concept, include the terms that also apply to the category rule. For more information, see Section 18.2 *Writing a Classifier Definition*.
3. Select **Project --> Settings**.

- 
- Click the **Concept** tab.



- Select **Allow Concepts in Rules**.

---

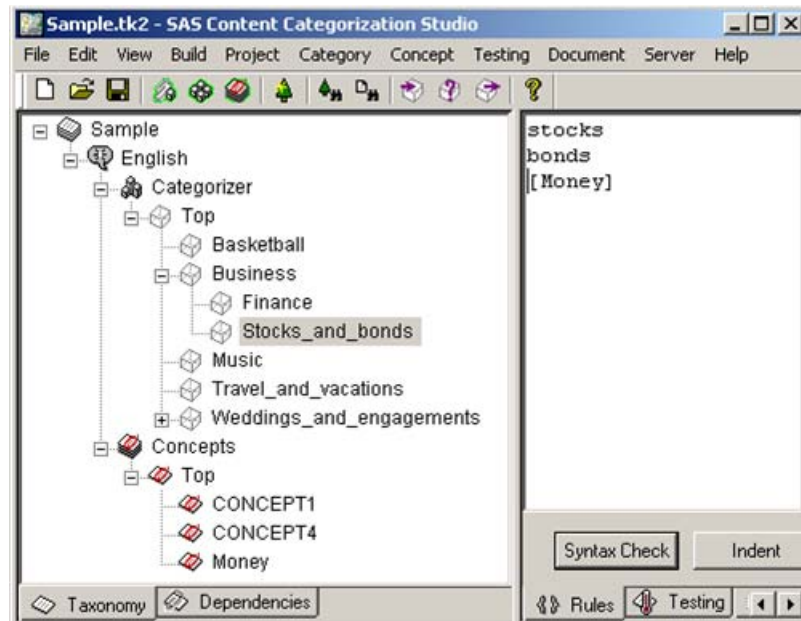
**Note:** If you do not enable **Allow Concepts in Rules**, unexpected behaviors might occur.

---

- Select **Case Insensitive**.
- Click **OK**.
- Select **Build --> Compile Concepts**.



9. Click the **Rules** tab and type the name of the classifier concept, surrounded by brackets ([ ]) on a separate line.



10. Select **Build --> Build Rulebased Categorizer**.
11. Test the rule. For more information, see Section 13.4 *About Batch Testing* on page 379.



---

# *Chapter: 11*

## Rule-Based Categorizer: Boolean Terms

---

- *Overview of Boolean Rules*
- *Benefits of Boolean Rules*
- *Quick Start Guide*
- *About Category Membership*
- *Benefits of Modes*
- *About Boolean Operators*
- *Specifying Special Symbols*
- *Specifying Structured Text Field Names*
- *Editing Rules*
- *Automating Parent and Child Rule Generation*
- *Defining Symbolic Links*
- *Dependencies between Categories or Categories and Classifier Concepts*
- *Quick Start Guide to Testing Boolean Rules*
- *Query an Index*

---

## 11.1 Overview of Boolean Rules

Boolean operators, and other available modifiers, make Boolean rules the most precise rules with the highest recall. Boolean operators qualify linguistic terms to specify location, distance, whether the presence of a term determines a match, and so on. You can add these operators to an existing linguistic rule to make it more precise.

---

**Note:** All linguistic rules become Boolean rules before they are tested. This is an internal operation. However, you can make this process more precise when you specify your own operators.

---

You can also modify Boolean rules by adding special symbols to the specified list of terms. Most of these special symbols can also be used with linguistic rules. Due to the precision capabilities of Boolean rules, the **Match Ratio** field in the **Data** tab does not apply to these rules. Although you can specify categories that use Boolean or linguistic rules in a single taxonomy, you cannot mix these rules in one category definition. Neither can you create dependencies between linguistic and Boolean rules.

## 11.2 Benefits of Boolean Rules

When you choose to use the Boolean rule-based categorizer you gain the benefits of the precision capabilities that are unique to Boolean operators.

In particular, you gain the following benefits:

- Use Boolean operators to add precision to the linguistic terms that form the basis of these rules.
- Use the at sign (@) to expand a word form in a Boolean rule into all of its word forms, or limit the expansion to noun and verb forms.
- Use structured-text fields to specify where a term is located in an input document in order to return a match. This is ideal for XML documents. Use this setting carefully when text documents are categorized.

- 
- Click the **Tree View** radio button in the **Rules** tab to automatically indent a Boolean category rule. It is easy to see and indent this rule, because each rule segment follows a Boolean operator.
  - Click **Indent** if you select the **Text View** mode in the **Rules** tab. Use this operation to see the text of a Boolean rule as it is separated by the parentheses ( ) that surround each Boolean operator and the terms that the operator qualifies.
  - Create dependencies, not only between categories and classifier concepts, but also between Boolean categories.

---

**Note:** If you define dependencies that reference concepts, and plan to use the server query operation with your index, write your concept rules in lowercase letters. The server query operation is not case sensitive at this time.

---

- You can use the paste macro command to simplify the task of creating dependencies.
- Like linguistic rules, you can also create symbolic links between categories that enable you to write a category rule once and use it multiple times in the taxonomy. All of the rule matches are returned as members of the source rule category.
- You can use the **Dependencies** tab to check forward and reverse dependencies between categories before you delete a category.

---

## 11.3 Quick Start Guide

To build and deploy the Boolean categorizer, complete these steps:

1. Specify installation-specific operations using the Options window:

### Testing

Select one of these operations:

**Always save before each test**

**Always rebuild before each test**

**Show best matches when testing all**

**Sort taxonomies automatically**

### Taxonomy as Text

Select one of these operations:

**Flag categories/concepts with no definitions**

Select this check box to locate rules that have not been defined.

**Flag categories/concepts with no dependencies**

Select this check box if you wrote rules that are dependent on other rules.

### Syntax Checking

Select the appropriate operations if you create dependencies between Boolean rules and classifier concepts.

2. Specify project-specific settings in the Project Settings - Category window:

### Default Relevancy Bias

Specify the minimum relevancy that is required for a document to be a match for a category.

### Relevancy Type

Specify the operation that is used to compute the best matches.

### Allow Short Macro Names

Use the short version of the path to the selected category when you define dependencies. For more information, see Section 11.12 *Dependencies between Categories or Categories and Classifier Concepts* on page 337.

---

## Boolean Morphological Expansion

Select one of these operations:

### Never expand word forms

(Default) Prevent word expansions. This is also true if the at sign (@) is appended to a word in the rule.

### Expand word forms with '@' sign

Automatically expand all of the words that have an appended at sign (@).

### Expand all word forms

Automatically expand the words in the rule into all of the word forms even if there is no appended at sign (@).

## Export Short MCO File

Produce a \*.short.mco file. This is a categorization binary file where the category names that are returned are the short names, instead of the full pathnames.

Enable **Duplicate ID's** if you want two or more categories to share the same identification number.

3. (Optional) If you are creating dependencies between Boolean categories and classifier concepts, select **Allow Concepts in rules** in the Project Settings - Concept window.
4. Specify project-specific settings using the Project Settings - Misc window:

### Compatibility Date


Enter the date of the older version of SAS Content Categorization Server that you are running. SAS Content Categorization Studio generates a binary file (.mco or .concepts) that is compatible with the older version. Use this operation until you have time to install and run a newer version of SAS Content Categorization Server.

### Use UTF-8 Test Files

Select this check box if your test files are in UTF-8 format and the language of the categorizer is not in UTF-8 format.

---

## Directory for Unmatched Populate Files

Click  to set the location for this file and SAS Content Categorization Studio places all of the unmatched testing files here.

5. Select a category in the **Taxonomy** tab and click the **Rules** tab to access the Rules window in the default Text View mode.
6. Enter a list of words that uniquely identify the selected category. For more information, see Section 10.5.2 *Write a Linguistic Rule* on page 271.
7. Use Boolean operators to modify the specified terms in the Rules window. For more information, see Section 11.6 *About Boolean Operators* on page 294.
8. Consider the weights of the various Boolean operators that are in your rules. For more information, see Section 9.2.4 *How Boolean Operators Affect Relevancy Weights* on page 251.
9. Add special symbols. For more information, see Section 11.7 *Specifying Special Symbols* on page 303.
10. Append suffixes to your rule terms as necessary. For more information, see Section 11.7.3 *Appending Suffixes* on page 307.
11. Check the grammar of your Boolean rule and if necessary, rebuild the categorizer. For more information, see Section 8.6 *Check the Syntax of a Boolean Rule* on page 224.
12. Choose to expand the word forms of any of the unique linguistic terms that define your category rule. For more information, see Section 11.9.4 *Expand Word Forms* on page 331.
13. Select appropriate structured-text fields if you are categorizing Web-based documents. For more information, see Section 11.8.2 *How to Specify a Structured Text Field* on page 309.
14. Create symbolic links where the *target* category uses the whole rule of the *source* category. For more information, see Section 11.11 *Defining Symbolic Links* on page 336.
15. Define dependencies. For more information, see Section 11.12 *Dependencies between Categories or Categories and Classifier Concepts* on page 337.



- 
16. Test the rule. For more information, see Section 11.13 *Quick Start Guide to Testing Boolean Rules* on page 341 and *Part 2: Testing*.
  17. Edit the rule. For more information, see Section 11.9 *Editing Rules* on page 322.
  18. Build the categorizer and save the project, now, if you have not performed these operations earlier.
  19. Upload the categorizer. For more information, see Section 2.10 *Upload the Categorizer or Concepts to SAS Content Categorization Server* on page 138.

## 11.4 About Category Membership

Category membership for a Boolean category works in many of the same ways that category membership works for a linguistic category rule. However, there are differences. Boolean category rules are more precise because they use Boolean operators to define the relationships that are necessary for matches.

Effective Boolean rules have the following qualities:

### **Accurate**

Boolean rules use Boolean operators and other modifiers to precisely determine category membership. These operators and modifiers qualify the terms to be located, define the location of matched terms, and specify whether stemming is performed.

### **Adequate**

Boolean rules should be sufficiently broad.

### **Appropriately limited**

While inclusive, the Boolean rule for one category should not exceed its appropriate boundaries by overlapping other category rules in the taxonomy.

In order to include appropriate documents, while excluding inappropriate texts, SAS Content Categorization Studio analyzes all of the category rules in the taxonomy. As you define each category, it is important to consider the entire taxonomy in order to avoid creating categories that are either too broad or narrow.

---

To simplify the complexity of writing Boolean rules, SAS Content Categorization Studio offers you three ways to see the category rule:

**Text View**

Write your Boolean rules across a single line. This view works well for short Boolean category rules.

**Indent**

Realign a lengthily rule according to the Boolean operators that qualify the linguistic terms.

**Tree View**

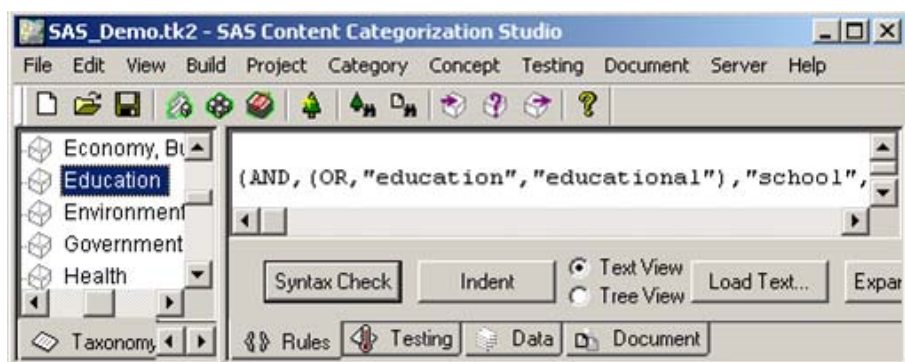
Display the Boolean rule like a taxonomy tree. Use this mode to see the selected category rule segment separated by Boolean operators. Each term in the rule also appears on a separate line similar to the list style of linguistic rules.

For more information, see Section 11.5 *Benefits of Modes* below:

## 11.5 Benefits of Modes

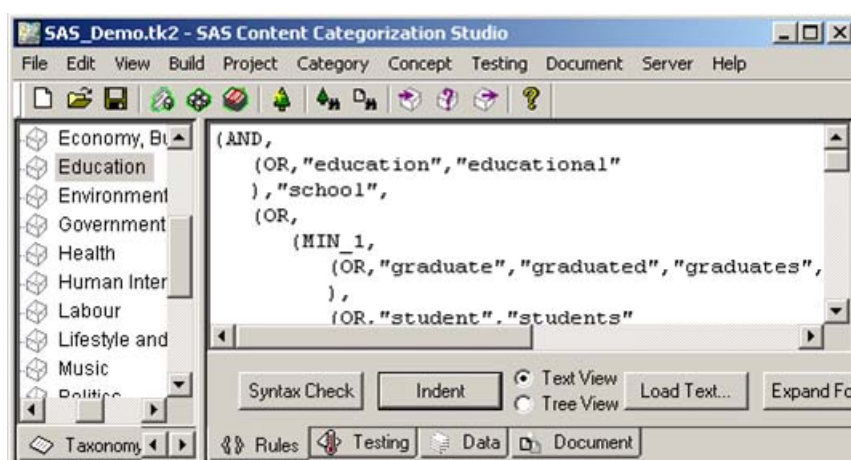
Write Boolean rules on a single line. Specify a sequence of Boolean operators that modify the unique linguistic terms in the Text View mode of the Rules window.

*Display 11-1 Text View Mode*



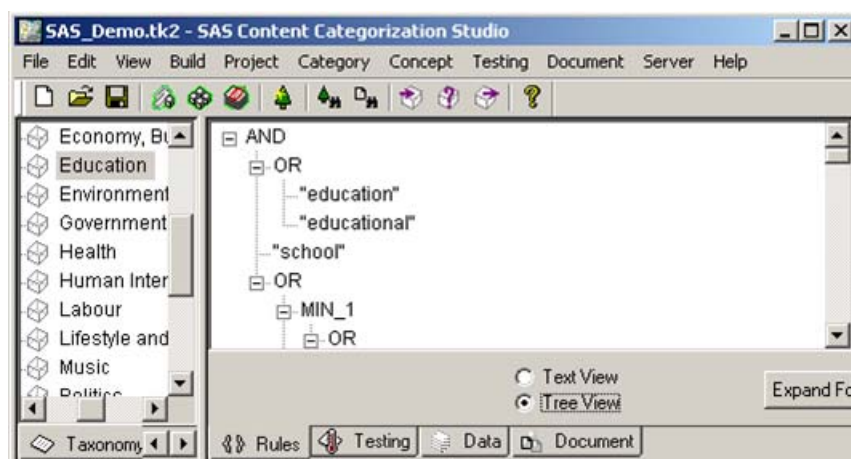
Click **Indent** in the Rules window to see the rule segments according to parentheses. Edit the rule using this view.

*Display 11-2 Indented Rules*



Click **Tree View** to view the rule, separated by Boolean operators and linguistic terms, in a tree layout.

*Display 11-3 Tree View Mode*



The Tree View mode simplifies seeing and editing the rule. The parentheses ( ) are not displayed. Click the plus (+) and minus (-) operators to the left of

---

the Boolean operators to expand and contract the nodes in the tree in order to see sections of the rule.

## 11.6 About Boolean Operators

### 11.6.1 Overview of Boolean Operators

This section provides an overview and examples of the Boolean operators that you use to develop Boolean rules. The operators in the two tables below are linked to the comprehensive descriptions and examples in the sections that follow.

---

**Note:** Boolean operators are case sensitive.

---

Table 11-1: Common Boolean Operators

Operator	Description
<u>AND</u>	Takes two or more arguments. True only if all of the arguments are true.
<u>OR</u>	Takes two or more arguments. True if at least one argument is true.
<u>NOT</u>	Takes one argument. Use with the AND operator. True if the argument is false.
<b>Note:</b> The NOT operator is always used in the context of an AND, and not an OR operator.	
<u>MIN_n</u>	Takes one or more arguments. True if at least <i>n</i> arguments are true.
<u>MINOC_n</u>	Takes one or more arguments. True if the total number of occurrences of the arguments in the document is at least <i>n</i> .
<u>MAXOC_n</u>	Takes one or more arguments. True if the total number of occurrences of the arguments in the text is no more than <i>n</i> .
<u>SENT</u>	Takes two or more arguments. True if all of the arguments occur in the same sentence.

Table 11-1: Common Boolean Operators (Continued)

<u>PAR</u>	Takes two or more arguments. True if all the arguments occur in the same paragraph.
<u>DIST_n</u>	Takes two arguments. True if the first word appears in a sentence and the second word does not appear in the same sentence.
<u>ORD</u>	Takes two or more arguments. True if all of the arguments occur in the order specified by the rule.

The situational Boolean operators are listed below:

Table 11-2: Situational Boolean Operators

Operator	Description
<u>NOTIN</u>	Takes two arguments. True if the first argument occurs outside of the second argument. For example, use NOTIN, "health", "health care".
<u>NOTDIST_n</u>	Takes two arguments. True if both word strings in the argument are not within <i>n</i> words of each other.
<u>NOTINSENT</u>	Takes two arguments. True if all of the arguments appear in the same document, but not if they occur in the same sentence.
<u>NOTINPAR</u>	Takes two arguments. True if all of the arguments appear in the same document, but not if they occur in the same paragraph.
<u>START_n</u>	Takes one argument. True if the argument is matched within <i>n</i> words of the start of the document field.
<u>END_n</u>	Takes one argument. True if the argument is matched within <i>n</i> words from the end of the document.
<u>ORDDIST_n</u>	Takes two or more arguments. True if both arguments occur in the same order specified by the rule and if both occur within a distance of <i>n</i> words to each other.
<u>MAXPAR_n</u>	Takes one or more arguments. True if all arguments appear within the first <i>n</i> paragraphs.
<u>MAXSENT_n</u>	Takes one or more arguments. True if all arguments appear within the first <i>n</i> sentences.
<u>PARPOS_n</u>	Takes one or more arguments. True if all terms appear in the <i>n</i> <sup>th</sup> paragraph of the document.

---

## 11.6.2 Common Boolean Operators

### 11.6.2.A The AND Operator

The `AND` operator is used for two or more arguments and requires each argument to be present in order for the rule to be true. For example, the following rule requires the words *columbia* and *film* to both be in a matched document:

```
(AND, "columbia", "film")
```

### 11.6.2.B The OR Operator

The `OR` operator takes any number of arguments. It returns a match if any of its arguments are true. In the example below, a match occurs only if *music* or *piano* is present within a document:

```
(OR, "music", "piano")
```

When combined with other operators, such as `AND`, the `OR` operator can create complex rules. For example, the following Boolean rule returns a match if the body of the document contains the word *symphony*. If this is true and if either the term *orchestra* occurs in the body section or *music* occurs in the `title` field, a match occurs.

```
(AND, _body: "symphony", (OR, _body: "orchestra",  
                           _title: "music"))
```

### 11.6.2.C The NOT Operator

The `NOT` operator takes exactly one argument and is present with the `AND` operator. This argument is true if the words specified with the `AND` operator are located, but the term preceded by the `NOT` operator is *not* located. For example, the following rule requires the words *music* and *piano*, if the word *flute* is not located, to be present in a matched document:

```
(AND, (OR, "music", "piano"), (NOT, "flute"))
```

The `NOT` operator can also be used to negate other operators. For example, the following rule specifies that if both *columbia* and *film* are present, the text is not a match:

```
(NOT, (AND, "columbia", "film"))
```

---

### 11.6.2.D The MIN<sub>n</sub> Operator

The MIN<sub>n</sub> (minimum) operator uses a number (*n*) as a parameter and takes any number of arguments. The rule is true if at least *n* of the elements in the rule are true. For example, this rule requires at least two of the three specified word strings to be matched:

```
(MIN_2, "hollywood", "columbia", "movie")
```

If only one of the words in the above example is located in a document, then MIN<sub>n</sub> returns false.

### 11.6.2.E The MINOC<sub>n</sub> Operator

The MINOC<sub>n</sub> (minimum occurrences) operator is similar to MIN<sub>n</sub>, except that the number *n* refers to the minimum number of occurrences in order for there to be a match.

For example, the following rule returns true if *Hollywood* and *Columbia* each occur once in the document (a total of two occurrences). It would also return true if *Hollywood* occurs two times in the document, but *Columbia* and *movie* never occur (also a total of two occurrences).

```
(MINOC_2, "hollywood", "columbia", "movie")
```

### 11.6.2.F The MAXOC<sub>n</sub> Operator

The MAXOC<sub>n</sub> (maximum occurrences) operator is the opposite of the MINOC<sub>n</sub> operator. The number *n* refers to the maximum number of matches that can be located for the arguments that this operator takes.

---

**Note:** MAXOC is useful for filtering out spam documents. In particular, those texts that repeat keywords to boost their ranking, and documents that are too general for a particular domain.

---

For example, the following rule returns true if *Hollywood* and *Columbia* each occur once in the document (a total of two occurrences). It would also return true if *Hollywood* occurs two times in the document, but *Columbia* and *movie* are not located (also a total of two occurrences).

```
(MAXOC_2, "hollywood", "columbia", "movie")
```

---

If *Hollywood* and *Columbia* each occur once in the document (a total of two occurrences) and *movie* occurs twice (making the sum total four) no match occurs.

### 11.6.2.G The SENT Operator

The `SENT` (sentence) operator takes any number of arguments and is true if all of the terms occur within the same sentence.

For example, this category rule returns a match only if the `body` field has a sentence with the words *fiscal*, *earnings*, and *rose* in it:

```
(SENT,_body:"fiscal",_body:"earnings",_body:"rose")
```

### 11.6.2.H The PAR Operator

The `PAR` (paragraph) operator takes any number of arguments and is true if all of the elements occur within a single paragraph.

For example, this rule returns a match only if the document has a paragraph with the words *representative*, *government*, and *announced* in it:

```
(PAR,"representative","government","announced")
```

---

**Note:** Specify one, or more, paragraph delimiters into the **Paragraph Separator** field of the Project Settings - Misc window. This operation enables you to use paragraph operators such as `PAR`, `MAXPAR`, `PARPOS`. For example, enter `"\n\n"`, `"<p>"` into the **Paragraph Separator** field.

---

### 11.6.2.I The DIST\_n Operator

The `DIST_n` (distance) operator uses a number (*n*) as a parameter and takes two arguments. The rule is true if both word strings in the argument are *within* *n* words to each other.

For example, the following rule is true only if the words *mutual* and *fund* occur within 10 words of each other. If these terms are instead within 11 words of each other, the rule does not return a match:

```
(DIST_10,"mutual","fund")
```



---

### 11.6.2.J The ORD Operator

The `ORD` (order) operator takes any number of arguments. It is true if all of the elements occur in the same order specified in the rule.

For example, the following rule returns a match only if the `body` field has the words *rates*, *insurance*, and *industry* in that order. The words do not have to be sequential, but they do have to occur in the prescribed order.

```
(ORD,_body:"rates",_body:"insurance",_body:"industry")
```

## 11.6.3 Situational Boolean Operators

### 11.6.3.A The NOTIN Operator

The `NOTIN` operator takes two arguments and is true if the string in the first argument appears outside the string in the second argument.

For example, consider the following rule:

```
(NOTIN,"a","a b")
```

This rule could return these values:

- `x x a x x`: True because *a* is not in the context of *a b*.
- `x x a b x`: False because *a* is in the context of *a b*.
- `x x a b a x`: True because the second *a* is not in the context of *a b*.

The following rule is true if the word *rock* is found in a string that is not *rock and roll*:

```
(NOTIN,"rock","rock and roll")
```

The matches could include *rock garden* and *a rock among the leaves*. In addition, the following sentence would also be a match because the first occurrence of *rock* is outside of the *Rock and Roll* phrase:

```
He saw a big rock near the Rock and Roll Hall of Fame.
```

### 11.6.3.B The NOTINDIST\_n Operator

The `NOTDIST_n` operator, a combination of the `NOT` and `DIST` operators, uses a number (*n*) as a parameter and takes two arguments. This operator is true only if the first element occurs within the specified distance and the other does not. The other argument does not need to appear in the document.

---

For example, the following rule is true only if the document has the words *black* and *white* that are not within three words of each other:

```
(NOTINDIST_3,"black","white")
```

### 11.6.3.C The NOTINSENT Operator

The NOTINSENT (not in sentence) operator takes two or more arguments and is true only if both arguments appear in the same sentence.

For example, the following rule returns a match only if the `body` field does not have a sentence with the words *fiscal* and *earnings* in it:

```
(NOTINSENT,_body:"fiscal",_body:"earnings")
```

### 11.6.3.D The NOTINPAR Operator

The NOTINPAR (not in paragraph) operator takes two arguments and is true only if both arguments are matched in the same paragraph. None of the other arguments are required to appear in the same document.

For example, this rule is true only if the document does not have a paragraph with the word strings *free agent* and *baseball*:

```
(NOTINPAR,"free agent","baseball")
```

---

**Note:** In order to use this operator, specify a paragraph delimiter. For more information, see Section 2.9.3 *The Misc(ellaneous) Tab for Categories and Concepts* on page 54.

---

### 11.6.3.E The START\_n Operator

The START\_*n* operator uses a number (*n*) as a parameter and takes one argument. The rule is true if the word string is found within *n* words from the start of the document field. For unstructured documents (not HTML, SGML, or XML texts) this number refers to the start of the document.

For example, the following rule is true only if the string *computer* is found within the first 20 words of the `body` field:

```
(START_20,_body:"computer")
```

---

**Note:** The `START` and `END` operators are useful when the structure of the input texts is homogenous and known. For example, academic research papers where the abstract is within the first 200 words and the references are within the last 300 words.

---

### 11.6.3.F The `END_n` Operator

The `END_n` operator uses a number ( $n$ ) as a parameter and takes one argument. The rule is true if the word string is found within  $n$  words from the end of the document field or, for unstructured documents, from the end of the text. (For more information, see the note above.)

For example, this rule is true only if the string *computer* is found within the last 20 words of the field:

```
(END_20,_body:"computer")
```

### 11.6.3.G The `ORDDIST_n` Operator

The `ORDDIST_n` operator is a combination of the `ORD` and `DIST` operators, uses a number ( $n$ ) as a parameter and takes two arguments. This operator is true if both elements occur in the same order that they appear in the rule and if both are within  $n$  words of each other.

For example, the following rule is true only if the document has the words *coach* and *team* in that order and within five words of each other:

```
(ORDDIST_5,"coach","team")
```

### 11.6.3.H The `MAXPAR_n` Operator

The `MAXPAR_n` (maximum paragraph) operator uses a number ( $n$ ) as a parameter and takes any number of arguments. A match occurs if all of the elements occur within the first  $n$  paragraphs of the document.

For example, the rule returns a match only if the document contains the words *representative*, *government*, and *announced* within the first three paragraphs of the document.

```
(MAXPAR_3,"representative","government","announced")
```

---

**Note:** Specify a paragraph delimiter in order to use this operator. For more information, see Section 2.9.3 *The Misc(ellaneous) Tab for Categories and Concepts* on page 54.

---

### 11.6.3.I The MAXSENT\_n Operator

The MAXSENT\_n (maximum sentence) operator uses a number (*n*) as a parameter and takes any number of arguments. A match occurs if all the elements occur within the first *n* sentences.

For example, the following rule returns a match only if the document contains the words *giants* and *baseball* within the first two sentences of the input text.

```
(MAXSENT_2, "giants", "baseball")
```

### 11.6.3.J The PARPOS\_n Operator

The PARPOS\_n (paragraph position) operator uses a number (*n*) as a parameter and takes any number of arguments. It is true if all the elements occur in the *n*<sup>th</sup> paragraph of the document.

For example, the following rule returns a match only if the document contains the words *representative*, *government*, and *announced* within the third paragraph.

```
(PARPOS_3, "representative", "government", "announced")
```

---

**Note:** Specify a paragraph delimiter in order to use this operator. For more information, see Section 2.9.3 *The Misc(ellaneous) Tab for Categories and Concepts* on page 54.

---

---

## 11.7 Specifying Special Symbols

### 11.7.1 Overview of Special Symbols

Modify your Boolean rules using some of the special symbols that are also used for linguistic rules. These symbols are explained in the table below:

Table 11-3: Special Symbols Used in Boolean Rules

Symbol	Type	Description
@	Suffix	Use the at sign (@) to apply stemming to the word that precedes this symbol. The Boolean category rule is expanded to include all of its word forms. See the examples that follow this table.
@N	Suffix	Use the at sign (@) followed by N to expand the category rule to include all of the noun forms of the word that precede this symbol. For example, if you specify <code>book@N</code> , the category rule is expanded to include <code>books</code> . <b>Note:</b> If the preceding word is not a noun, no stemming is applied.
@V	Suffix	Use the at sign (@) followed by V to expand a word into all of the verb forms of the word. For example, if you specify <code>run@V</code> , the category rule is expanded to include <code>ran</code> , <code>run</code> , <code>running</code> , and <code>runs</code> . <b>Note:</b> If the preceding word is not a verb, no stemming is applied.
*	Suffix	Append the single asterisk (*), which is a wildcard character, to the end of a word. The asterisk matches any characters at the end of the word. For example ( <code>OR</code> , <code>"not*"</code> ) matches <i>not</i> , <i>notebook</i> , <i>notice</i> , and <i>note</i> .
_L	Suffix	Use the underscore (_) and uppercase L together stand for a literal. This combination matches a literal without the meaning associated with either of these special symbols. For example, see the following rules:  ( <code>OR</code> , <code>"end\$"</code> ) match <i>end</i> at the end of the document  ( <code>OR</code> , <code>"end\$_L"</code> ) matches <i>end\$</i> , if it appears anywhere in the text.
_C	Suffix	Use the underscore (_) followed by the letter C to specify case-sensitive matching. For example ( <code>OR</code> , <code>"USA"</code> ) matches <i>USA</i> , <i>usa</i> , <i>Usa</i> , and so on, while ( <code>OR</code> , <code>"USA_C"</code> ) matches <i>USA</i> only.

Table 11-3: Special Symbols Used in Boolean Rules (Continued)

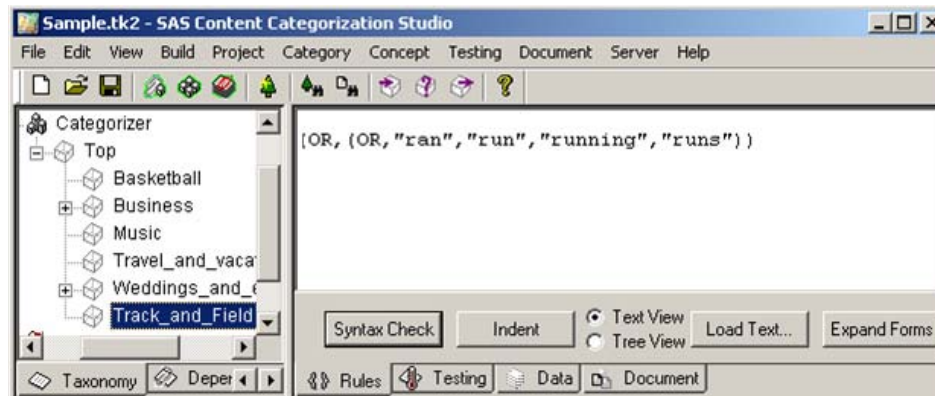
_Q	Suffix	Use the underscore (_) followed by the uppercase Q to specify that any matching instances of this term qualify the document to match the rule. These matches do not contribute to the relevancy score for the document.
_C_Q	Suffix	Use the suffix _C_Q (underscore [_] uppercase C followed by underscore uppercase Q) after a word. These characters indicate that the qualifying, case-sensitive match does not contribute to the relevancy score.
_L_Q	Suffix	Use the suffix _L_Q (underscore [_] uppercase L followed by underscore uppercase Q) after a term in a Boolean rule. This suffix qualifies a literal match. A match on this term makes the text a match for the category rule, but does not count when the relevancy score is computed.
\$	Suffix	Use the dollar sign (\$) to signal the end of a document. For example (OR, "The End\$") matches the string <i>The End</i> when the match occurs in the last string of the text. If the document contains the term \$19.99, this string can be matched as a literal. For example, this match is returned as a literal when (OR, "\$19.99") is specified.
!	Suffix	Use the exclamation point (!) to suppress stemming. If you select <b>Expand all word forms</b> in the <b>Category</b> tab, all of the words in the category rule, except those that are followed by an exclamation point, are stemmed.
<b>Note:</b> Most of these special symbols are also used for linguistic rules. For more information, see Table 10-2 on page 278.		

---

## 11.7.2 About Stemming

If you choose to expand a word by using the at (@) symbol, click **Expand Forms** in the Rules window to see the expanded rule form. For more information, see the example below and Section 11.9.4 *Expand Word Forms* on page 331.

*Display 11-4 Expand Forms Button*



---

**Note:** You can also expand your word forms, without appending an @ sign to the rules. Select **Expand all word forms** in the **Boolean Morphological Expansion** section of the Project Settings - Category window.

---

The words that you stem with the at sign (@) are replaced by the following syntax within SAS Content Categorization Studio. However, you cannot see this substitution, unless you click **Expand Forms**.

```
(OR, (OR, "original word", "word form1", "wordform2"))
```

---

**Note:** Use **Expand Forms** to see your rule terms, but return the rule to its original format before you test.

---

For example, use the following symbols to expand the word *train*:

**all word forms**

Append the at sign (@) to the word *train* in your category rule:

---

```
(OR,"train@")
```

The expanded rule that appears in the Rules window after you click **Expand Forms** is similar to the following example:

```
(OR,(OR,"train","trained","training","trains"))
```

#### **noun forms only**

Append the stemming symbol @N to the word *train*:

```
(OR,"train@N")
```

The expanded rule that appears in the Rules window after you click **Expand Forms** is similar to the following example:

```
(OR,(OR,"train","trains"))
```

#### **verb forms only**

Append the stemming symbol @v to the word *train* and also see the rule example shown below:

```
(OR,"train@V")
```

The expanded rule that appears in the Rules window after you click **Expand Forms** is similar to the following example:

```
(OR,(OR,"train","trained","training","trains"))
```

#### **use @ as a literal**

Append \_L after the at sign (@) to ensure that word expansion does not take place. See the example below:

```
(OR,"train@_L")
```

Click **Expand Forms** in the Rules window. A SAS Content Categorization Studio status window appears. The @\_L symbols determines that only a literal match on the word *train* is returned.

*Display 11-5 SAS Content Categorization Studio Status Window*





---

## 11.7.3 Appending Suffixes

### 11.7.3.A The Suffix \_C

The suffix \_C (underscore [\_] uppercase c) is used after a word in a Boolean rule. This term indicates that a match occurs only if there is also a match on the case of the term. For example, the acronym *WHO* is written into a category rule for the purposes of matching *World Health Organization*:

```
(OR, "WHO_C", "World Health Organization_C")
```

Using the example above, matches on the word *WHO* or *World Health Organization* are returned. If the suffix \_C was *not* appended to *WHO*, all instances of *who* in the input documents would also be matched.

### 11.7.3.B The Suffix \_L

The suffix \_L (underscore [\_] uppercase L) can be appended to a word. This string indicates that a preceding suffix, special symbol, and any other characters including whitespace characters are matched as literals. See the following example of a rule where the \_C in *version \_C* is specified as a literal match:

```
(AND, "version_A", "version_B", "version_C_L",  
      "version_D")
```

### 11.7.3.C The Suffix \_Q

The suffix \_Q (underscore [\_] uppercase Q) specifies that any matching instances of this qualifying term make the document a match for the rule. However, these instances do not contribute to the relevancy score for the text. See the following example:

```
(AND, "Orange County", "California_Q")
```

In this example, all matches for the words *Orange County* are counted toward the relevancy score, regardless of the string that follows. While any matches for *Orange County, California* are also matched, they do not contribute to the relevancy score for the document.

See the following section for more examples of how to use the suffix \_Q:

---

### 11.7.3.D The Suffix `_C_Q`

The suffix `_C_Q` (underscore [`_`] uppercase `c` followed by underscore capital `Q`) can be used after a word in a Boolean rule. This sequence indicates that the qualifying, case-sensitive match does *not* contribute to the relevancy score. See the following example:

```
top dog_C_Q
```

In this example, a match on the rule would be *top dog*, but not *Top Dog*.

### 11.7.3.E The Suffix `_L_Q`

The suffix `_L_Q` (underscore [`_`] uppercase `L` followed by underscore capital `Q`) can be used after a term in a Boolean rule to qualify a literal match. This means that the match does not count when the relevancy score is computed, although the match does make the document a match for the category rule. For example, a match on the following rule would be *version\_A*, *version\_B*, *version\_C*, *version\_D*, and so on:

```
(AND, "version_A", "version_B", "version_C_L_Q",  
      "version_D")
```

This match would not count toward the relevancy score, but it does permit other matches in the document to contribute to this score.

## 11.8 Specifying Structured Text Field Names

### 11.8.1 How to Use Structured Text Fields

Structured text is defined as Web pages where tags differentiate the various sections of HTML, SGML, and XML documents. For example, `<link>`, `<title>`, and `<description>`.

The text that defines these tags such as `link`, `title`, and `description`, cannot be matched. Term matches are located only within the specified sections. The default behavior for XML documents is that the sections that have the same tag names are conflated into one searchable section. By merging multiple sections of the same type, SAS Content Categorization Studio optimizes the matching function for rules that use Boolean operators such as `DIST` and `PAR`.

---

You can restrict term matching to one XML field when you specify `_field:"term"`. For example, specify `(OR,_title:"hurricane")` where the term *hurricane* is matched only if it occurs in the *title* field of an `.xml` document.

There are two specifications to use when you want to optimize Boolean rule matching for input documents at the local and project levels. You can specify project-wide settings in the Project Settings - Misc window. Alternatively, choose individual category matching by qualifying specific terms in category rules with section tags. However, the section tags that are specified in Boolean rules override the **Default Field** specification in the **Misc** tab.

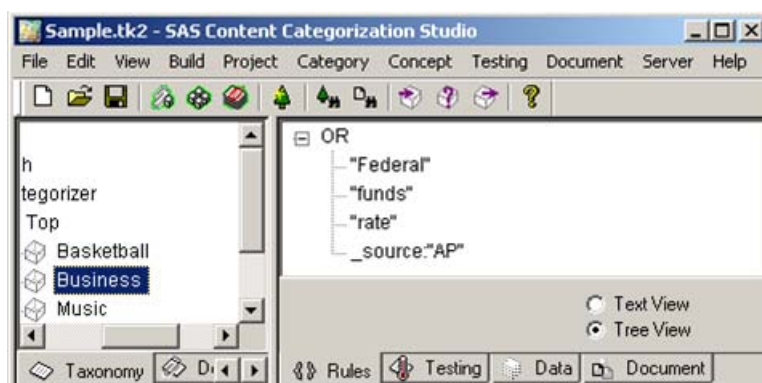
## 11.8.2 How to Specify a Structured Text Field

If you are using Boolean rules to categorize Web documents, you can specify a field to limit matches. For example, limit your searches to the `<description>` field. This specification, written within a rule, overrides any section specifications that you make in the Project Settings - Misc window. For more information, see Section 11.8.4 *How to Use Project Settings with Structured Text* on page 318.

For example, specify a term to be matched in the document `description` field. Field names, as specified in Boolean rules, are case sensitive and begin with an underscore when they are entered into the rule. For example, specify `_title`. The syntax is shown below:

```
_title:"unique linguistic term"
```

*Display 11-6 \_source Field Name*



---

See these fields in the abbreviated sample of an XML document:

Table 11-4: XML Field Examples

Field	Description
<title>	Specifies the name of the document.
<pubDate>	Specifies the date that the article was published.
<source>	Specifies the document originator.
<description>	Specifies a main section of the document.

*Example 11-1: An Uncategorized XML Document*

```
<?xml version="1.0" encoding="utf-8"?>
<title>Yahoo! News</title>
  <source>AP</source>
    <pubDate>Tue, 18 Sep 2007 16:54:07 GMT</pubDate>
      <description>AP - Federal Reserve policymakers began
their closed-door discussions Tuesday with investors
widely expecting the central bank will decide to cut its
target for the federal funds rate, the interest that
banks charge each other, for the first time in more than
for years.</description>
    <!-- server fe9.news.spl.yahoo.com --> </rdf:RDF>
```

For the sample document above, you could write the following Boolean rule:

```
(OR, "Federal", "funds", "rate", _source: "AP")
```

This Boolean rule specifies that when *Federal*, *funds*, or *rate* appears in the document *or* the word *AP* is present in the `<source>` field, a match occurs.

Before you can test a rule, use the **Misc** tab of the Project Settings window to set the searchable, and unsearchable, fields. For more information, see the following section.

## 11.8.3 How to Match Attributes in XML Fields

### 11.8.3.A Matching Attributes and Attribute Values

You can choose to match attribute fields and attribute values in XML documents. For example, choose to perform matching based on the existence of an attribute. You could also choose to return a match only on a document

---

where there is a match on the terms that are located in the specified attribute fields.

See the following attribute example and the explanation that follows:

```
<doc type="text" name="test.txt"/>
```

Table 11-5: Attribute Component

Attribute Component	Description
doc	Specifies the field name.
type and name	Specifies the attributes.
text	Specifies the value of the type attribute.
test.txt	Specifies the value of the name attribute.

Perform the following types of matching with attributes in XML documents:

- Match only if an attribute exists
- Match only if an attribute exists and the field text matches the rule term
- Match only if an attribute contains the specified value
- Match only if an attribute contains the specified value and the field text matches the rule term

When you write these rules use the following characters:

Table 11-6: Characters for Attribute Matching

Character	Description
\ (backslash)	Separates the fields and attributes.
:1 (colon and number one)	Means "if true."
_nameofattribute	One underscore character (_) precedes the name of the attribute.

---

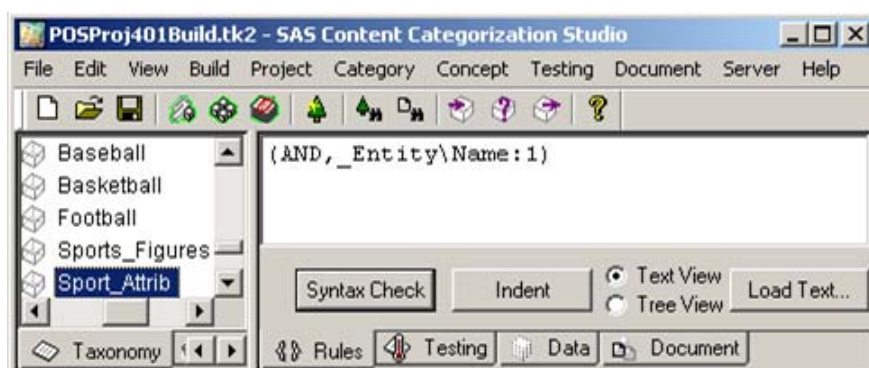
### 11.8.3.B Match Only If an Attribute Exists

If you choose to return matches when the attribute is present within the input document, the document is marked **PASS**. If the attribute is not present, the document is marked **FAIL**. There are no rule terms to match. For this reason, you do not see any matched terms in the Document window.

To return matches when an attribute is present in an input document, complete these steps:

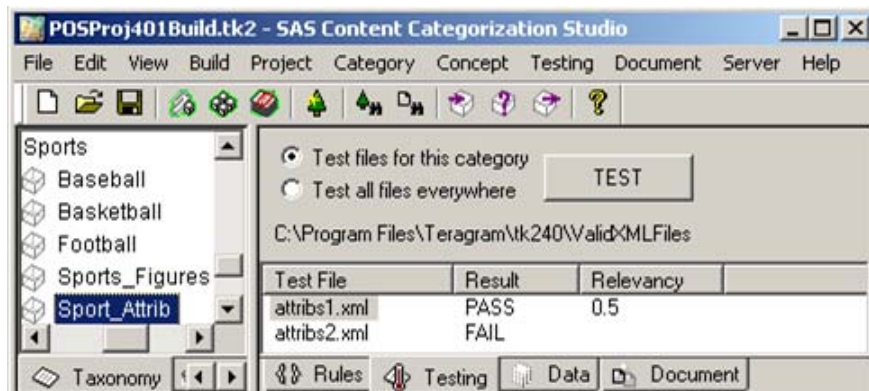
1. Enter a rule specifying an attribute in the **Rules** tab. For example, type:

```
(AND, _Entity\Name:1)
```



2. Click **Syntax Check**. (If the syntax is not OK, rewrite the rule, and check its syntax until the syntax is OK.)
3. Select **Build --> Build Rulebased Categorizer**.

4. Click **Testing**. (If you have not already specified the path to your XML testing files, see Chapter 12.)



5. Click **TEST**.
6. See the testing results in the Testing window. For example, see that the `attribs1.xml` document passed with a relevancy of 0.5.

---

**Note:** There are no matches displayed in the Document window for this type of rule match.

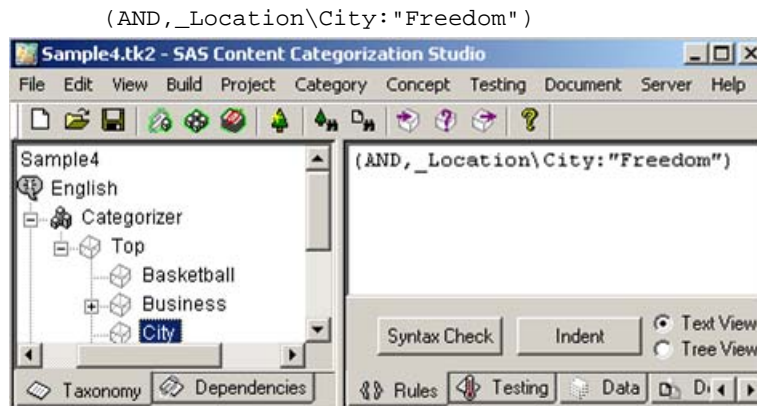
---

### 11.8.3.C Match If an Attribute Exists and the Field Text Matches the Rule Term

Use this type of rule to locate a term that appears in the specified XML field that contains the specified attribute. When you write this rule, you specify the term that you want to locate. A match is returned if this term is located within the document in the specified XML field and that XML field contains the specified attribute.

To write a rule that matches a specified term, complete the following steps:

1. Enter a rule specifying an attribute in the **Rules** tab. For example, type:

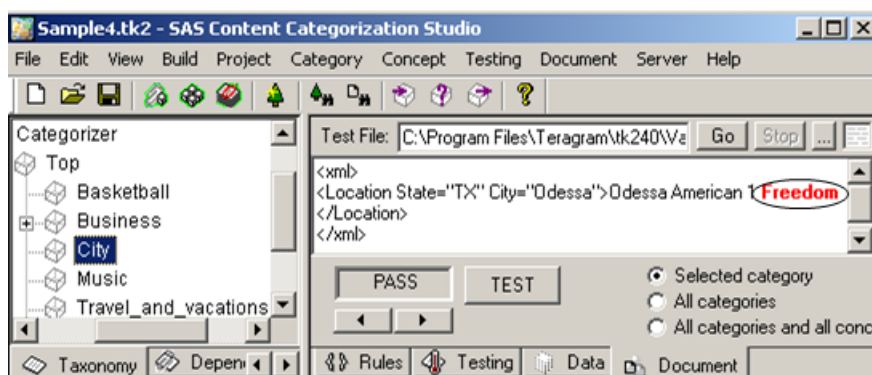


2. Complete Step 2 through Step 5 on page 313.





3. Double-click the passing file and see the matched text in the Document window. For example, double-click `attribs2.xml`.



4. See the matching term in the Document window. For example, see Freedom.

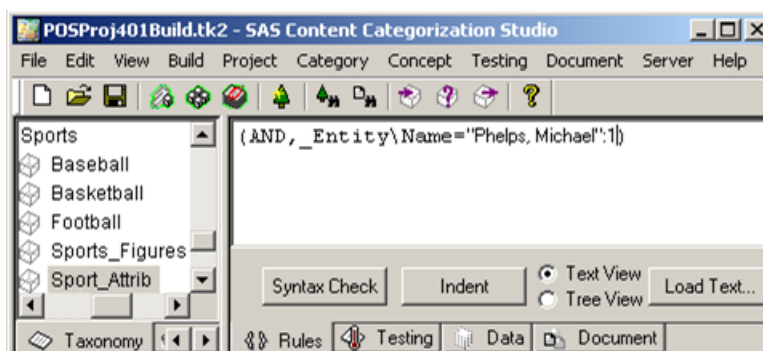
### 11.8.3.D Match Only If an Attribute Contains the Specified Value

Write a rule that matches an input XML document only if an attribute contains a specified value.

To write a rule that matches a specified term, complete the following steps:

1. Enter a rule specifying an attribute in the **Rules** tab. For example, type:

```
(AND, _Entity\Name="Phelps, Michael":1)
```



2. Complete Step 2 through Step 5 on page 313.

- 
3. See the following example of the testing results.



---

**Note:** There are no matches displayed in the Document window for this type of rule match.

---

#### 11.8.3.E Match Only If an Attribute Contains the Specified Value and the Rule Text Matches

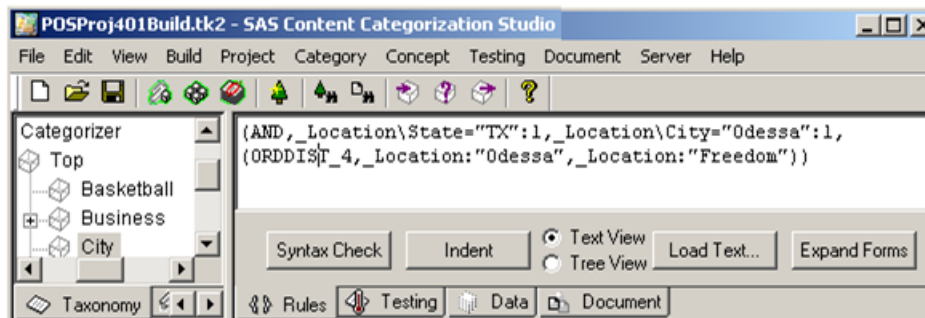
Write a rule to match text within an XML field that contains a specific attribute and value. In this case, you can also see the matched term highlighted in the Document window.

To return rule text matches within a specific XML field and for a specific attribute field value, complete these steps:

1. Enter a rule specifying an attribute in the **Rules** tab. For example, type:

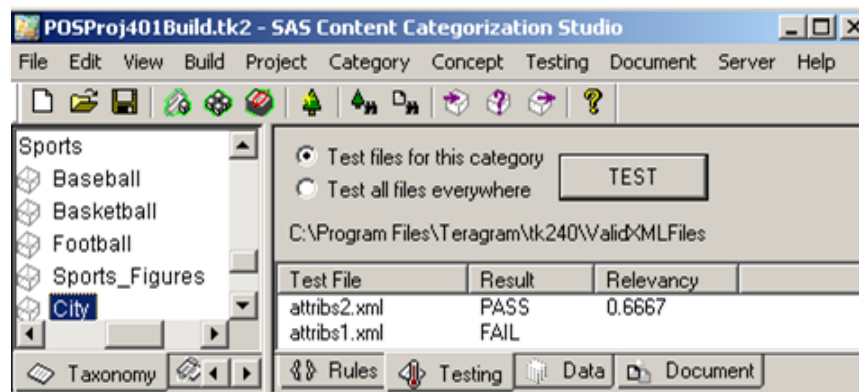
---

```
(AND,_Location\State="TX":1,_Location\City="Odessa":1, (
ORDDIST_4,_Location:"Odessa",_Location:"Freedom"))
```

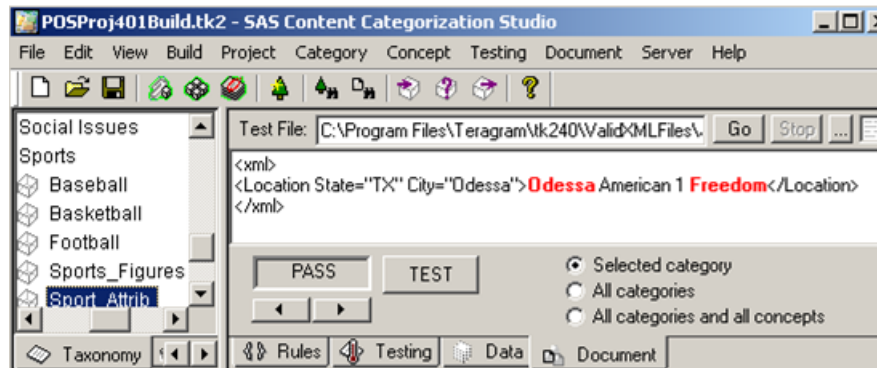


2. Complete Step 2 through Step 5 on page 313.

See the results under the **Result** and **Relevancy** column headings. For example, see that the `attribs1.xml` file passed with a relevancy value of 0.6667 for the `Sport_Attrib` category.



3. Double-click the passing file and the full text of the document appears in the Document window.



#### 11.8.4 How to Use Project Settings with Structured Text

To search XML documents, you can specify the default field that limits the locations where matches occur. Otherwise, SAS Content Categorization Studio treats the entire document except for the text that defines its tags as a searchable stream of text. In this case, a match can occur anywhere in the input document.

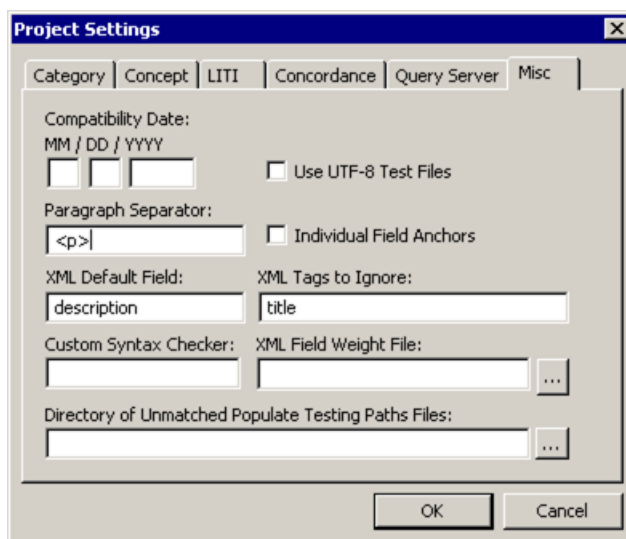
Project-level settings enable you to specify matching requirements across the taxonomy. These settings lack the specificity of the rule-writing process where you can also limit matches to selected fields.

Use the three settings in the **Misc** tab of the Project Settings window to specify the searchable and non-searchable structured text fields at the project level. These settings, **Individual Field Anchors**, **XML Default Field**, and **XML Tags to Ignore** enable you to determine the matching requirements across all of the categories in the taxonomy.

For more information, see the example provided in the display below and Section 2.9.3 *The Misc(ellaneous) Tab for Categories and Concepts* on page 54.

---

### Display 11-7 Project-wide Settings



To set the structured text matching specifications, complete these steps:

1. Select the **Individual Field Anchors** check box and SAS Content Categorization Studio maintains each section as a separately searchable field. (For more information, see Section 11.8.5 *Specifying the Caret and Dollar Symbols* on page 320.) When you make this selection, you limit the application of the `DIST` and `PAR` Boolean operators.

---

**Note:** Select **Individual Field Anchors** if you choose to use either the caret (^) or the dollar sign (\$) when you write Boolean category rules. For more information, see Section 11.8.5 *Specifying the Caret and Dollar Symbols* on page 320.

---

2. Specify **XML Default Field** the default search field for Boolean rules. For example, specify the `description` tag without any angle brackets (<>). You can specify multiple fields separated by commas (.). When you choose this setting at the project level, you restrict the rule matches to the `<description>` field for the applicable category rules.

- 
3. **XML Tags to Ignore:** Type in the names of any fields that SAS Content Categorization Studio ignores. For example, specify `title`. If you choose to specify multiple fields, use commas (,).

The settings selected in the **Misc** tab enable you to shorten the rule-writing process. For example, you can abbreviate the following category rule using these project settings:

```
(OR,_description:"Federal",_description:"funds",  
_description:"rate",_source:"AP")
```

This rule can be rewritten as follows when the **Default Field** in the **Misc** tab is specified as `description`:

```
(OR,"Federal","funds","rate",_source:"AP")
```

In this example SAS Content Categorization Studio performs the following operations:

- Search the `<description>` field for the words *Federal*, *funds*, or *film*.
- Search the `<source>` field, only, for the word *AP*.
- Ignore any possible matches in the `<title>` field of input XML documents.

### 11.8.5 Specifying the Caret and Dollar Symbols

Select the **Individual Field Anchors** check box in the Project Settings - Misc window. Choose this setting with Boolean category rules (with classifier concepts) that use disambiguation. By default, if you have more than one instance of an XML tag in a Web document, SAS Content Categorization Studio collapses the sections into one searchable area. When you select this check box, each section of a Web-based document is searched separately. This feature also enables you to specify the location of a matching term.

Use the caret (^) symbol to specify that a match occurs only on the first instance of a section that shares the same tag name as other document sections. If you want to specify that your matches can be returned only when they are located in the last instance of a field type, append the dollar (\$) sign.

For example, use a dollar sign with `body:"film$"` when there are several `<body>` fields in the input documents. In this example, SAS Content

Categorization Studio searches only the *last* <body> field. A match is not returned if this term is located in another field.

If you do not select **Individual Field Anchors** and you choose to use one or both of these symbols, unexpected behaviors might occur.

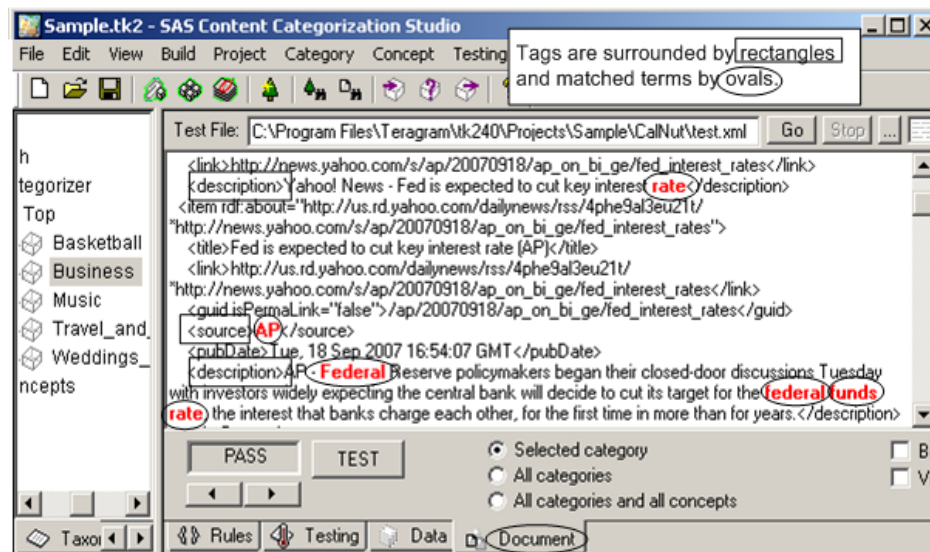
## 11.8.6 Testing the Structured Text Rule

Test a category rule in the **Document** tab to see the matched terms. For example, if you test the following rule:

```
(OR, "Federal", "funds", "rate", _source: "AP")
```

the results might be similar to the example shown below.

Figure 11-1 XML Document Test Results



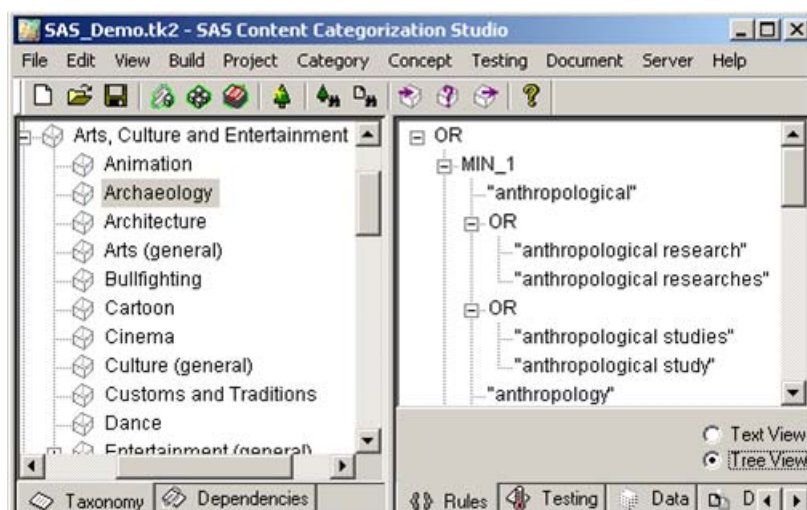
---

## 11.9 Editing Rules

### 11.9.1 Edit Rules in the Tree View Mode

Click **Tree View** in the Rules window to edit a Boolean rule. This mode enables you to display the Boolean rule in a format that is similar to the taxonomy tree. However, you can also expand and collapse the Boolean operators that form the taxonomy. Use these operations to view, or to edit, your rules.

*Display 11-8 Tree View*

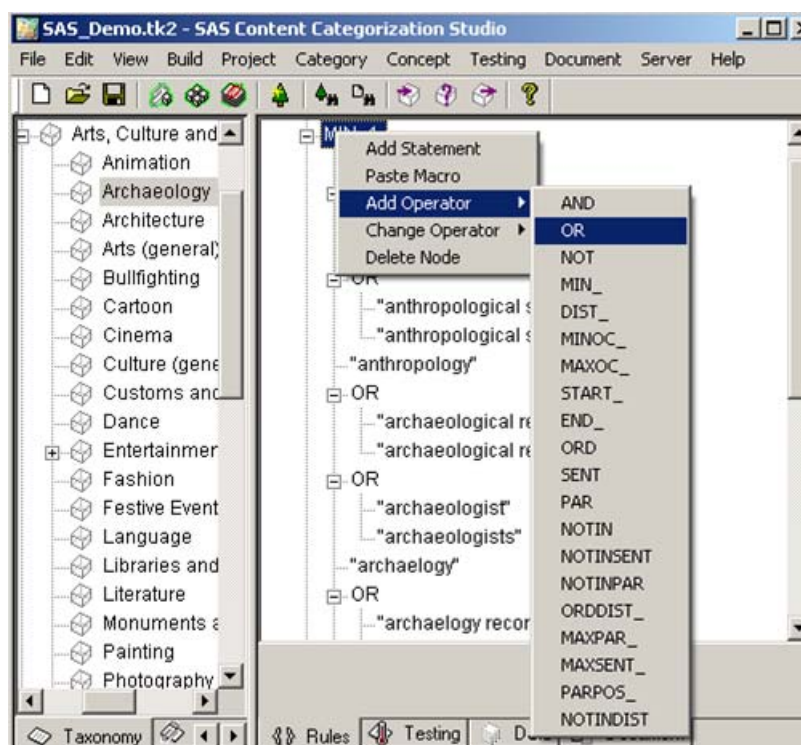




---

To edit Boolean rules by changing an operator in the Tree View mode, complete these steps:

1. Select a category in the **Taxonomy** tab and click the **Rules** tab.



2. Right-click on a Boolean operator in the Rules window that appears.
3. Select an operation such as **Add Operator**.
4. Choose one of the commands that is listed in the menu that appears. For example, select **OR**.

---

See the table below for a list and description of the rule tree commands.

Table 11-7: Rule Tree Commands

Command	Description
Add Statement	Insert an empty statement field below the selected Boolean operator. (This is the only way to <i>add</i> a statement, but you can edit a statement when you right-click on a term.) For more information, see Section 11.9.2 <i>About Statements and Operators</i> on page 324.
Paste Macro	Add the last copied macro into the selected Boolean statement. For more information, see Section 11.12.2 <i>Paste a Macro</i> on page 337.
Add Operator	Place the selected Boolean operator into the rule. For more information, see Section 11.9.2.B <i>Add an Operator</i> on page 326.
Change Operator	Change the selected Boolean operator to another operator. For more information, see Section 11.9.2.C <i>Change an Operator</i> on page 328.
Delete Node	Remove the selected operator. When you delete a node or an operator, all of its statements and child operators are also eliminated.

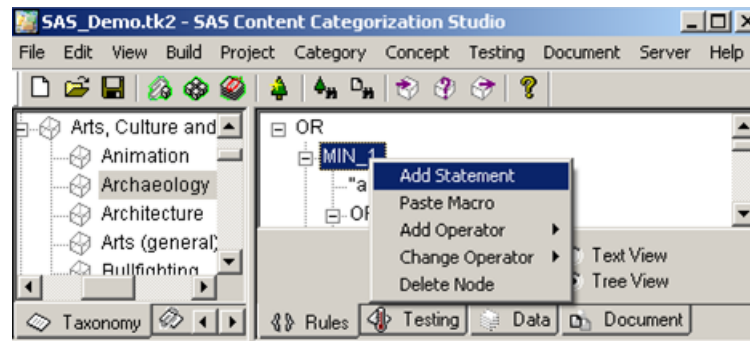
## 11.9.2 About Statements and Operators

### 11.9.2.A Add a Statement

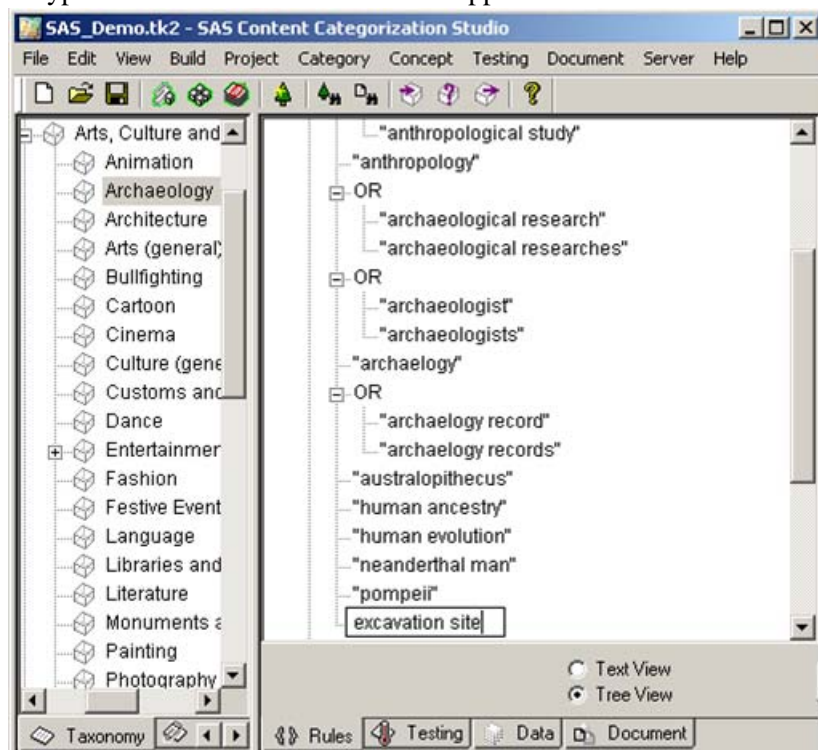
You can expand a category rule when you add a statement. Use this operation after you test a rule and see that it is not sufficient.

To add a statement to a Boolean category rule, complete these steps:

1. In the Tree View mode, right-click on the Boolean operator.



2. Select **Add Statement** from the menu that appears.
3. Type the new term into the box that appears.



4. Select **Build --> Build Rulebased Categorizer**.

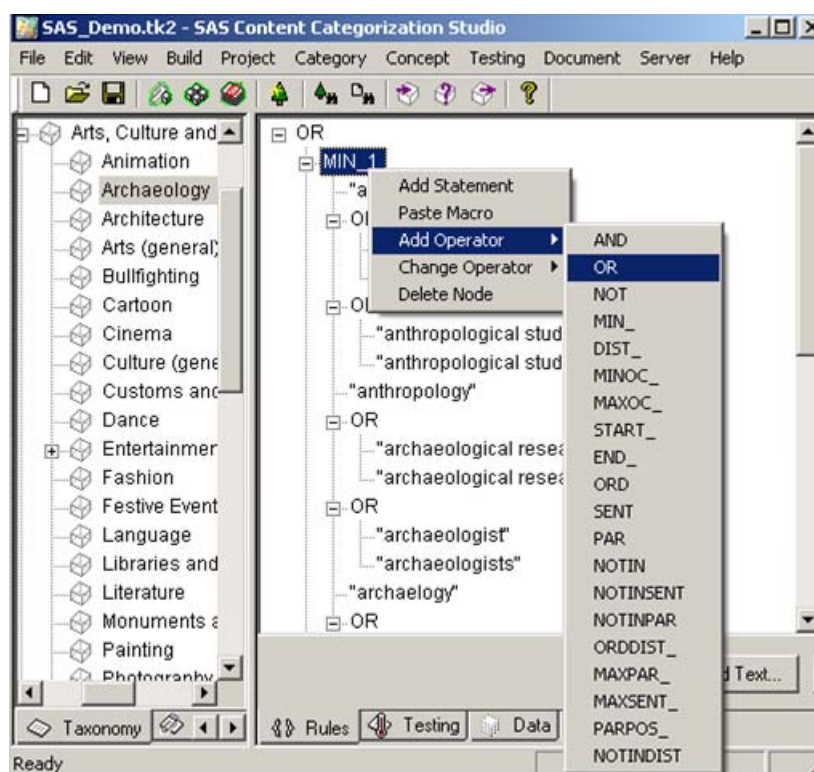
---

### 11.9.2.B Add an Operator

Place a selected Boolean operator at the end of a rule. When you add an operator, this operator appears at the bottom of the rule tree.

To add an operator to a Boolean rule, complete these steps:

1. In the Tree View mode, right-click on a Boolean operator.
2. Select **Add Operator** from the menu that appears. A drop-down list of operators appears.



3. Select one of the operators in the list. For example, choose **OR**. This operator is added to the bottom of this section of the category rule.



4. Select **Build --> Rebuild Categorizer**.

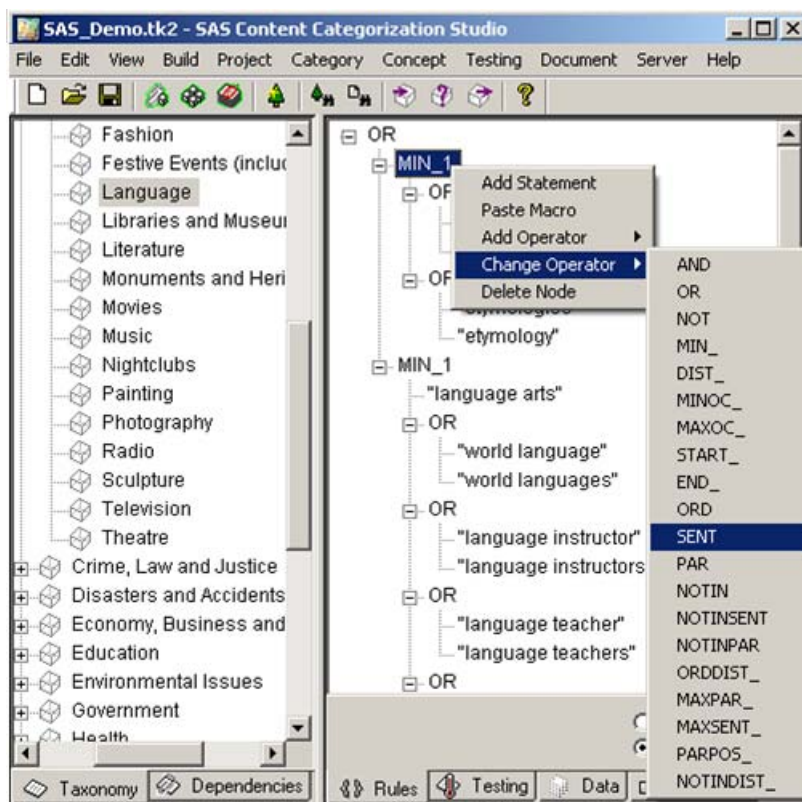
---

### 11.9.2.C Change an Operator

You can replace an operator in a Boolean rule when you select **Change Operator**.

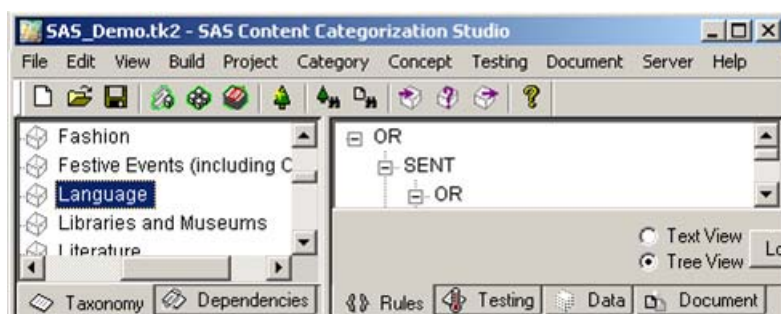
To change an operator, complete these steps:

1. In the Tree View mode, right-click on the Boolean operator that you want to change. For example, select MIN\_1.



2. Select **Change Operator** from the menu that appears.

3. Select the new operator. For example, choose **SENT**. The change appears in the rule.



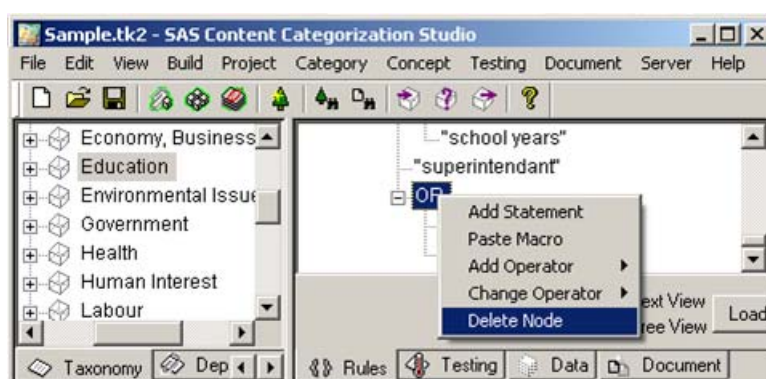
4. Select **Build --> Rebuild Categorizer**.

#### 11.9.2.D Delete a Node

To remove a node from a category rule use the **Delete Node** command. When you perform this operation, the node and all of its children are removed from the rule.

To delete a node in a rule, complete these steps:

1. In the Tree View mode, select a category in the **Taxonomy** tab.
2. Right-click on the node that you want to delete.



3. Select **Delete Node** from the drop-down menu that appears.



4. A SAS Content Categorization Studio confirmation window appears.



5. Click **Yes**.

### 11.9.3 About Statement Commands

You can access two statement commands when you right-click on one of the rule terms in the tree view of the Rules window.

*Display 11-9 Statement Commands*



Select one of the following operations to make changes to your Boolean category rule:

#### **Edit Statement**

Change the statement, or an identifier term. For example, you could change *school* to *school system*.

#### **Delete Node**

Remove the selected statement node of the tree. For example, delete the *school* node.



---

## 11.9.4 Expand Word Forms

Click **Expand Forms** in the Rules window to see and test the list of terms that are possible rule matches when you append:

@

Expand this word into all of its word forms.

@N

Expand this word into all of its noun forms.

@V

Expand this word into all of its verb forms.

The expansion type that you specify for a term is automatically incorporated into the <language>.mco file. For this reason, you might want to return all expansions for the original form before you test your rules.

Click the **Expand Forms** button to see, and edit, the list of expansions that would otherwise automatically be applied to input documents for matching purposes. For example, a rule defining the *Safety* category might list *securities* as an expanded form of the word *security*. However, the word *securities* relates to financial markets and does not mean to be *protected* or *secure*. For this reason, if your rule specifies safety and protection, you should *not* append an @ sign to this term.

After you see and test the expanded word forms, you can select either **Expand words with '@' sign**, or **Expand all word forms** in the Project Settings - Category window.

To expand and delete a word form, complete these steps:

1. Select a Boolean category rule, and append the @ sign to one of its words.



2. Click **Expand Forms** and the word is automatically expanded into all of its forms.



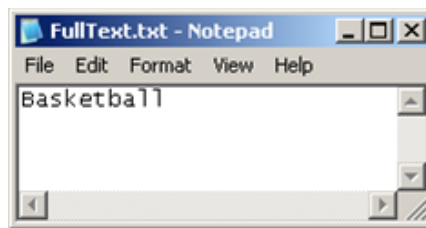
3. Select **Edit --> Undo**.

### 11.9.5 Flag Categories without Definitions

Select this operation to see a list of categories that do not have definitions. These categories are displayed in a *Notepad* window.

To automatically flag categories without rules, complete these steps:

1. Select **Edit --> Options** and the Options window appears.
2. Select **Flag categories/concepts with no definitions**.
3. Click **OK**.
4. Select **Build --> Build Rulebased Categorizer**.
5. Select **View --> Taxonomy as Text**. The FullText.txt - Notepad window appears and displays a list of any categories that are not defined by a rule.



- 
6. Select a listed category. For example, select `Basketball` and click the **Rules** tab. Write a rule that defines this category.

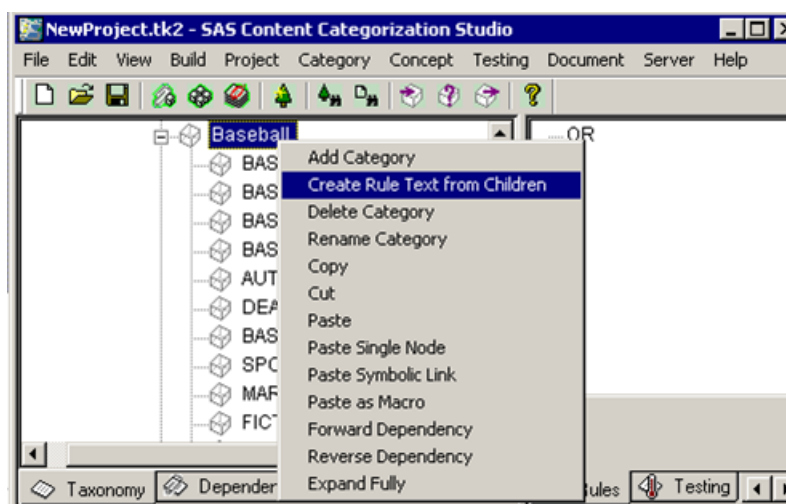
## 11.10 Automating Parent and Child Rule Generation

### 11.10.1 Generate a Parent Rule

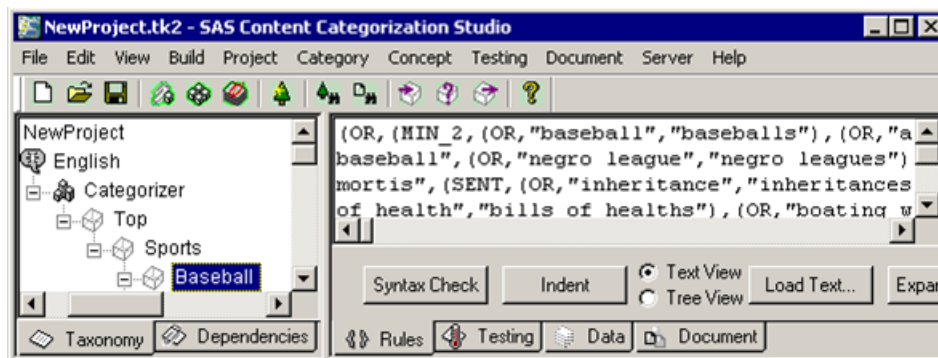
SAS Content Categorization Studio enables you to automatically combine the rules for child categories into one Boolean rule for their parent. SAS Content Categorization Studio uses the **OR** operator to join all of the child category rules into a single string. You can edit this rule, delete some or all of its subcategories, or use the existing parent and child rules in their current forms.

To develop an automatically generated rule from its automatically generated subcategories, complete these steps:

1. Select a parent category in the **Taxonomy** tab.
2. Right-click on the parent node. For example, `Baseball`. Select **Create Rule Text from Children** from the menu that appears.



3. Click **Text View** in the **Rules** tab to see the new category rule.



4. (Optional) Edit this rule.

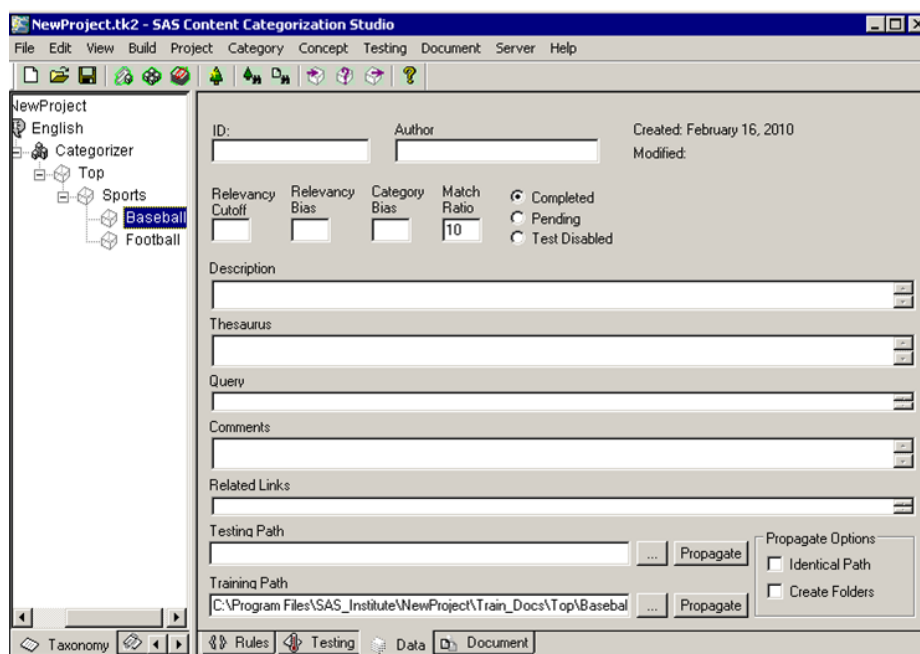
## 11.10.2 Generate Child Rules

You can automatically generate child categories from a parent category when you use the **Generate Subcategories** operation in the Category menu.

To automatically generate children, complete these steps:

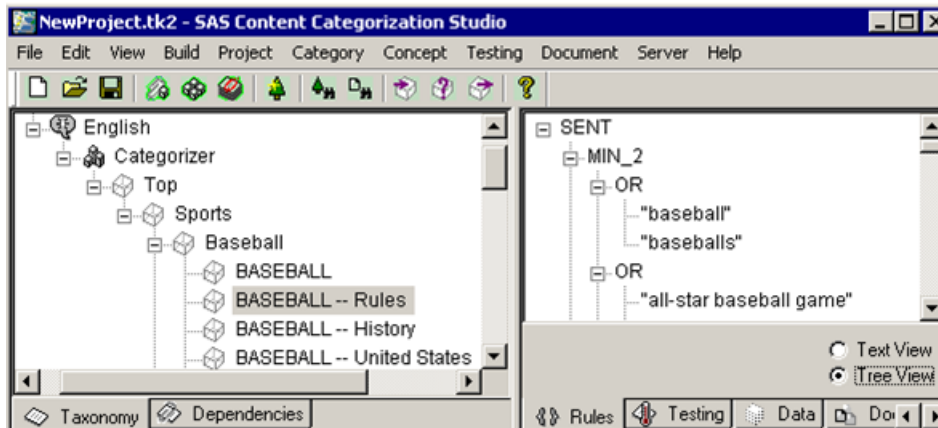
1. Select a category. For example, choose `Baseball`.
2. Click the **Data** tab and type the path to the training folder for this category into the **Training Path** field. For more information, see

Section 6.5.3 *Set Training Paths to the Training Directory* on page 189.



3. Select **Category** --> **Generate Subcategories**.

4. The automatically generated child categories appear beneath the selected parent node in the **Taxonomy** tab.



5. (Optional) Edit these rules.

## 11.11 Defining Symbolic Links

When you define Boolean rules for a rule-based categorizer, you can use symbolic links between a *source* and one or more *target* categories. The target category rules all point to the *source* category. When you choose to create a symbolic link, you write one category rule for the *source* category that is used by multiple *target* categories or subcategories in a single taxonomy. For more information, see Section 8.8 *Create Symbolic Links* on page 226.

---

## 11.12 Dependencies between Categories or Categories and Classifier Concepts

### 11.12.1 About Dependent Nodes

Dependencies create a link between two taxonomy nodes. In this relationship, the source category references the rule of a target category or classifier concept. The source category can use the entire, or part of, the target rule. You can create a dependency between two categories when each category is defined by a Boolean rule. Alternatively, you can define a dependent relationship between Boolean categories and classifier concepts.

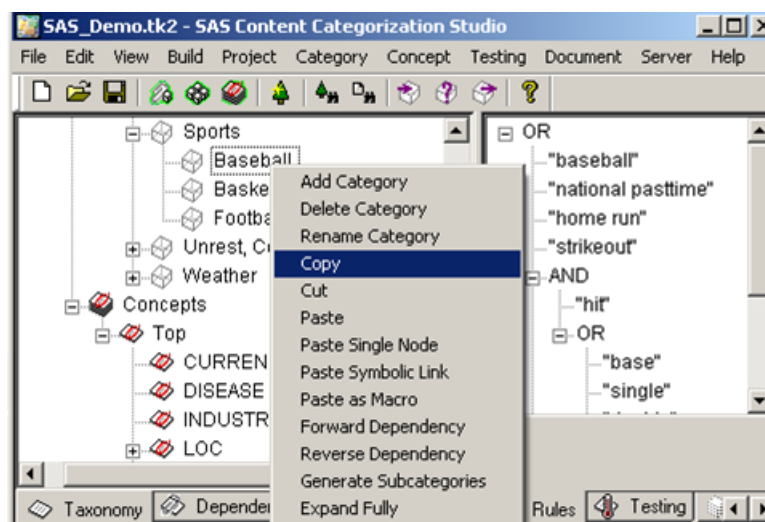
### 11.12.2 Paste a Macro

To define dependencies between Boolean rules, use the **Paste Macro** command within the rule of the target category. When you specify a macro, you create a pointer to the source category rule within the target rule. You can also select **Allow Short Macro Names** in the Project Settings - Category window. This operation shortens the path.

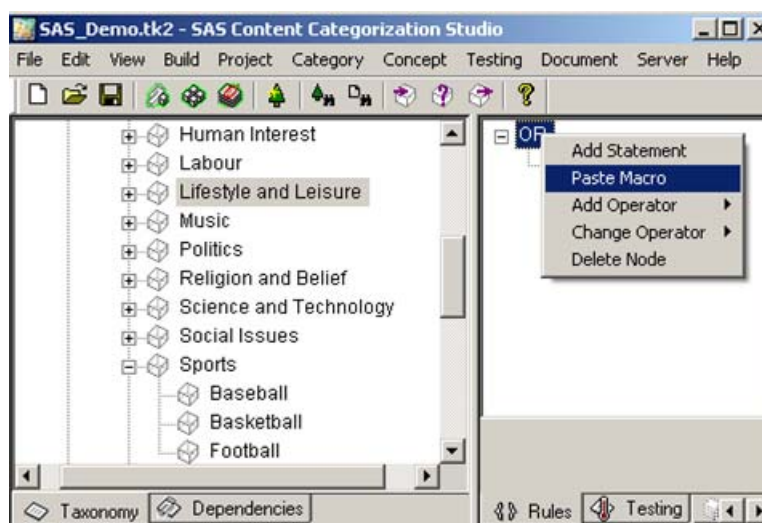
Dependencies use a macro in Boolean rules to reference an entire classifier concept definition as *part* of the selected category rule. The only type of concept that can be referenced by dependencies is the classifier concept where the rule is written in lowercase letters.

To create a dependency with the past macro operation, complete these steps:

1. Right-click on the target category and select **Copy** from the menu that appears.

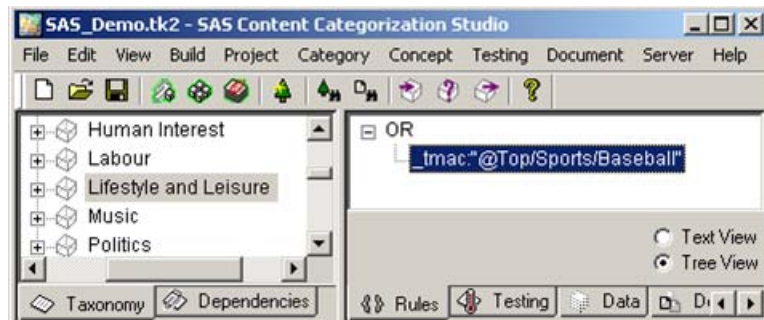


2. Select the category that is the source category and click the **Rules** tab. For example, select *Lifestyle and Leisure*.





- 
3. Select **Tree View** in the **Rules** tab.
  4. Right-click on a Boolean operator and select **Paste Macro** from the menu that appears. A macro pointing to the category that you selected, with its full path, is automatically pasted into the Boolean rule.



---

**Hint:** The `_tmac:` term is a macro rule that enables you to reference another Boolean rule.

---

5. Before you edit or delete a category or classifier concept, after you create one or more dependencies, click the **Dependencies** tab. You can see any dependencies in this tab before you delete a target node.

### 11.12.3 Shorten Pathnames

To make it easier to see and edit dependencies, select **Allow Short Macro Names** in the Project Settings - Category window. If you enable this operation, SAS Content Categorization Studio does not specify the full pathname of the referenced category.

Short macro names are specified only when your subcategory names are unique. For example, you might have two *Composers* categories whose full pathnames are: *Top/Music/Baroque/Composers* and *Top/Music/Romantic/Composers*. If you enable the **Allow Short Macro Names** operation, SAS Content Categorization Studio incorporates the first rule in the taxonomy structure. This is true regardless of the category that you copied to paste as a macro.

---

To use short macro names, complete these steps:

1. Select **Project --> Settings** and the **Category** tab appears.
2. Select **Allow Short Macro Names**.
3. Click **OK**. The paths to the source categories are abbreviated by SAS Content Categorization Studio when the macro is pasted. For example, instead of typing the full path, the short path shown in the following display is entered into the **Rules** tab.



Shorter paths make dependent relationships easier to locate in either the Rules or Dependencies windows.

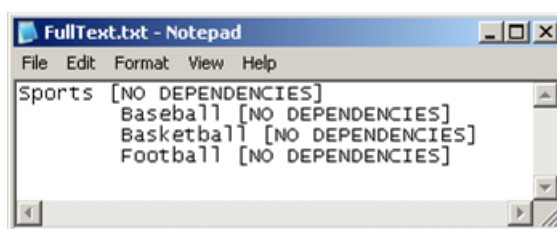
## 11.12.4 Flag Categories with No Dependencies

To locate Boolean categories that can be deleted without affecting other rules, select **Flag categories/concepts without dependencies** in the Options window.

To flag any categories that do not have dependencies, complete these steps:

1. Select **Edit --> Options**.
2. Under the **Taxonomy as Text** heading, select **Flag categories/concepts with no dependencies**.
3. Click **OK**.
4. Select **Build --> Build Rulebased Categorizer**.
5. Select **View --> Taxonomy as Text**.

- 
6. The FullText.txt window appears in the *Notepad* application.



7. The FullText.txt window displays a list of all of the categories that have no dependencies.
8. Click **X** to close this window.
9. (Optional) You can safely delete any categories that do not have a dependency.

## 11.13 Quick Start Guide to Testing Boolean Rules

To test the Boolean rules that you developed for the rule-based categorizer see *Part 2: Testing*. As you test your Boolean rules, consider the following factors:

### Category membership

Examine the testing results to see whether the categories are too broad or too narrow. When a category rule is too broad, documents that should not be categorized into a single category are matched to this category. However, if category rules are too narrow texts that should match do not.

### Unique linguistic terms

Precisely define the terms that uniquely identify members of each category for accurate matching.

### Boolean Operators

Check the rule syntax to see whether the selected Boolean operators obtain the results that you expect from your rules.

### Project Settings - Misc window

---

Check the specifications for the structured-text fields that you set here.  
The **Default Field** and **XML Tags to Ignore** should not conflict with the Boolean operators.

#### Category membership

Check any stemming operators if the rule matches appear incorrect.

#### Building and Rebuilding the Categorizer

Rebuild the categorizer if you did not select **Always rebuild before each test** in the **Edit --> Options** window.

## 11.14 Query an Index

The server query operation enables you to enter a search term that is a Boolean rule. This Boolean rule is automatically turned into a query and used to search the index. Click **Server Query** in the **Rules** tab to preview the way that documents in the index are categorized when you query the index.

Use the server query feature to make SAS Content Categorization Studio compatible with your index. This operation is not case sensitive at this time.

To use the server query operation, complete these steps:

1. Build an index.
2. Create the SAS Content Categorization Studio taxonomy project.
3. Query the index using the rule-based categorizer with Boolean rules.

---

**Notes:** When you choose to create dependencies between Boolean categories and classifier concepts, make these concept definitions lowercase.  
The server query operation does not support case sensitivity at this time.  
The server query operation converts linguistic rules into Boolean rules.

---

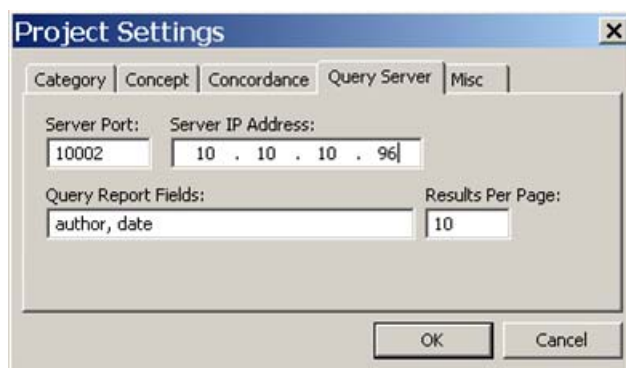
Connect your machine to the server where the index is located. SAS Content Categorization Studio automatically replaces the string in the query syntax

---

with the Boolean category rule for the category that you selected. You can see the number of documents that match the selected category rule in the Query Server Results window. For more information, also see the text of the matching documents in the **Document** tab.

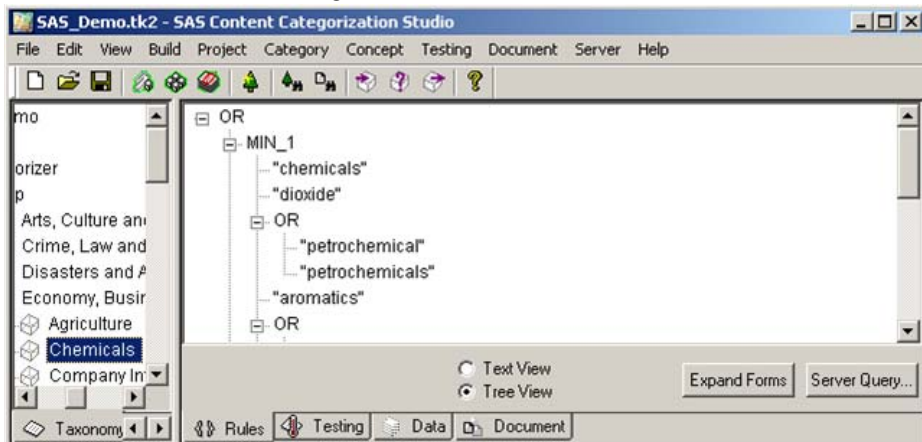
To preview the documents that SAS Content Categorization Studio returns from an index, complete these steps:

1. Create and test a SAS Content Categorization Studio taxonomy of Boolean categories.
2. Build an index.
3. Select **Project --> Settings** and click **Server Query** to access the Query Server tab.

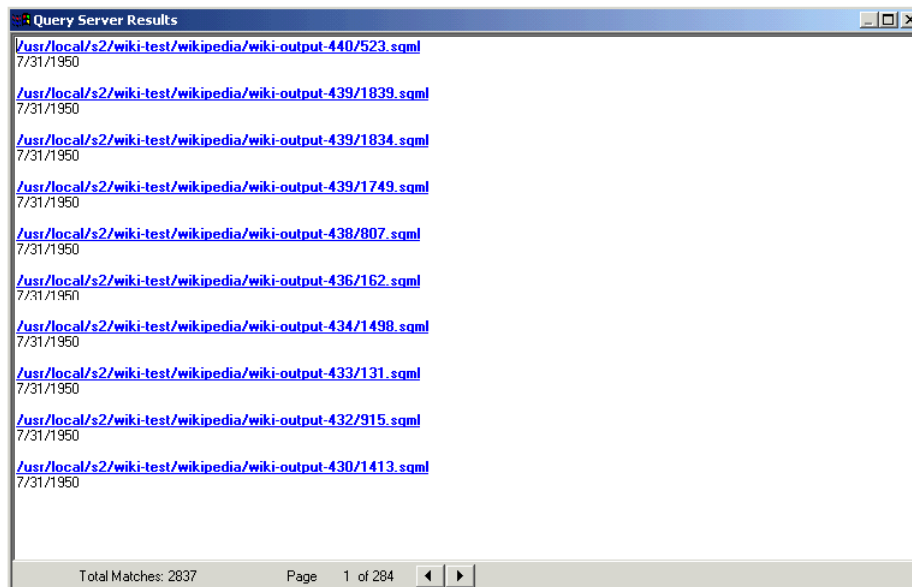


4. Type in the number to the query server port where the index is located in the **Server Port** field. For example, enter 10002.
5. Type the relevant IP address into the **Server IP Address** field. For example, enter 10 . 10 . 10 . 96.
6. Type the fields for your query report into the **Query Report Fields** field. These are the XML tags for the stored documents on the server.
7. Type the number of results returned and displayed in the Query Server Results window into the **Results Per Page** field. The default is 10.
8. Click **OK**.

9. Select a category in the **Taxonomy** tab. This is the category rule that the query protocol uses to search the index that has been built with a field configuration file that specifies Boolean terms.
10. Click **Server Query** in the **Rules** tab.



11. The Query Server Results window appears. This window displays a list of the specified query report fields that are a match as document links.



---

The bar at the bottom of the Query Server Results window displays the following information about all of the matches:

**Total Matches**

The total number of documents in the index that match the selected category rule.

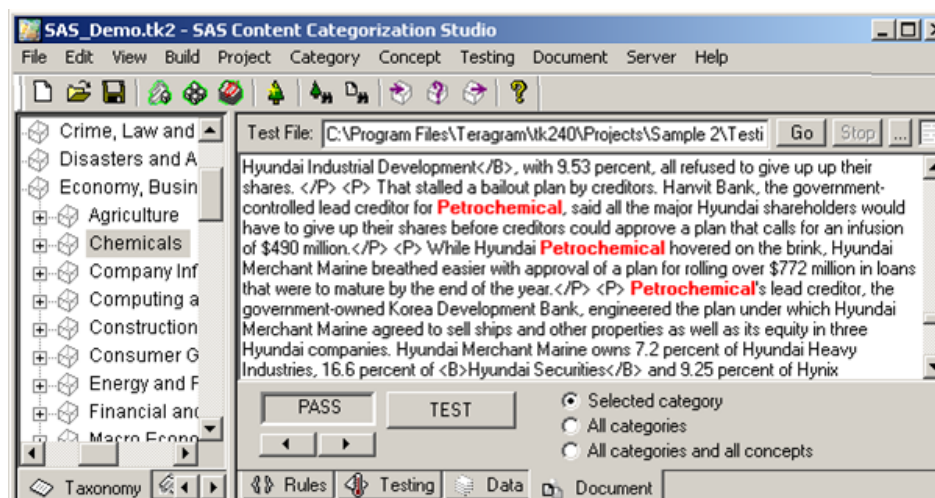
**Page**

This set of numbers specifies the page number that you see and the total number of pages found.

**Left and Right Arrow buttons**

These buttons enable you to click backward and forward, respectively, to see each of the result pages.

12. Click on one of the links to the returned texts to display it in the **Document** tab. The matched terms are highlighted in red.



13. (Optional) Use this process reiteratively until you obtain the results that you require.





---

## Part 2: Testing

---

- Chapter 12: *Assembling Testing Sets on page 349*
- Chapter 13: *Batch Testing on page 373*
- Chapter 14: *Testing with One Document on page 385*
- Chapter 15: *Other Testing Operations on page 397*



---

# Chapter: 12

## Assembling Testing Sets

---

- *Overview of Assembling Testing Sets*
- *Creating Testing Folders*
- *Collecting Test Files*
- *Manually Populating a Testing Folder*
- *Special Usages for a Central Repository*
- *Delete Testing Files*

### 12.1 Overview of Assembling Testing Sets

You gather groups of documents, or testing sets, for the purposes of testing the category rules that you develop in SAS Content Categorization Studio. These documents enable you to see the results that you can expect when SAS Content Categorization Server applies the rules to input texts.

To set up a directory of test documents, choose documents for each category that you expect to match the rule for that category. Place each set of these texts into a testing folder. Create one folder for each taxonomy node.

After you test the testing directory, set up a central repository that is one folder of testing documents. Place documents that are similar to the real world texts that you plan to categorize, but not matched to individual categories, into this folder. For this reason, the central repository is a large group of documents that test the entire taxonomy. This repository can also contain a directory for files that do not match any category.

Before you begin testing your category rules, use the directions in this chapter to develop each of the types of testing directories that you want to use. An overview of the process detailed in this chapter is provided below:

1. Create the directory of testing folders for individual categories that matches the taxonomy of categories.

- 
2. Collect 5 - 10 documents that you expect to match each category.
  3. Place these testing documents into the folders that you created.
  4. Set the paths to these files.

You can also automate some of these steps. For example, you can create a top level testing folder and use the **Create Folders** check box and the **Propagate** button in the Data window. These operations simultaneously create testing subdirectories and set the paths to these directories. For more information, see Section 12.2.1 *Create a Testing Directory While You Set Paths* on page 350. For this reason, you might want to read through this chapter before deciding how to create your testing folders.

Testing documents help to determine whether, and why, a category rule should be changed so that the rule correctly categorizes texts. For this reason, the test files that together comprise the testing set, or sets, of documents are integral to developing a successful SAS Content Categorization Studio application. The process of testing and refining rules can be used reiteratively until you obtain a satisfactory set of rules.

## 12.2 Creating Testing Folders

### 12.2.1 Create a Testing Directory While You Set Paths

Use SAS Content Categorization Studio to automatically create the testing directory while setting the testing paths to these folders. This operation saves time and ensures that an exact replication of the taxonomy displayed in the Taxonomy window is copied for the testing documents.

---

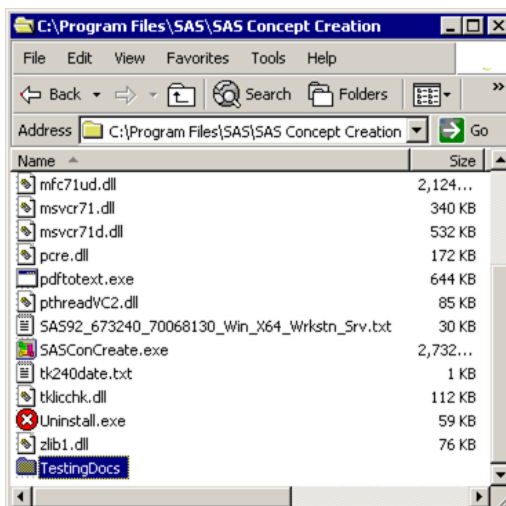
**Note:** If you rename a category, remember to also change the name of the testing folder.

---

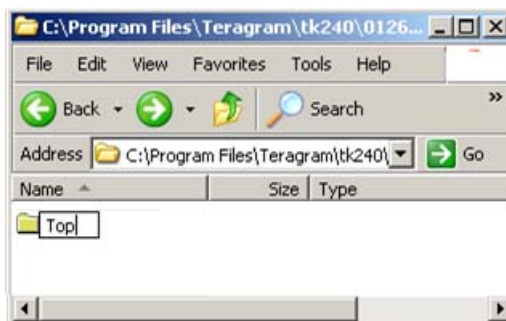
---

To define the testing taxonomy while simultaneously setting the testing paths, complete these steps:

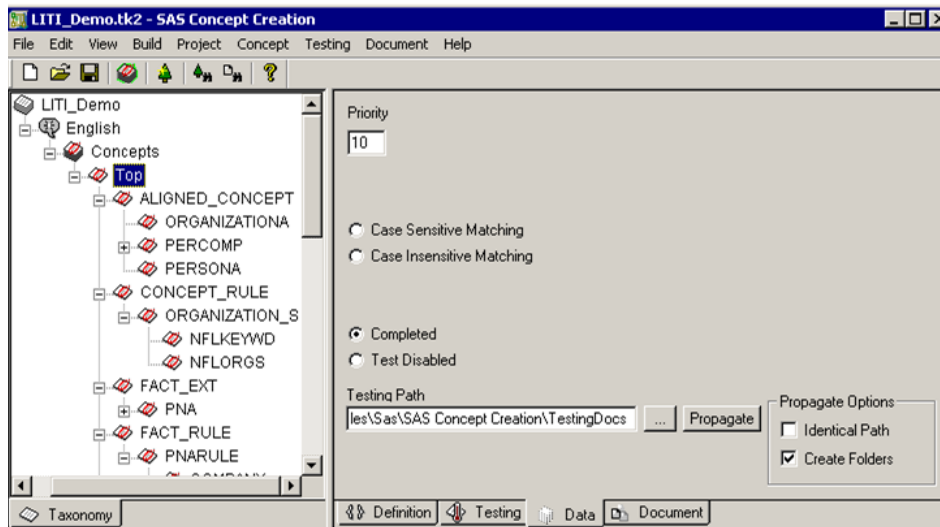
1. Access the folder for your project and create a new file for the testing documents. Name this folder. For example, type in the name `docs`.



2. Double-click the testing folder and create a new folder named *Top* to match the `Top` folder in the Taxonomy window. This folder is used to automatically propagate the testing paths to each of the concepts in your taxonomy.



- 
3. Select the `Top` folder in the Taxonomy window.




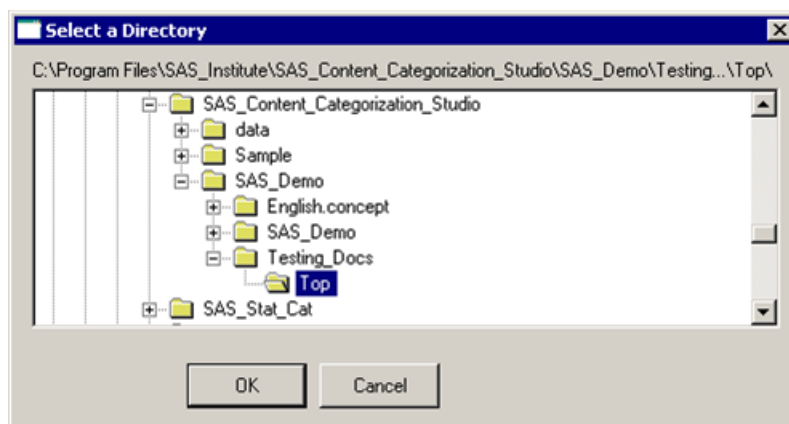
---

**Note:** If you click another node, SAS Content Categorization Studio creates only subdirectories for the selected concept node.

---

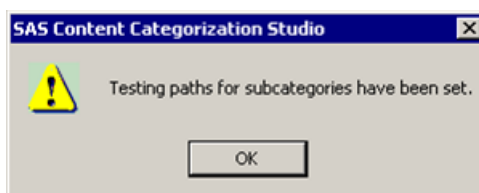
4. Select **Create Folders** under the **Propagate Options** heading in the Data window.

- 
5. Click  to the right of the **Testing Path** field and the Select a Directory window appears.



6. Select the **Top** directory where SAS Content Categorization Studio creates the testing taxonomy.
7. Click **OK**.
8. Click **Propagate** in the **Data** tab.

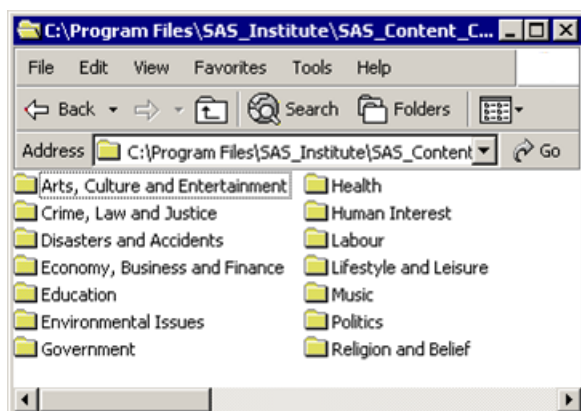
A SAS Content Categorization Studio confirmation window appears.



9. Click **OK**.

---

A directory structure that is identical to the categories taxonomy is created inside the `TOP` folder.



10. Click some of the category nodes in the Taxonomy window to see that each **Testing Path** field displays the path to the matching testing directory.

Unless each folder in the testing directory is populated with your testing documents, you cannot test your categories. For more information, see Section 12.4 *Manually Populating a Testing Folder* on page 359.



---

## 12.2.2 Create and Set a Path to the Central Repository

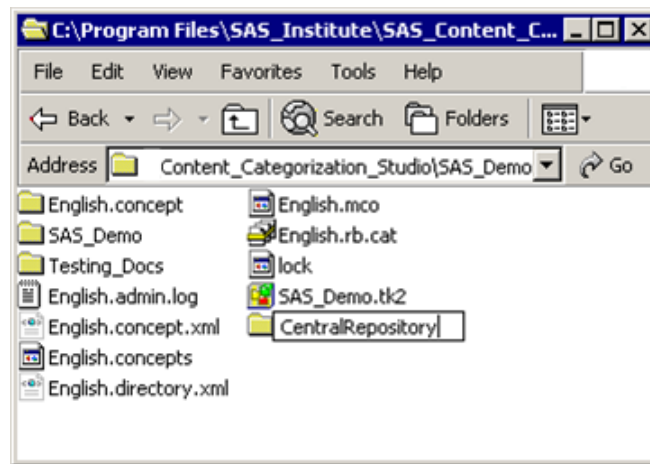
A central repository of testing documents contains a set of texts that are not selected to match individual categories. For this reason, when you test the central repository, you gain a realistic approximation of the results that you might obtain with real-world documents.

Use a central repository of testing documents for the following purposes:

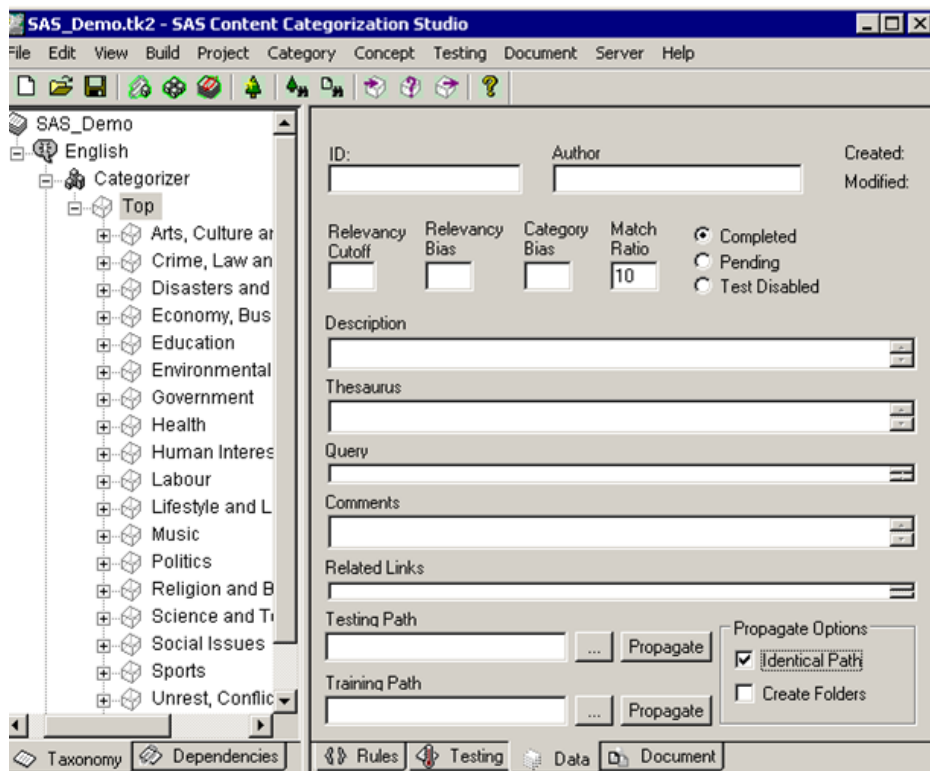
- This operation is typically the final testing stage and should replicate real-world results.
- A central repository can also be used as an alternative to creating a directory tree structure, or it can be used to populate the testing taxonomy. For more information, see Section 12.5.1 *Automatically Populate Testing Paths* on page 360.
- This selection can be a temporary substitute for a testing directory structure.

To create and set a path to the central repository, complete these steps:


1. Create a single folder that is the central repository in the project directory on your hard drive. For example, create `CentralRepository`.

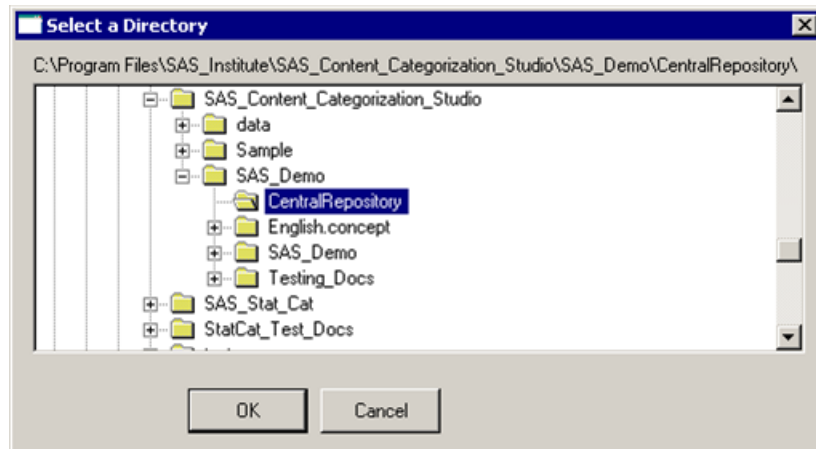


2. Select the **Top** folder in the Taxonomy window.

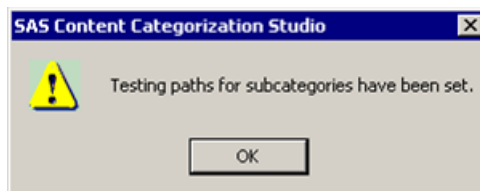


3. Select **Identical Path** under the **Propagate Options** heading in the Data window.

- 
4. Click  to the right of the **Testing Path** field and the Select a Directory window appears.



5. Click **OK**.
6. Click **Propagate** in the **Data** tab. A SAS Content Categorization Studio confirmation window appears.



7. Click **OK**.
8. (Optional) Click some of the category nodes and you can see that each node each displays the same path to the central repository in the Data window.
9. Click **TEST** in the **Testing** tab to test each category against the same set of testing documents.

Unless a folder in the testing directory is populated with testing documents, you cannot test the matched category. For more information, see Section 12.4 *Manually Populating a Testing Folder* on page 359.

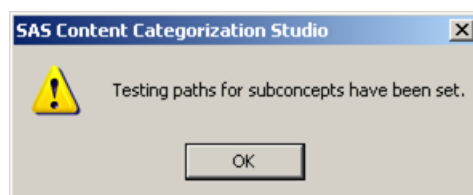
---

### 12.2.3 Create a Testing Folder and Set a Path for a Newly Created Category

If you add one or more categories to the taxonomy, after you set up the testing directory, you can add a matching testing folder. Manually set the path to this folder.

To add a test folder and set the path, complete these steps:

1. Access the testing directory. Create and name a new folder for the category that you added to the taxonomy.
2. Type the path to this folder into the **Testing Path** field of the Data Window. Do not select either of the check boxes under **Propagate Options**.
3. Click **Propagate**. A SAS Content Categorization Studio confirmation window appears.



4. Click **OK**.

---

## 12.3 Collecting Test Files

After you create repositories and set the paths to these directories, assemble different sets of testing documents. Choose texts that should be categorized into the specific categories that comprise your overall taxonomy structure.

The SAS Content Categorization Studio testing process uses the testing taxonomy to determine the precision and recall of your categorizer. Precision measures the relevancy of the matched documents, while recall measures whether all of the texts that should be returned are matched. For these reasons, each category rule should be broad enough to include all of the texts that you expect to match. These rules should also exclude any documents that do not belong to the selected concept.

Use the following two steps to assemble the different types of texts required to test your taxonomy. In each case, choose documents of the types that are input to SAS Content Categorization Server. For example, select `.html`, `.xml`, `.sgml`, `.pdf`, `.rtf`, and `.txt` documents.

First, select 10 or more documents that are matches for each category in your taxonomy. These texts should have varying degrees of categorization complexity levels for the category rules that you define. Copy and paste each group of documents into the testing folder named for the category that they are expected to match.

Second, collect a group of documents that include texts that are similar to the types of documents that are used when this application is applied in real time. Place this group of texts into the central repository that you created. When you choose to use a central repository, you can see whether your documents match more than one category and if so, why. For more information, see Section 15.2 *Test a Central Repository* on page 398.

## 12.4 Manually Populating a Testing Folder

You can place all of the testing files that you collect for a testing directory or a central repository into these folders using cut and paste operations. Also use this operation when you add a category to the taxonomy.

---

## 12.5 Special Usages for a Central Repository

### 12.5.1 Automatically Populate Testing Paths

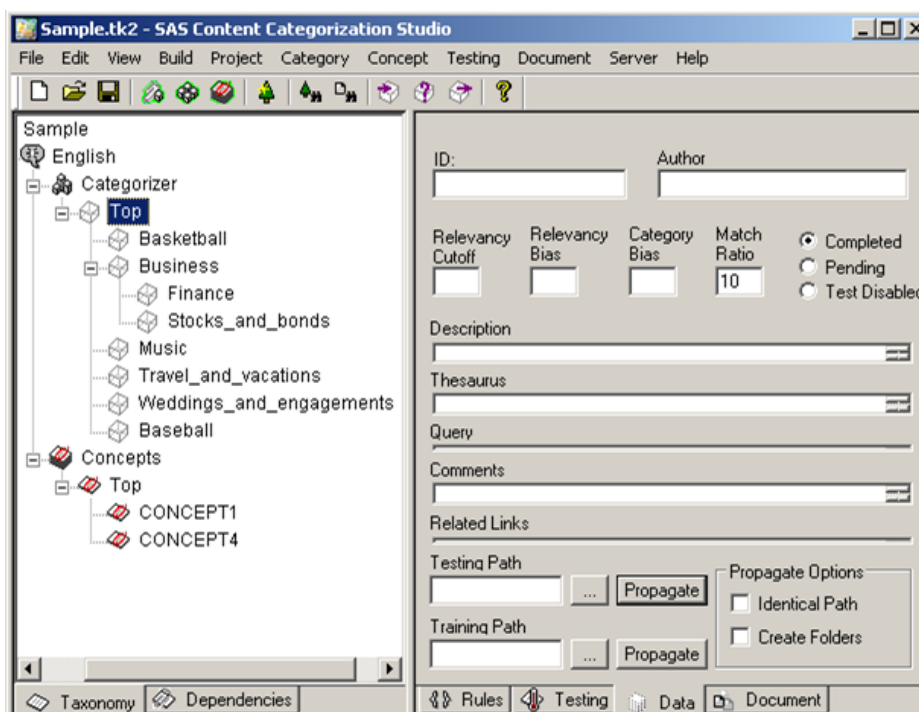
The Populate Testing Paths operation runs all of the testing documents, located in one repository, through the taxonomy. This operation automatically matches the documents to categories. The testing path is automatically entered into the **Testing Path** field of the Data window for each category.

You can choose to use the central repository or another folder of testing documents. Matching texts are automatically assigned to the categories that they match. You can see the results in graph format after the operation is complete. You can also see the numbers of matches in the Taxonomy window.

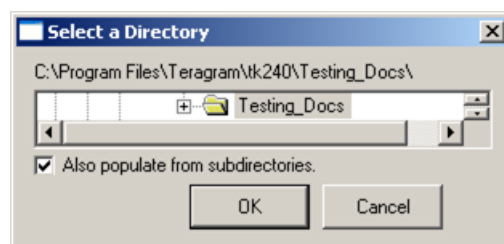
If you want to locate any unmatched files in a separate directory, see Section 12.5.2 *Create a Directory of Unmatched Testing Files* on page 365 after you use this section.

To populate the testing paths and see the graphed results, complete these steps:

1. Select the **Top** node and set the **Testing Path** in the Data window to the folder of testing documents. For example, specify the central repository.

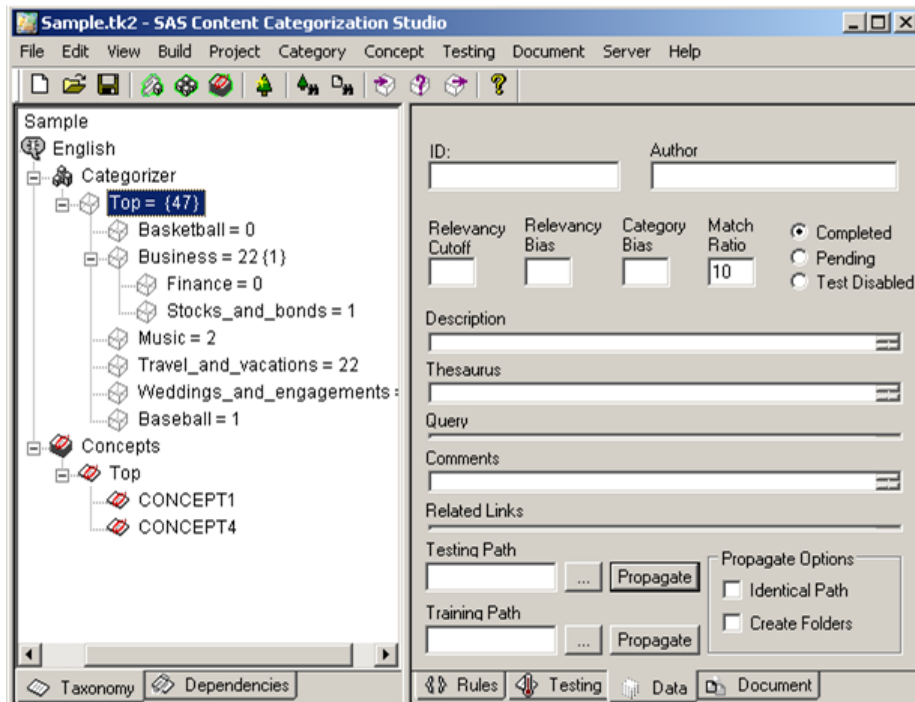



2. Select **Testing --> Populate Testing Paths**. The Select a Directory window appears.



3. Select a repository of testing documents in the Select a Directory window. For example, choose `Testing_Docs`.

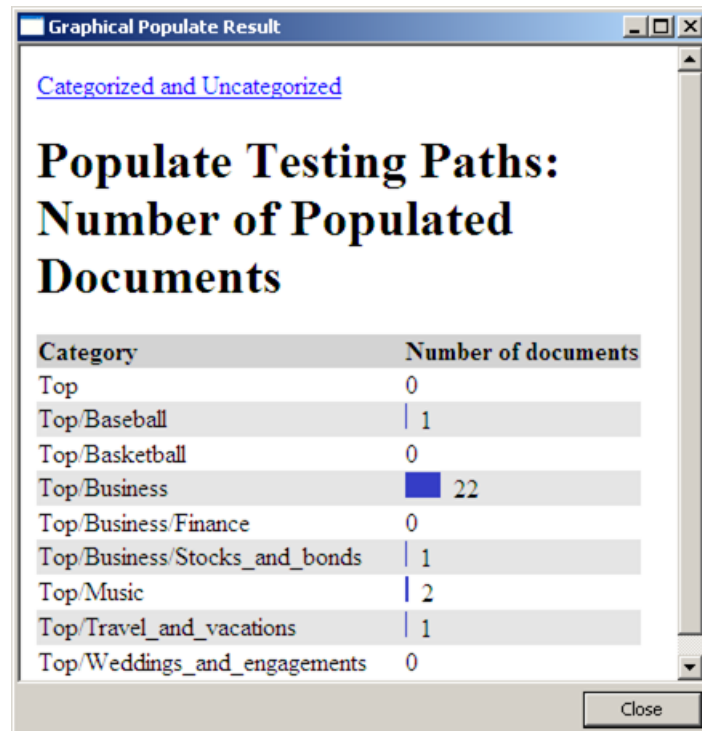
4. (Optional) If you place documents in subfolders within the central repository, select **Also populate from subdirectories**.
5. Click **OK**. The Taxonomy window displays the number of testing documents that match each category. The number for parent categories is followed by bracketed ({} ) numbers representing the matches for each child.



**Hints:** Erase these matches when you click . These matches might also disappear when you close and access your project.

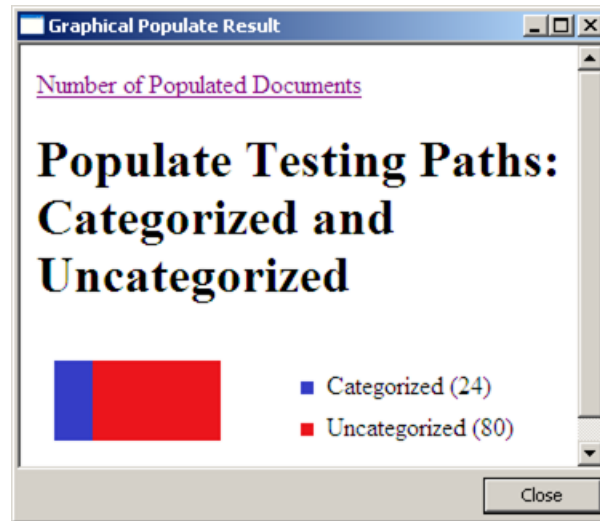


6. Select **Testing --> Show Graphical Populate Results**. The Graphical Populate Result: Number of Populated Documents page appears.



7. (Optional) Click the **Category** heading to display the categories alphabetically starting from the letter Z down or from the letter A down.
8. (Optional) Click the **Number of documents** heading to display the results starting from the lowest, or the highest, number of matches.

- 
9. Click **Categorized and Uncategorized** to see the Populate Testing Paths: Categorized and Uncategorized page.



10. (Optional) Click **Number of Populated Documents** to return to the Populate Testing Paths: Number of Populated Documents page.
11. Click **Close**.
12. (Optional) To see these results after you have closed these pages, select **Testing --> Restore Populate Results**.

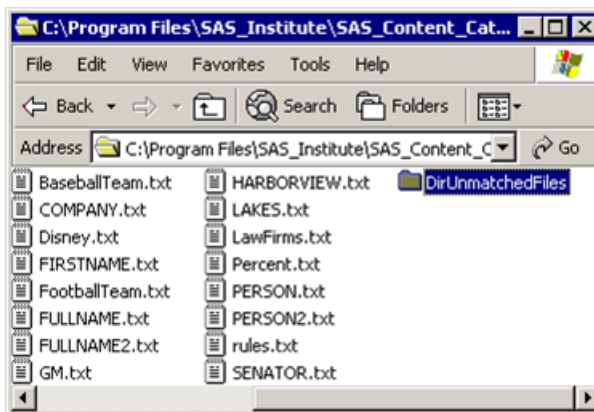
---

## 12.5.2 Create a Directory of Unmatched Testing Files

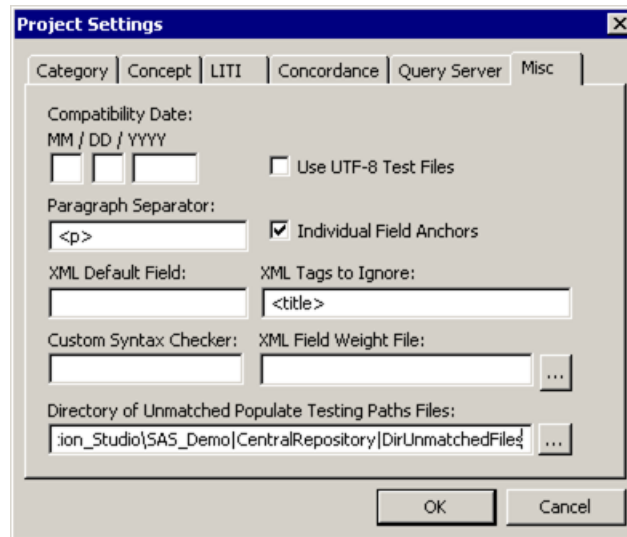
Create a folder where SAS Content Categorization Studio can load all of the testing files that do not match any of your category rules. You create this folder within the central repository of testing files. Use this directory to determine what types of documents do not match your category rules and why.

To create a folder for unmatched testing files that reside in this central repository of documents, complete these steps:

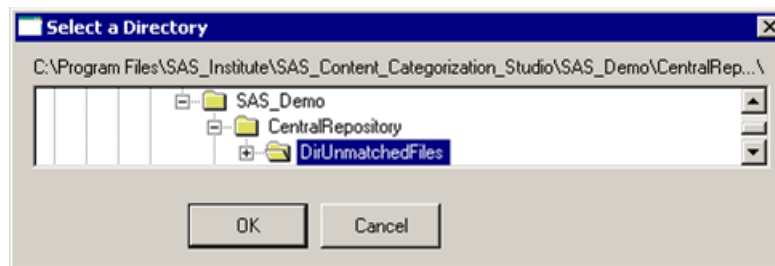
1. Create an unmatched file folder. For example, add `DirUnmatchedFiles` to the central repository.



2. Select **Project --> Settings** and select the **Misc** tab.

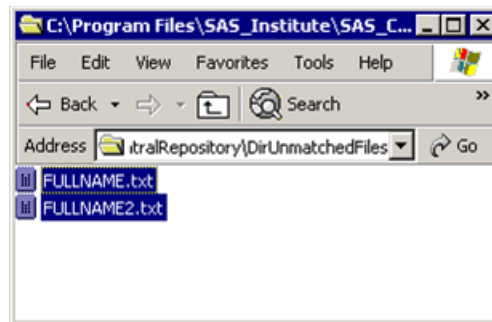


3. Click  to the right of the **Directory for Unmatched Populate Files** field. The Select a Directory window appears.



4. Select the directory of unmatched files where your testing documents are located. For example, choose `DirUnmatchedFiles`.
5. Click **OK**.
6. The path to the unmatched files directory appears in the **Directory for Unmatched Populate Files** field.
7. Click **OK**.

- 
8. Use the appropriate steps in Section 12.5.1 *Automatically Populate Testing Paths* on page 360.
  9. (Optional) Access the directory of unmatched testing files to see whether there are any unmatched files. For example, see `FULLNAME.txt` and `FULLNAME2.txt`.



10. Access each of these files and compare them to the category rules.
11. (Optional) Edit your category rules, or add additional categories.

---

### 12.5.3 Import Test Files from a Central Repository

You can add additional testing files to the Testing window for a selected category when you use the **Import Test Files** operation. Use this operation with a central repository or any other folder of files.

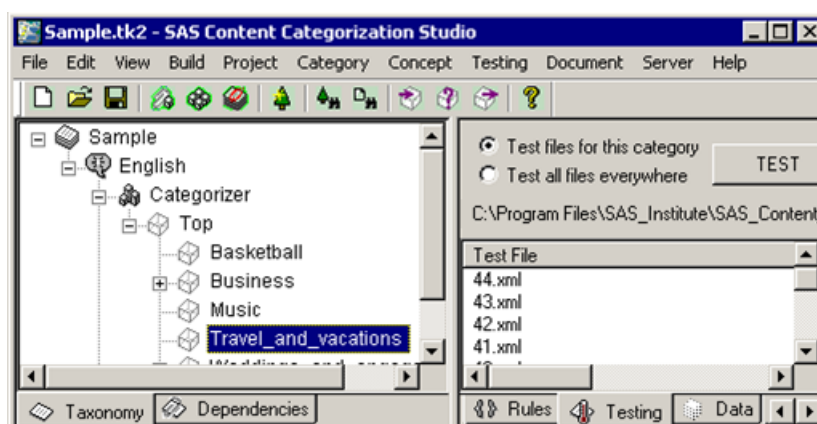
---

**Note:** Before you use the steps below make sure that the Testing window is populated with some files.

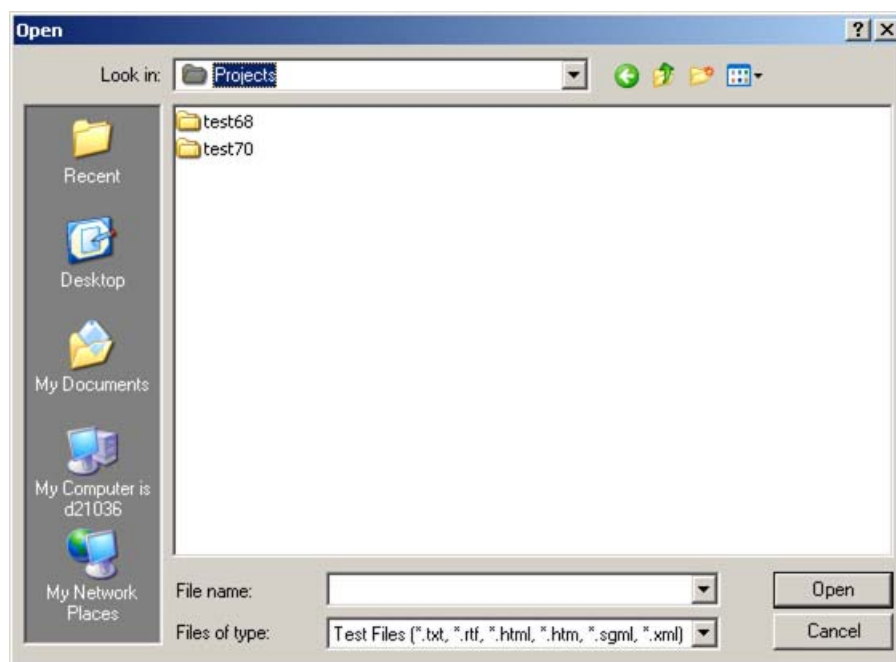
---

To import test files, complete these steps:

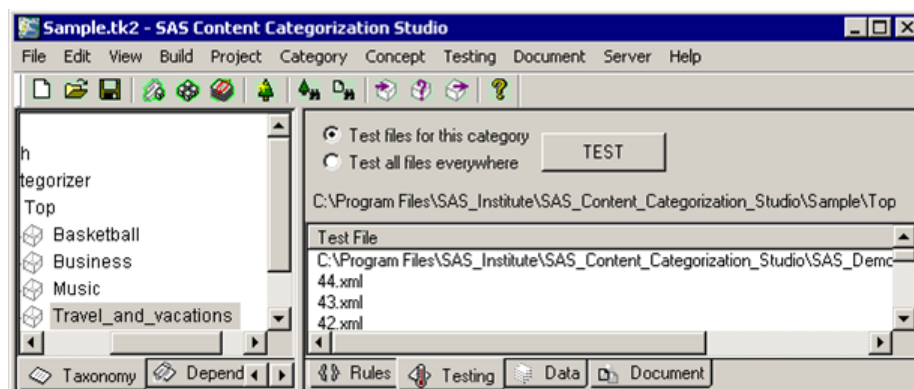
1. Select any category and click its **Testing** tab. The **Test File** window displays the testing files that are found in the matched testing folder.



2. Select **Testing --> Import Test Files**, and the Open window appears.



3. Select the test documents that you want to add.
4. Click **Open**. The selected test files are copied to the category's testing directory and listed in the **Testing** window.



- 
5. Repeat Step 1 on page 368 through Step 4. above to add testing files to any other categories.
  6. Begin testing these files. For more information, see Section 13.4.1 *Option 1A: Batch Testing All of the Documents in One Category* on page 379.

When you import the testing files, these files are referenced by the `tg_status.xml` file that SAS Content Categorization Studio creates in the selected testing folder.

---

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <TeragramTestStatusV1>
- <System>

    <HardwareID>3T4X75W1CLSXW1CHVXBTEX9D99959H9T8XG1EL8XGLELGL8XETG5FX</HardwareID>
    <NumPopulateMatches>0</NumPopulateMatches>
    <NumPopulateRelevantMatches>0</NumPopulateRelevantMatches>
    <NumPopulateChildMatches>0</NumPopulateChildMatches>
- <TestFiles>
    <File />
    <File>C:\Program
      Files\SAS_Institute\SAS_Content_Categorization_Studio\SAS_Demo\CentralRepository\Afric
    </TestFiles>
```

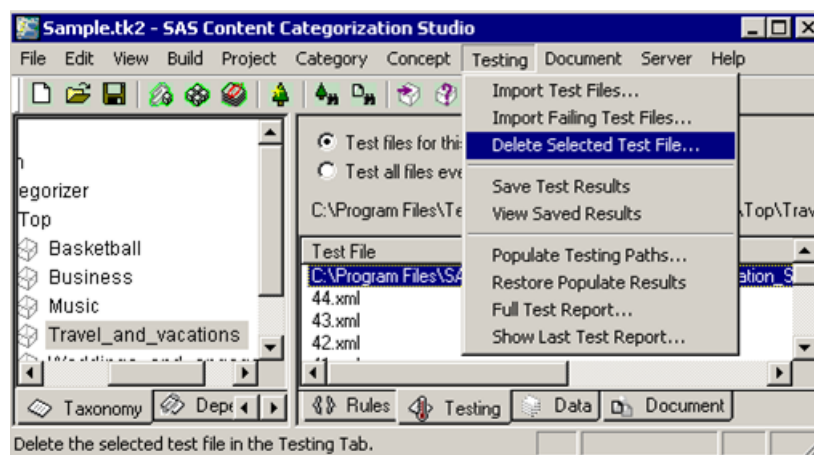


---

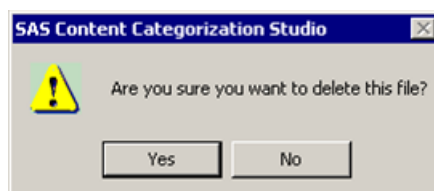
## 12.6 Delete Testing Files

To delete any of the testing files that you added to a testing folder, complete these steps:

1. Select a file in the **Testing** tab.



2. Select **Testing --> Delete Selected Test File**.
3. A SAS Content Categorization Studio confirmation window appears.



4. Click **Yes**.



---

# Chapter: 13

## Batch Testing

---

- *Overview of Batch Testing*
- *About Testing Window Messages*
- *Save and Compare Test Results*
- *About Batch Testing*
- *Remove a Testing File*

### 13.1 Overview of Batch Testing

#### 13.1.1 Batch Testing Operations

A batch of testing documents is defined as the group of texts that you assemble to test the precision and recall of a category rule. Before you begin to gather and test these documents, you should define at least some of the categories in your taxonomy.

When you test multiple categories using batches of testing documents, you gain information about the precision and recall of each category rule. This is true when a category rule is applied to the testing documents selected for this category. It is also true for the other test documents in the testing directory.

However, if the testing documents that are specified as part of the testing set for one category also match another rule, one of these rules might be too broad. If, on the other hand, the texts selected for the specified category fail to match the rule, the rule could be too narrow. Use the batch testing process to examine the results of numerous documents. See how the testing sets perform against the entire taxonomy. See why these documents match, or do not match, the category that they are selected to match.

Batch testing, or testing one group of documents at a time, is only one of the five testing operations available in SAS Content Categorization Studio. Use a

---

combination of these operations to develop a step-by-step, customized testing process that meets the specific requirements of your organization:

- Batch test your documents using the following operations in the Testing window:

**Test files for this category**

Batch test all of the files that you selected for each category against its rule. The test files that you assembled should pass the membership requirements for this category. For more information, see Section 13.4.1 *Option 1A: Batch Testing All of the Documents in One Category* on page 379.

**Test all files everywhere**

Use all of the documents in the testing directory. This means that you test all of the documents matched to each of the categories in the taxonomy at one time, and against one category. For more information, see Section 13.4.2 *Option 1B: Batch Testing the Testing Taxonomy or Out-of-Category Files* on page 381.

- Use the Document window to see the matching results for one document highlighted in red. For more information, see Chapter 14: *Testing with One Document*.
- Test all of the documents in the central repository. This folder contains documents that should, and should not, match the selected category. In this case, you obtain test results that might be closer to the real project application. For more information, see Section 15.2 *Test a Central Repository* on page 398.
- Import failing test files at any time during the testing process. Failing test files are defined as documents that could pass, but should fail. For example, documents that mention *President George W. Bush* should not match category rules such as *Gardening bushes*. For more information, see Section 15.3 *Import Failing Documents* on page 402.
- Use the category Test Report window to see a statistical analysis of the testing results for your categorizer. For more information, see Section 15.5 *About the Full Test Report* on page 408.

---

**Note:** When you use the rule-based categorizer, you can test each category as it is added to the taxonomy. When you use either the statistical categorizer or the automatic rule generator tool, it is necessary to define the entire taxonomy before you test your categories.

---

In summation, the batch testing operation provides an overview of the precision and recall of the category rules. You can also see the test results in detail by viewing individual texts in the Document window. Use these operations and test the central repository to obtain in-depth testing information. See the Category Test Report window to view the test information in table format.

### 13.1.2 About Testing Windows

You can use the following windows to test your categories:

#### Testing

Batch test the testing directory using the Testing window. `PASS` and `FAIL` messages appear in this window for each tested document.

#### Document

Select the Document window to test and view the testing results for a single document. You can test one document against a single category, all of the categories, or against all of the categories and concepts in the project.

#### Best Matches

When you test against multiple nodes, this window appears and the Taxonomy window displays `PASS` and `FAIL` messages for the entire taxonomy.

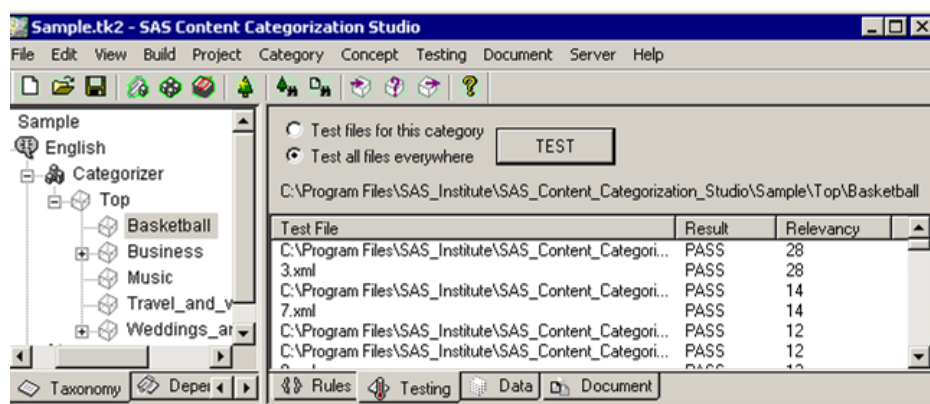
#### Category Test Report

Interpret the testing results for the list of categories that is displayed with counts and statistics to help you understand the testing results.

## 13.2 About Testing Window Messages

Before you use the Testing window, you should understand the testing messages that appear. For information about the components of the Testing window, see Section 2.7.3 *The Testing Tab* on page 31.

Display 13-1 Testing Window



The following types of messages are displayed in the **Testing** tab:

Path to the testing set of document

This path appears below the **TEST** button and above the **Test File** heading.  
For example:

```
C:\Program  
Files\Teragram\tk240\Sample\test\Top\Basketball
```

### Test File

A list of the test files is preceded by a full path to the out-of-category test files. These test files are imported using the **Test all files everywhere** or the **Testing --> Import Test Files** operation. The test files without a path belong to the testing folder that is matched to the selected category.

Missing folders and files

No testing folder

If there is no testing folder that matches the selected category in the testing taxonomy, a message such as `This directory does not`

---

`exist` is displayed. Set the path to the testing directory using Section 12.2.1 *Create a Testing Directory While You Set Paths* on page 350

Testing folder is empty

If the testing folder is empty, the message `No files found` appears. Place test files into the testing directory. For more information, see Section 12.4 *Manually Populating a Testing Folder* on page 359.

### Result

PASS

The percentage of matching terms located in the document meets the **Default Relevancy Cutoff** setting in the Project Settings - Category window. Alternatively, this percentage meets the **Relevancy Cutoff** specified in the Data window for this category. For linguistic rules, unless the percentage of matched terms is equal to or exceeds the match ratio, no match on the input document occurs.

PASS\*

A text that fails to meet the relevancy requirements, is considered to be *conditionally* passing. This is true for matches on linguistic rules if they also meet the match ratio specification.

FAIL

The document failed to meet the relevancy setting. In the case of linguistic rules, this might also mean that the text lacked the percentage of matches specified in the **Match Ratio** field.

### Relevancy

This column appears after you click **TEST**. The relevancy numbers are displayed here for each passing, or conditionally passing, document.

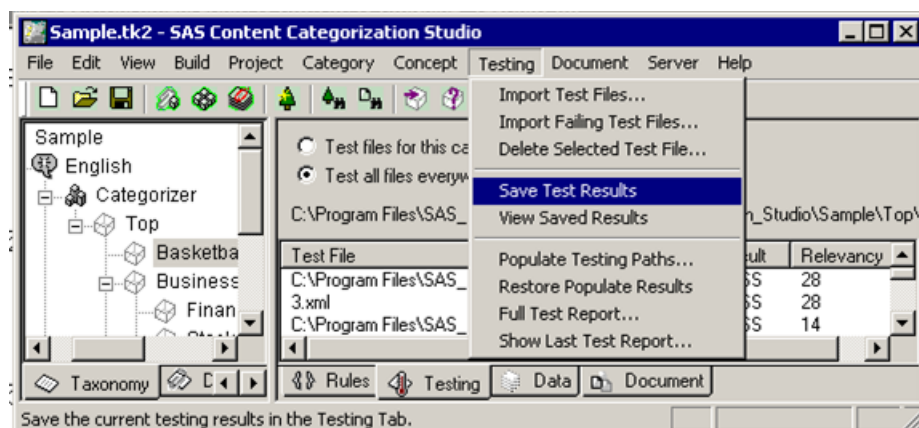
---

## 13.3 Save and Compare Test Results

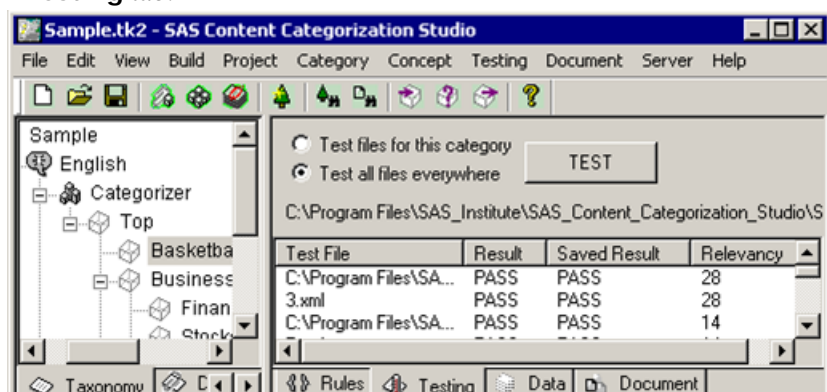
You can save and view your test results to compare them when you make rule or setting changes, by using the **Saved Result** column.

To see the **Saved Result** column, complete these steps:

1. After you test your testing directory, select **Testing --> Save Test Results**.



2. Repeat the testing process.
3. Select **View Saved Results** to see the **Saved Result** heading in the **Testing** tab.





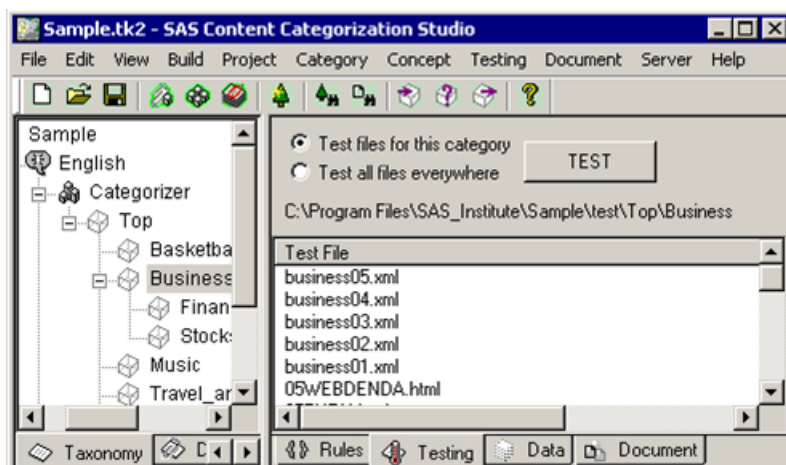
---

## 13.4 About Batch Testing

### 13.4.1 Option 1A: Batch Testing All of the Documents in One Category

To batch test a testing set of documents against the category that they are selected to match, complete these steps:

1. Create a testing taxonomy for your testing documents. For more information, see Section 12.2 *Creating Testing Folders* on page 350.
2. Select and assemble your testing documents. For more information, see Section 12.3 *Collecting Test Files* on page 359.
3. Set your testing paths. For more information, see Section 12.2.1 *Create a Testing Directory While You Set Paths* on page 350.
4. Populate the testing taxonomy. For more information, see Section 12.4 *Manually Populating a Testing Folder* on page 359.
5. Select a category to test in the Taxonomy window. For example, choose Business.

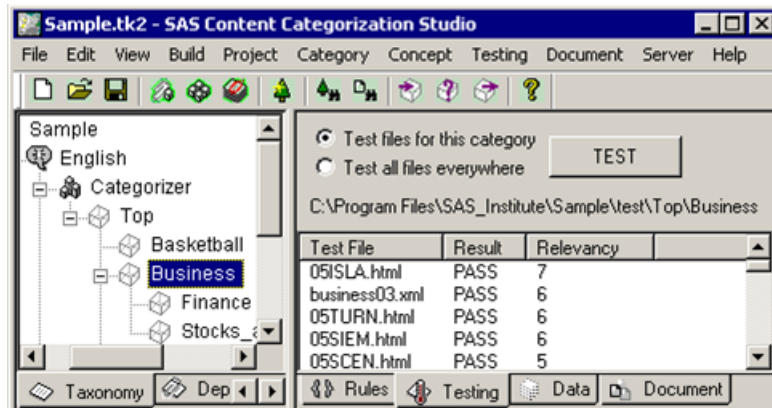


6. Click the **Testing** tab where the list of testing documents for this category is displayed under the **Test File** heading.

---

In order to ensure the accuracy of your test file location, the path to the testing directory appears above the **Test File** heading.

7. Click **Test files for this category**.
8. Click **TEST**. The testing and relevancy results appear in the Testing window.



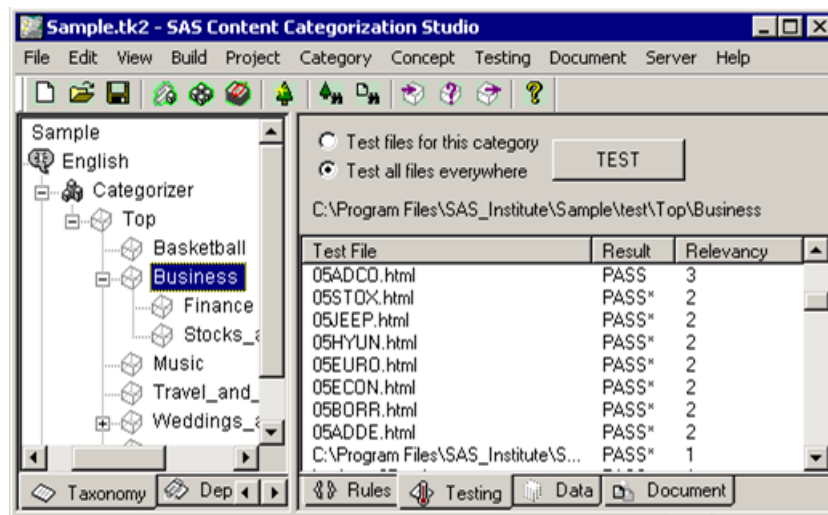
---

### 13.4.2 Option 1B: Batch Testing the Testing Taxonomy or Out-of-Category Files

Batch test the entire testing taxonomy to see how test files selected for other categories in the taxonomy perform.

To test all of the files in the testing directory, complete these steps:

1. Use Step 1 through Step 6 on page 379.
2. Click **Test all files everywhere**.
3. Click **TEST**.



The testing files fall into one of two types:

#### In-category files

These are the testing files that you assembled as optimal matches for the selected category. When these names are displayed in the Testing window, no paths to these files are displayed. Instead the path to this testing folder is shown above the **Test File** heading and below the **TEST** button.

#### Out-of-category files

---

Members of the testing folders that are selected to match other categories in your taxonomy are displayed with their full paths. When you test out-of category files, you could see multiple instances of each file. However, the testing results are the same.

4. (Optional) To reverse the testing document ordering, click the **Test File** heading.
5. Compare the testing results for both types of files.

### 13.4.3 Comparing Test Results

The testing results displayed in the Testing window for both *in-category* and *out-of-category* files enable you to compare the test results. These results provide a more comprehensive view of the appropriateness of your rules.

For example, if one of the passing documents for the Business category was matched to the Basketball category. Analyze the selected category rule for the purposes of understanding why this document matched. Also examine the Basketball rule and the matched document. One, or both, of these rules might be too broad.

If you double-click the matching Basketball document, this text appears in the Document window. Examine the matched terms in this window to gain a better understanding of why this document matched the Basketball category. For more information, see Chapter 14: *Testing with One Document*.

Conduct additional testing to evaluate whether the performance of other documents. Further testing could identify whether you should take one or more of the following actions:

- Narrow a category rule. For example, you can perform this operation by removing the term *Basketball* from the Business category rule.
- Broaden the category rule. For example, you can perform this operation by adding one or more of the terms that are used to define the Basketball category rule to the *Business* rule.
- Eliminate one, or more, of these categories from your taxonomy.
- Add categories to your taxonomy structure. For example, add a child node below the *Travel\_and\_vacations* category that is *Basketball\_vacations*.

---

**Note:** When you perform any of these operations, test your results after each step in the process. Rebuild the categorizer and save the project.

---

## 13.5 Remove a Testing File

You can remove one testing file from the Testing window when you take this step:

Select **Testing --> Delete Selected Test File** to remove the selected test file from the **Testing** tab.

When you clear a test document from the Testing window, the **Test File** field is empty and the `PASS` and `FAIL` messages in the Taxonomy window are removed.



---

# Chapter: 14

## Testing with One Document

---

- *Overview of Testing with One Document*
- *Test a Text in the Document Window*
- *Testing a Web Page in the Document Window*
- *See a Taxonomy of the Matching Nodes*
- *See the Best Matches*
- *Editing a Document in the Document Tab*

### 14.1 Overview of Testing with One Document

After you batch test a folder of testing documents against the category that these texts were selected to match, test one document. This operation provides more detailed testing information by enabling you to see the matching terms for the selected category within the document. When you test all of your documents in the Testing window, in contrast, you see a list of passing and failing texts.

You can also test this text against all of the categories or all of the categories and concepts in the taxonomy. In either case, you see what terms matched in the Document window. Use the match highlighting to see what changes should be made to the rules.

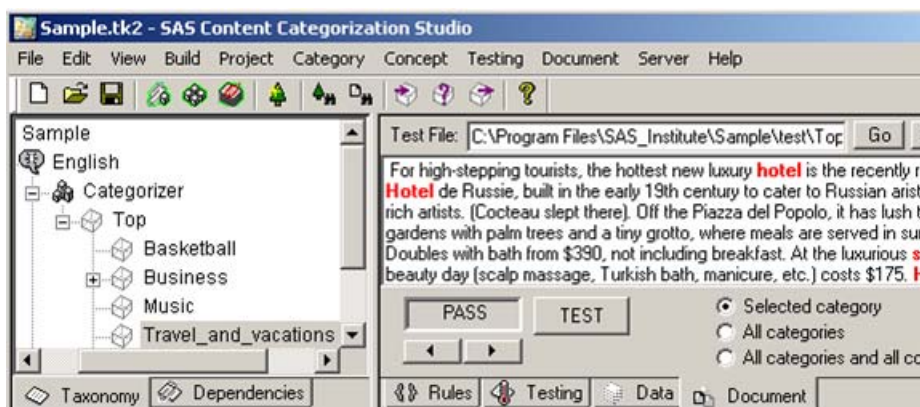
You can also use the Document window to load and test Web pages. If you load them as text, these pages appear in their source code and for this reason look like any HTML or XML page in the testing documents.



---

## 14.2 Test a Text in the Document Window

To test your documents in the Document window, complete these steps:

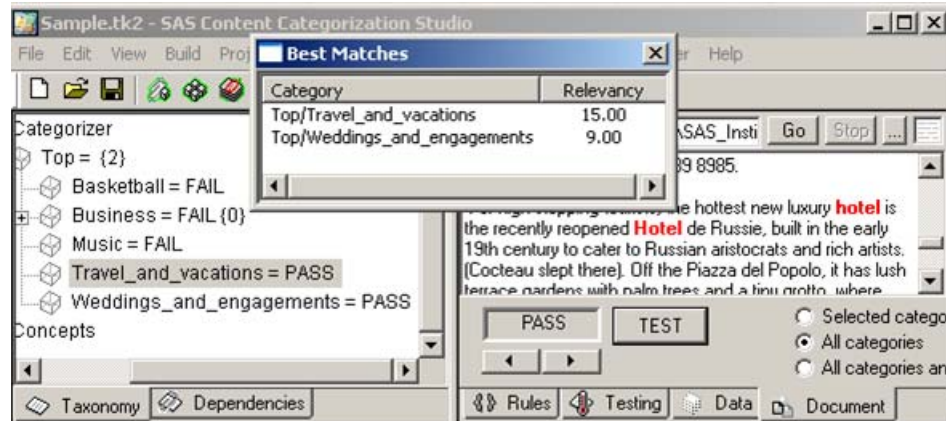
1. Double-click on a document in the Testing window and this text appears in the Document window.



2. By default you see the test results for the **Selected category** displayed in the document. Use the matching terms, highlighted in red, to see the words that made this document a passing text for the selected category.
3. Click the  and  to jump through each of the matches in the window.
4. (Optional) To remove the markup tags in an XML or HTML document, select **Document --> Remove Tags**. If you perform this operation, click **TEST** to see the tags reinstated.
5. A PASS or FAIL message for this text appears in the blank field to the left of the **TEST** button. Status messages are also displayed in the Taxonomy window, when the document is retested, if you select **All categories** or **All categories and concepts**.



The Best Matches window appears when you select **Edit --> Options** and select **Show best matches when testing all**. In addition, select either **All categories** or **All categories and all concepts**.



6. (Optional) Use the Best Matches window to see the matching nodes under the **Category** heading. You can also see the relevancy score for each passing category under the **Relevancy** heading.

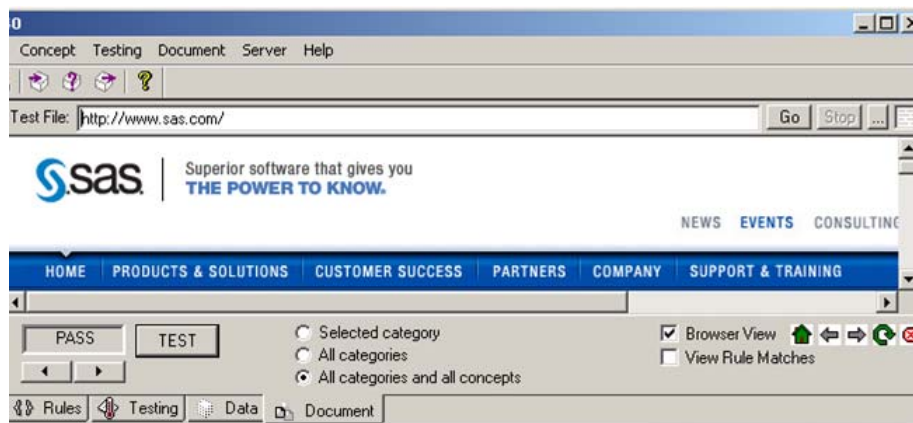
---

## 14.3 Testing a Web Page in the Document Window

### 14.3.1 Choosing Browser Operations

Use the Document window to view Web pages. Also use this window to access operations that are specific to a Web browser such as viewing, testing, and so on. Select **Browser View** to access these operations.

*Display 14-1 Web Page in Browser View*



---









**Note:** Web pages are tested in their source format.

---

---

The table below explains the operations that can be used to see and test your Web pages using the browser in the Document window:

Table 14-1: Browser Operations

Operation	Description
Test File	Type the URL for the Web page that you want to test into this field.
	Click <b>Go</b> to load the Web page that you specified in the <b>Test File</b> field.
	Click <b>Stop</b> to prevent a Web page from loading.
	See whether the Web page is loading using this window.
Browser View	See a Web page in the browser mode.
Remove Tags	Choose this selection to see the Web page in text view without any markup language.
	Click <b>Home</b> to return to the home page.
	Click <b>Back</b> to go to the last Web page that you visited.
	Click <b>Forward</b> to go to the next page.
	Click <b>Refresh</b> to update the Web page.
	Click <b>Stop</b> to end the loading process.

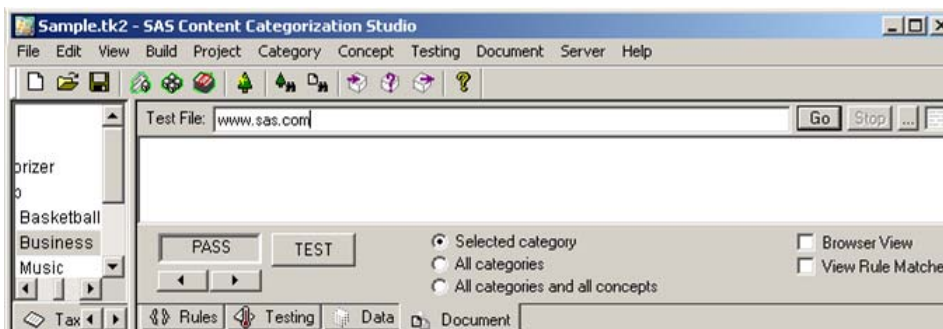
---

## 14.3.2 Load and Test the Source Document

The browser feature of the Document window enables you to load and test a Web page as a text document.

To test a Web page as a text document, complete these steps:

1. Select a category in the Taxonomy window. For example, choose *Business*.



2. Click the **Document** tab.
3. Type the URL of the Web page that you want to test into the **Test File** field.
4. Click **Browser View**.
5. Click **TEST**. The results of the testing operation appear in the source document.



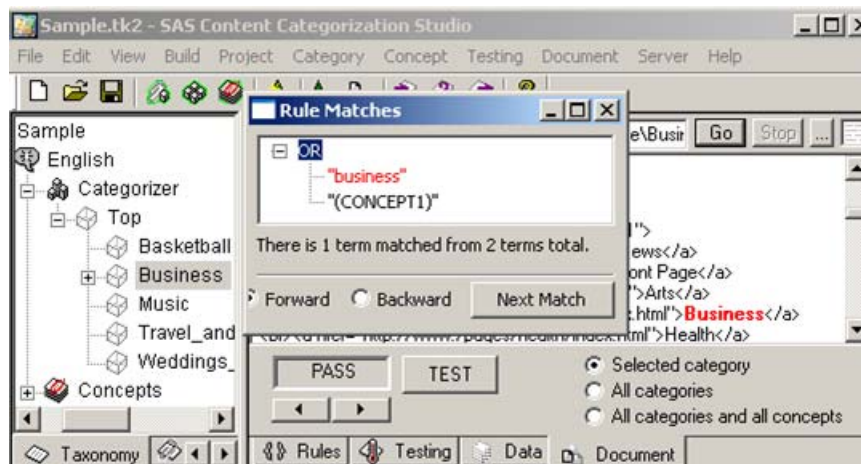
---

## 14.4 See a Taxonomy of the Matching Nodes

The Rule Matches window displays the matching Boolean rule terms for the selected category. Each of the matches is highlighted in red and the entire rule is displayed in rule tree format. Use this window to quickly see a list of matched terms.

To see the matching Boolean terms, complete these steps:

1. Select a category that is defined by a Boolean rule. For example, select Business.
2. Click **TEST** in the **Testing** tab to test the documents for the selected category.
3. Double-click a test document and it appears in the **Document** tab.
4. Select **View Rule Matches**.
5. Select **Selected category**, if this default operation is not selected.
6. Click **TEST**. The Rule Matches window appears.



7. See the matching terms, highlighted in red, in the Rules Matches window. Unmatched terms and dependencies, for example CONCEPT1 appear in black lettering.

- 
8. See the statement below this pane that explains the returns shown in the **Taxonomy** tab. For example, *There is 1 term matched from 2 terms total.*
  9. Click **Forward** if you want to see the next matching term in the taxonomy.
  10. Click **Backward** if you want to see the last matched term that you viewed.
  11. Click **Next Match** to use either the **Forward** or **Backward** operations.
  12. Click **X** in the Rules Matches window to close this window.

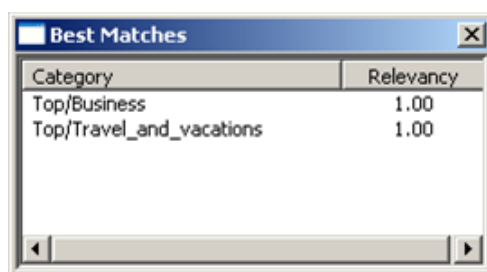
---

## 14.5 See the Best Matches

Use the Best Matches window to see all of the categories, or all of the categories and concepts, matched by the selected document.

To see the best matches for an input document, complete these steps:

1. Select **Edit --> Options**. Select **Show best matches when testing all** in the Options window that appears.
2. Select **Build --> Build Rulebased Categorizer**.
3. Test a document in the **Document** tab.
4. Select **All categories**.
5. Click **TEST** and the Best Matches window appears.



6. See the relative path to the named category specified under the **Category** heading.
7. Compare the relevancy score of the tested documents using the numbers listed beneath the **Relevancy** heading.

If you select **All categories and concepts** in the **Document** tab, the Best Matches window also displays the names, paths, and relevancy scores for all matching nodes.

---

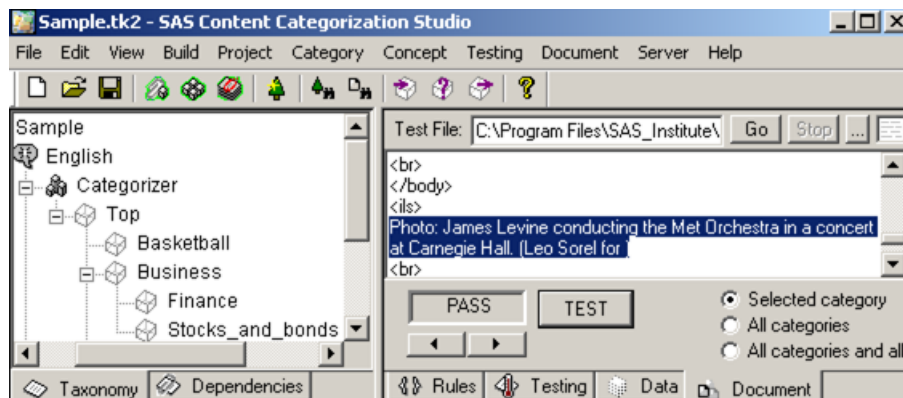
## 14.6 Editing a Document in the Document Tab

### 14.6.1 Choosing Windows Commands

#### 14.6.1.A Delete and Replace Text

To remove text from your testing document, complete these steps:

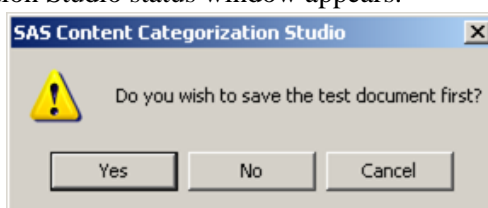
1. Double-click on a document in the **Testing** tab and the contents of the text appear in the **Document** tab.



2. Highlight the text that you want to delete using either your cursor or Ctrl A on your keyboard.
3. Click **Delete**.
4. (Optional) Type in the words that you want to add to the document.
5. Click **TEST** to see whether you obtain the results that you require.



- 
- (Optional) Click the **Testing** tab and click **TEST**. The SAS Content Categorization Studio status window appears.



If you click **Yes**, the changes that you made to the testing document are preserved in that file.

- When you leave the Document window and try to test another document, a SAS Content Categorization Studio confirmation window appears.



- Click **Yes** if you want to save the changes that you made to the test document in the Document window.

#### 14.6.1.B Copy and Paste a Test File

You can copy and paste a test file directly into the Document window. Use this operation if you want to test a text without including it in the test file folder.

To copy and paste a test file, complete these steps:

- Access the Document window.
- Access another document that you want to test in the application of your choice.
- Highlight the text that you want to test.
- Copy this text, or the whole document, and paste it into the Document window using the Ctrl V command.
- Click **TEST**.

---

## 14.6.2 Clear a Test Document

When you select the **Document --> Clear Test Document** operation, the document that currently appears in the Document window is removed from the Document window. However, this text is not deleted from the list that appears in the Testing window or from the testing folder.

## 14.6.3 Refreshing the Taxonomy Tree

Refresh your taxonomy tree when you want to retest your document by deleting all of the `PASS` and `FAIL` messages that appear in the Taxonomy window. These messages appear after you test a text using either the **All categories** or **All categories and all concepts** radio buttons in the Document window.

To delete the `PASS` and `FAIL` messages in the Taxonomy window, click the **Refresh Tree** button, or access a new document in the Document window.

## 14.6.4 Changing the Font Size of a Tested Document

You can choose to increase or decrease the size of the text that is displayed in the Document window. These operations can make it easier to see the matching terms within their context.

---

**Note:** You can decrease the size of the letters in the document only after you have increased their size.

---

To increase the font size select **Testing --> Increase Font Size**.

To decrease the font size, select **Testing --> Decrease Font Size**.

## 14.6.5 Removing Markup Tags

To see an HTML, or an XML, document as a text without any markup tags, select **Document --> Remove Tags**. The testing document in the Document window is displayed as a text document without any markup language.

---

# Chapter: 15

## Other Testing Operations

---

- *Overview of Other Testing Operations*
- *Test a Central Repository*
- *Import Failing Documents*
- *About the Graphical Reports*
- *About the Full Test Report*

### 15.1 Overview of Other Testing Operations

This chapter provides information about testing operations that complete the suite of possible category tests. Use this chapter to test a repository that represents the types of texts that you plan to categorize using SAS Content Categorization Server. This central repository does not contain folders with texts that are selected to match the concepts in your taxonomy. Instead, all of the testing files for all of the taxonomy nodes reside in a single folder.

You can also use this chapter to test documents that should fail to match a specific category, but do not. For example, the occurrence of the term *card stock* in an input document should not return a match for the *stocks and bonds* category. Run a full test report, interpret its results, and put a hold on a category to exclude it from the exported `.mco` file.

---

## 15.2 Test a Central Repository

There are two ways to test against a central repository of testing documents. You can choose to test all of the testing files in one folder against a single category. You could also select **Test all files everywhere** in the **Testing** tab and click **TEST** to perform this operation.

These testing examples enable you to see how your definitions perform against a wide variety of documents. This process can help you to identify documents that do match but should not. For more information, see Section 15.3 *Import Failing Documents* on page 402.

Use the following two sections to perform both of these operations.

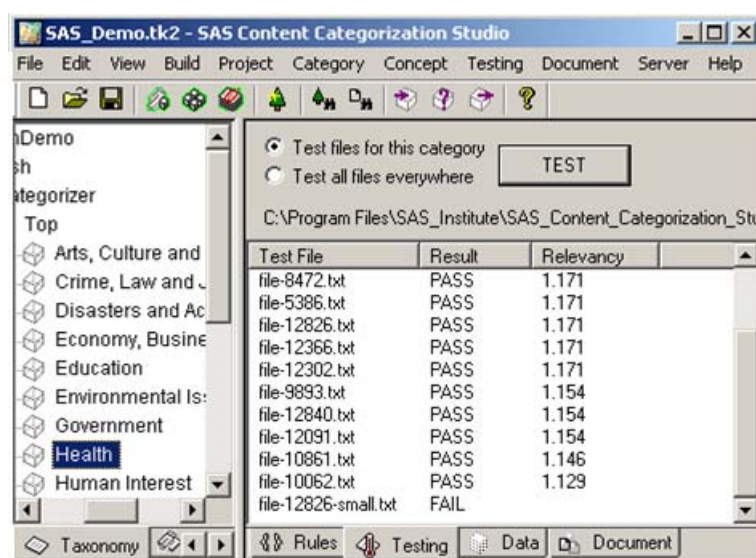
### 15.2.1 Test against a Single Testing Folder

Before you can test a category against all of the files in a single folder, locate the folder. This operation assumes that these test files are not matched to the category that they are tested against. Use this testing operation against a folder of files where you do not expect all of the files to match.

To test against a single folder, complete these steps:

1. Use the steps in Section 12.2.2 *Create and Set a Path to the Central Repository* on page 355.

2. Select a category in the Taxonomy window and click the **Testing** tab.



When you select any category in the taxonomy, the **Testing Path** specification in the Data window is identical to the path for the **Top** directory. The list of testing texts is also identical.

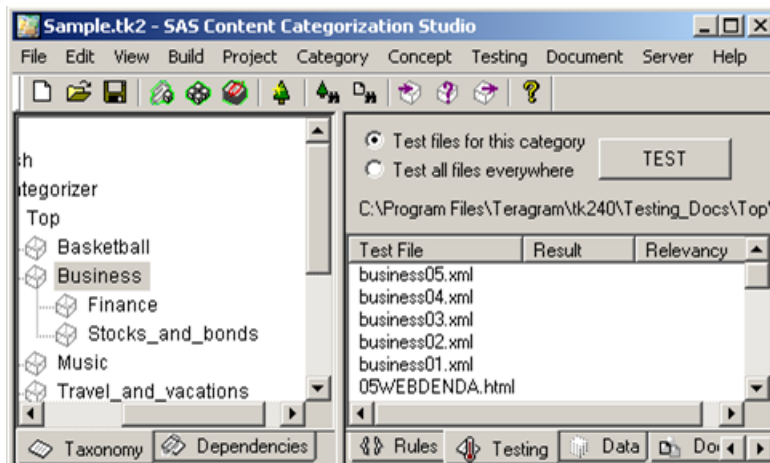
3. Click **TEST**. See the results in the Testing window.
4. (Optional) Repeat Step 3 above for each category that you want to see the testing results for

## 15.2.2 Test Against a Central Repository

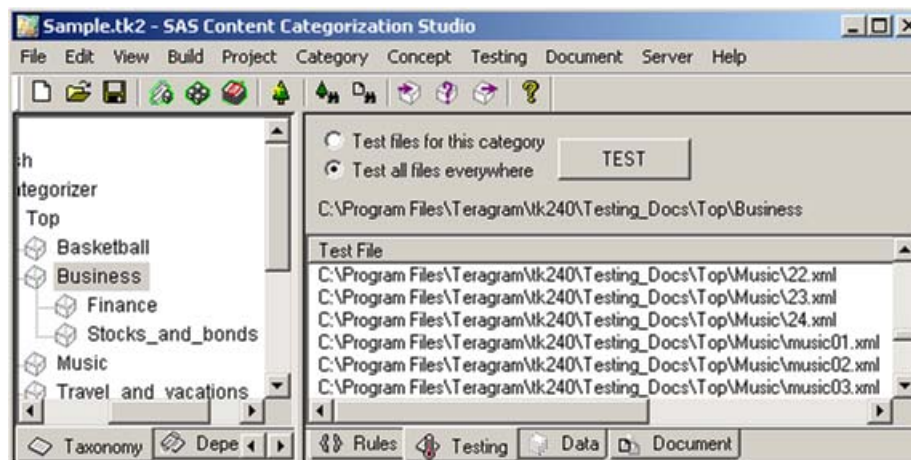
When you test against the central repository, you test against all of the documents that are assigned to each of the taxonomy nodes. This testing operation, like the operation explained in Section 15.2.1 *Test against a Single Testing Folder* provides a real world example.

To test against a central repository of testing documents, complete these steps:

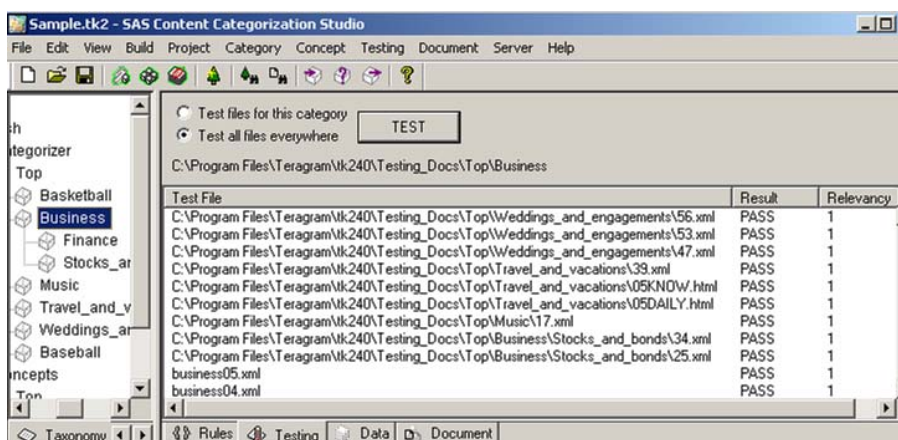
1. Select a category in the Taxonomy window.



2. Click the **Testing** tab.
3. Click **Test all files everywhere**. See the test files from all of the testing paths displayed in the Testing window.



4. Click **TEST** to see the testing results.



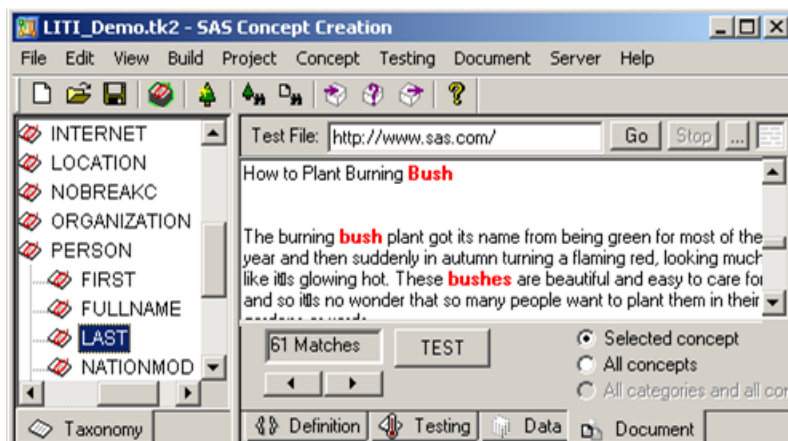
5. (Optional) Access each passing file in the central repository in the document window to see the matching terms. For more information, see Section 14, *Testing with One Document*.

---

## 15.3 Import Failing Documents

During testing, you might discover that certain test documents should not be matched to a specific category. For example, landscaping texts that contain the word *bush* should not match the President Bush category.

*Display 15-1 Failing Document Example*



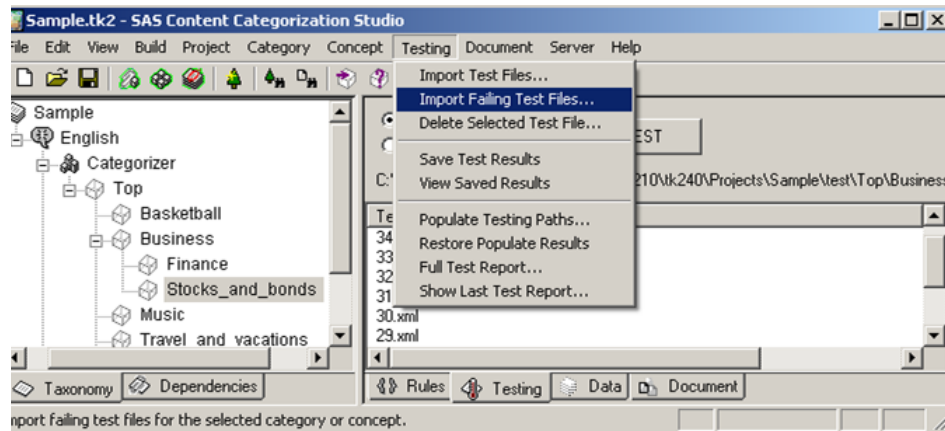
In the example provided above, the passing document entitled *How to Plant Burning Bush*, contains the word *bush* in the context of a plant. This is an example of a document that you do *not* want to pass the test for the Presidents category where one of the qualifying terms is *bush*. In this taxonomy, the term *bush* should match documents that mention George H. W. or George W. Bush, only.

As you test and define category rules, copy documents that should fail, but are not, into a Fail directory. You can then test this directory.

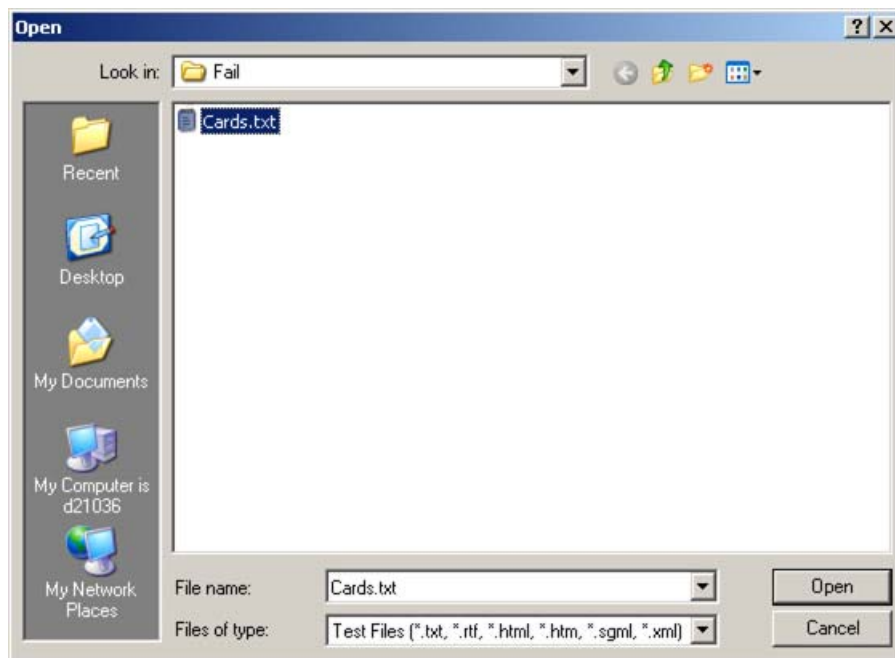
To test documents in the Fail directory, complete these steps:



1. Click the **Testing** tab.

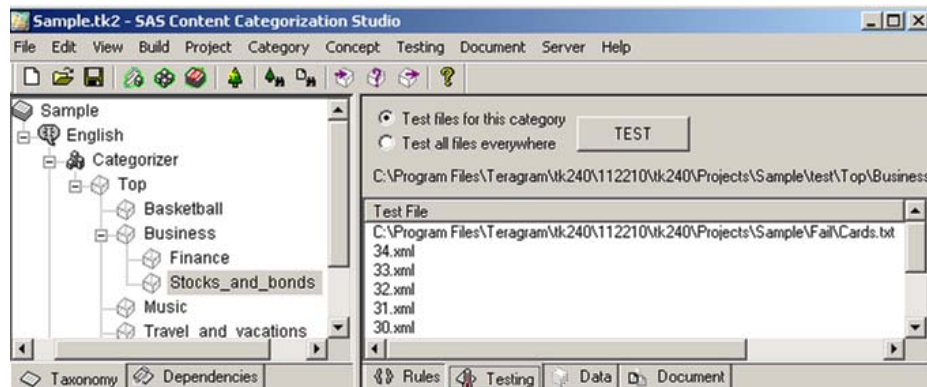


2. Select **Testing --> Import Failing Test Files**. The Open window appears.



3. Select a file. For example, choose `Cards.txt`.

- Click **Open**. The failing testing document appears in the Testing window preceded by its path.



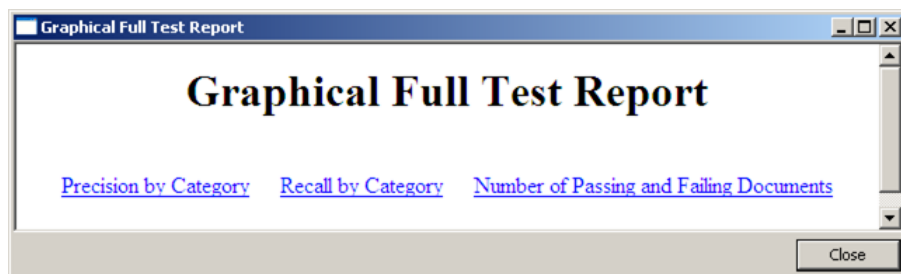
- Click **TEST** to see whether this file fails, or whether you need to make further rule adjustments.

## 15.4 About the Graphical Reports

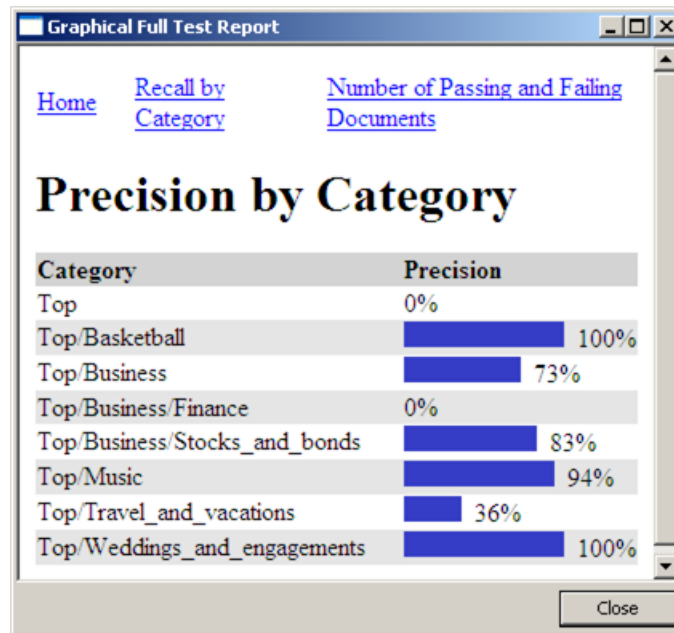
Use the graphical reports to see the statistics for category matches. You can see the precision, recall, and numbers of passing and failing documents in these reports.

To access and use the Graphical Full Test Report pages, complete these steps:

- Select **Testing --> Graphical Full Test Report**. The Graphical Full Test Report page appears.

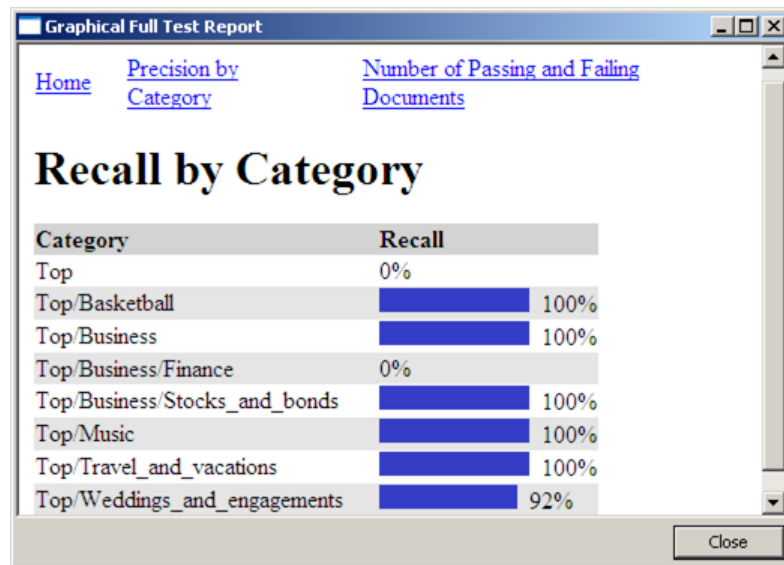


2. Click **Precision by Category**. The Precision by Category page appears.

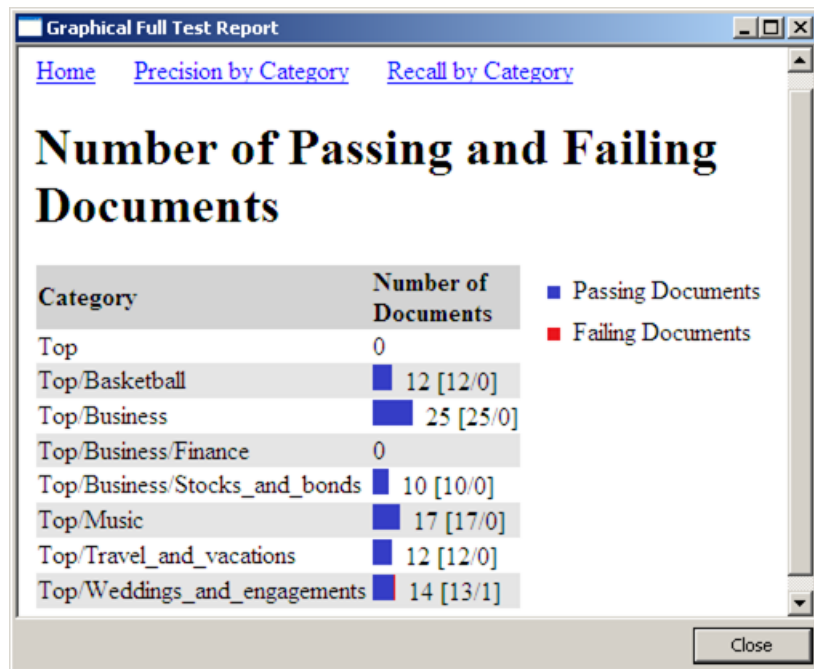


3. (Optional) Click the **Category** heading to display the categories alphabetically starting from the letter Z down or from the letter A down.
4. (Optional) Click the **Precision** heading to display the results starting from the 0%, or from 100%, down.

- 
5. Click **Recall by Category**. The Recall by Category page appears.



6. (Optional) Click the **Category** heading to display the categories alphabetically starting from the letter Z down or from the letter A down.
7. (Optional) Click the **Recall** heading to display the results starting from the 0%, or from 100%, down.



8. See the number of **Passing Documents** in blue and the number of **Failing Documents** in red.
9. (Optional) Click the **Category** heading to display the categories alphabetically starting from the letter Z down or from the letter A down.
10. (Optional) Click the **Number of Documents** heading to display the results starting from the 0%, or from 100%, down.
11. Click **Close**.
12. (Optional) Click **Testing --> Show Last Full Graphical Testing Report** after you close this report. This operation restores the last report.

---

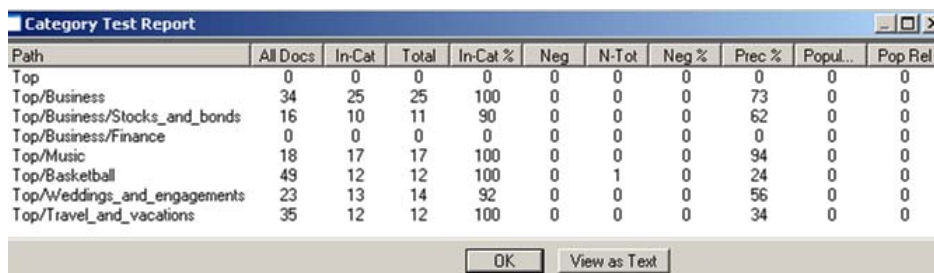
## 15.5 About the Full Test Report

### 15.5.1 Completely Test the Categorizer

After you develop and test your taxonomy, run a complete test of the categorizer to determine the precision of its rules. Use the generated report to determine whether your rules return the anticipated results or it is necessary to make changes. For example, if you are using documents that you selected to match the categories in your taxonomy, you could expect a 100% matching rate. If you do not see this percentage, you might need to adjust your rules.

To generate a full test report, complete these steps:

1. After you develop and test every node in the taxonomy against a set of testing documents, select **Build --> Build Rulebased Categorizer**.
2. Select **Testing --> Full Test Report**.
3. The **Category Test Report** window appears displaying the testing results.



Path	All Docs	In-Cat	Total	In-Cat %	Neg	N-Tot	Neg %	Prec %	Popul...	Pop Rel
Top	0	0	0	0	0	0	0	0	0	0
Top/Business	34	25	25	100	0	0	0	73	0	0
Top/Business/Stocks_and_bonds	16	10	11	90	0	0	0	62	0	0
Top/Business/Finance	0	0	0	0	0	0	0	0	0	0
Top/Music	18	17	17	100	0	0	0	94	0	0
Top/Basketball	49	12	12	100	0	1	0	24	0	0
Top/Weddings_and_engagements	23	13	14	92	0	0	0	56	0	0
Top/Travel_and_vacations	35	12	12	100	0	0	0	34	0	0

OK View as Text

4. (Optional) Click **View as Text** to see this report in Notepad where you can print it and so on.
5. Click **OK** to close this report.

---

## 15.5.2 Interpreting the Report Statistics

Use the full-test report as both a reporting and an analysis tool. As you analyze the displayed results, pay particular attention to the **All Docs**, **In-Cat %**, and **Prec %** columns:

### **All Docs** column

The figures in this column represent the total number of texts that matched this category in the test process. If there is a large discrepancy between the numbers reported here and in the **Total** column, your rule might be too broad. This discrepancy would indicate that texts selected as matches for other categories are also matching this category.

### **In-Cat %** column

The number of testing documents that matched this category out of all of those matched to it, is listed here. A number below 90% means that you are either providing inappropriate test documents or that you need to refine your category rules.

### **Prec %** column

The precision percentage for each category is displayed here. This number is a comparison between the **In-cat** number and the **All Docs** number. The **In-cat** number is the number of category test documents that passed. The **All Docs** number is the number of all of the testing documents for all of the categories that passed. For example, an **All Docs** figure of 200 and an **In-Cat** figure of 10 means that 10 of the category's test documents passed. 190 test documents from other categories also passed. Precision is 5%.

The accuracy of precision depends on the actual subject matter of the documents, the taxonomy, and your requirements. For this reason, a low precision percentage should be investigated, but a low number does not necessarily indicate imprecise rules. For example, a document about the new health care law could match both the Health care and Politics categories.

For more information about the other columns, see Section 2.12.12 *The Full Test Report Window* on page 87.

Examine the rules for each category that has unexpected results, and revise and retest as necessary. You can interactively create and test one or more individual documents to help you locate necessary changes.





---

## Part 3: Concepts

---

- Chapter 16: *Developing a Concepts Taxonomy on page 413*
- Chapter 17: *Defining Concepts on page 425*
- Chapter 18: *Writing Classifiers on page 441*
- Chapter 19: *Writing Grammar Rules on page 469*
- Chapter 20: *Testing Concepts on page 491*



---

# Chapter: 16

## Developing a Concepts Taxonomy

---

- *What is a Concept?*
- *Determining How to Extract Concepts*
- *Planning Your Taxonomy Structure*
- *Create the Taxonomy Structure*
- *Changing the Concepts in a Taxonomy*

### 16.1 What is a Concept?

A concept is defined as a piece of information. For example, a name or a place can be a simple concept. A relational concept, on the other hand, is defined as two or more terms that are identified as related to one another. For example, see United States President Barack Obama.

Concepts extraction is a key feature of SAS Content Categorization Studio. Use this technology to extract metadata from input documents, whether the data that you want to return is known or not. For example, you might want to return all of the names of the presidents in all of the countries in the world.

This chapter lays the necessary foundation for the following chapters in this section that enable you to identify the key information that you require.

---

## 16.2 Determining How to Extract Concepts

Before you can create a taxonomy, or concepts structure, consider the concepts that you want to match in your input documents. To identify these concepts, complete these steps:

1. Analyze your documents to understand the types of information that you want to return. For example, you might want to locate the names of people, organizations, and their titles.
2. Assess the needs of your end users. For example, is it sufficient to provide a list of names? Would better results be returned if these names are related to a company, or to each individual's position in that company?
3. Choose the names of the concepts in your taxonomy to reflect the type of information that you want to match in input documents. For example, create an Organizations concept to extract general organizational information.
4. Select either a flat, or a hierarchical, taxonomy structure based on whether you want to define subnodes for some concepts. Using the Organizations example above, define subcategories that include Trade, Finance, and Farming departments.
5. Choose to use classifier concepts that are lists of terms with, or without, regular expressions to identify simple concepts that are either known or unknown to you. For example, use classifier definitions to specify a list of terms that are movie names. Use regular expressions to locate information that is unknown, but follows a known pattern. For example, regular expressions can be used to identify e-mail addresses, phone numbers, or street addresses. Use grammar concepts to identify previously unknown data by writing rules that use parts of speech to identify entities. For example, use parts of speech to identify proper nouns such as movie producers or company executives.

Together, these steps form an important background planning component for your project.

---

## 16.3 Planning Your Taxonomy Structure

Before you define your concepts, develop a taxonomy that serves as the prototype for the concepts part of your project. Choose a set of names for your concepts and a taxonomy structure that is either flat or hierarchical. A flat taxonomy does not have children, but a hierarchical taxonomy does.

The following is an example of a taxonomy consisting of three concepts `Books`, `Business`, and `Music`. The two child concepts for the `Business` concept are `Finance` and `Stocks and bonds`:

```
Books
Business
  Finance
  Stocks and bonds
Music
```

Unlike category names, each concept name is unique for any given language. This rule applies to both flat and hierarchical taxonomies. In other words, you cannot define a `Finance` child concept under the `Books` parent concept.

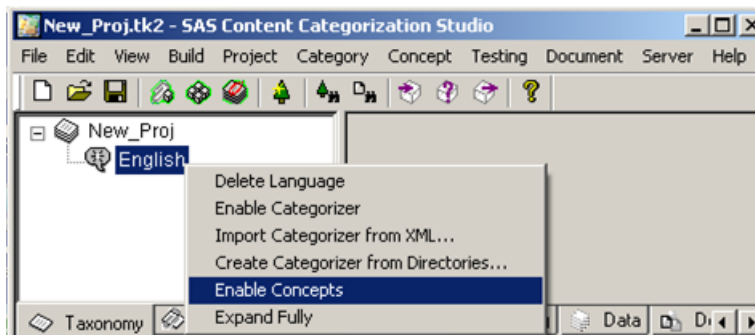
When you develop the concepts taxonomy, it is important to consider the definitions that you plan to use to define each concept. You should also consider any potential interrelationships that could affect these definitions.

---

## 16.4 Create the Taxonomy Structure

To develop a concepts taxonomy, complete these steps:

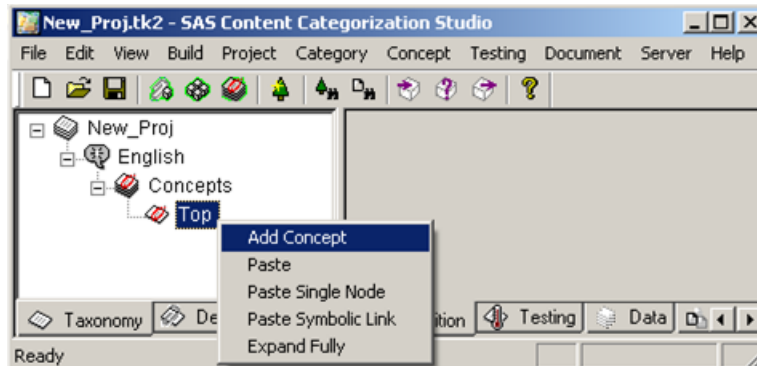
1. Right-click on the language node. For example, choose *English*. Select **Enable Concepts** from the drop-down menu that appears.



Two nodes are added to the taxonomy. These are the *Concepts* and *Top* nodes.



- 
2. Right-click on the **Top** node and select **Add Concept** from the drop-down list.



3. Type the name of the concept into the box that appears around the new node.
4. Use Step 1 on 416 through Step 3 above, reiteratively, until the taxonomy is complete.
5. (Optional) Select **Build --> Compile Concepts**. If you selected **Always rebuild before each test** in the Options window, this step is not necessary.

---

## 16.5 Changing the Concepts in a Taxonomy

### 16.5.1 Recompile Concept Changes

Recompile the concepts whenever you make a change that affects the taxonomy. When you are in the process of building a taxonomy, you should recompile after making additions, deletions, and changes to the existing taxonomy.

To recompile your concepts, select **Build --> Compile Concepts**.

---

**Hint:** If you make a change to a concept definition, click **Syntax Check** in the Definition window to check the definition.

---

### 16.5.2 About Moving Concepts

The following pointers are important to understand before you move any of your concepts:

- Concepts cannot be moved to the categorizer branch of the taxonomy.
- Concepts can be moved to become either parents or children, regardless of their original location.
- The moved concept retains its name, definition, and the associated metadata that is specified in the Data window. This is true unless you change this information.
- If you move a dependent concept, you might need to rewrite the definition of the referring concept. For more information, see Section 19.5.4 *Defining Dependencies in Grammar Rules* on page 487.



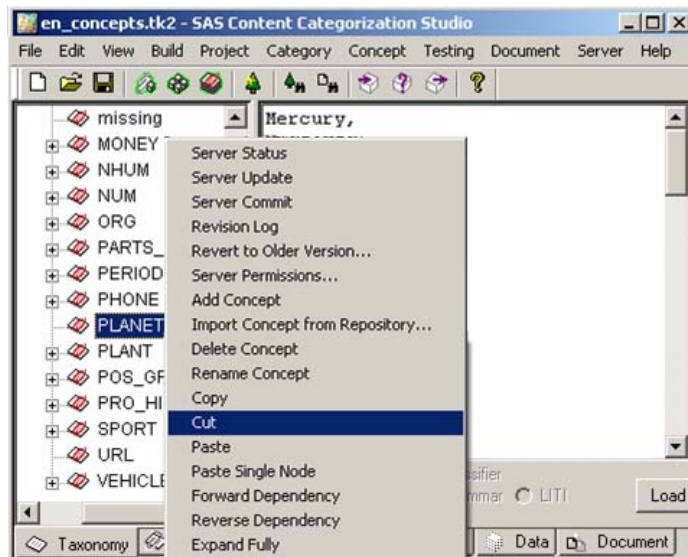
---

### 16.5.3 Move a Concept

You can move a concept in order to make a parent node a child node, or to make a child a parent.

To move a concept, complete these steps:

Right-click on a concept in the Taxonomy window and select **Cut** in the menu that appears.



Right-click the concept that becomes the parent of the cut concept and select **Paste** from the menu that appears.



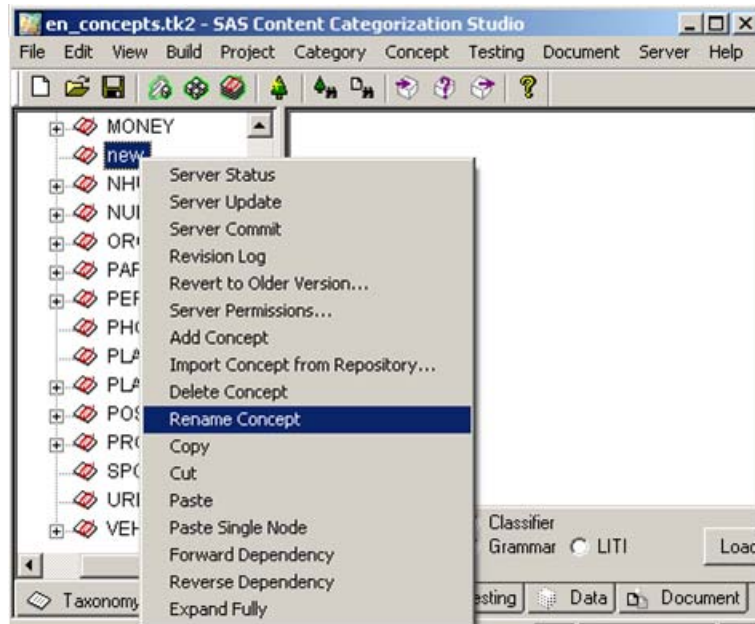
---

The concept is pasted into its new location in the taxonomy.

### 16.5.4 Rename a Concept

To rename a concept, complete these steps:

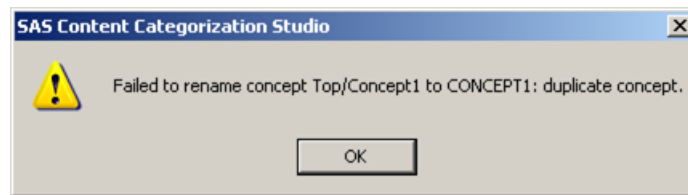
1. Right-click on a concept and select **Rename Concept** from the drop-down list that appears.



2. Type the new name of the concept into the box that encloses the concept name. For example, enter NAMES.

---

If the renamed concept has the same name as an existing concept, a SAS Content Categorization Studio status window appears. See the example below:



Click **OK**.

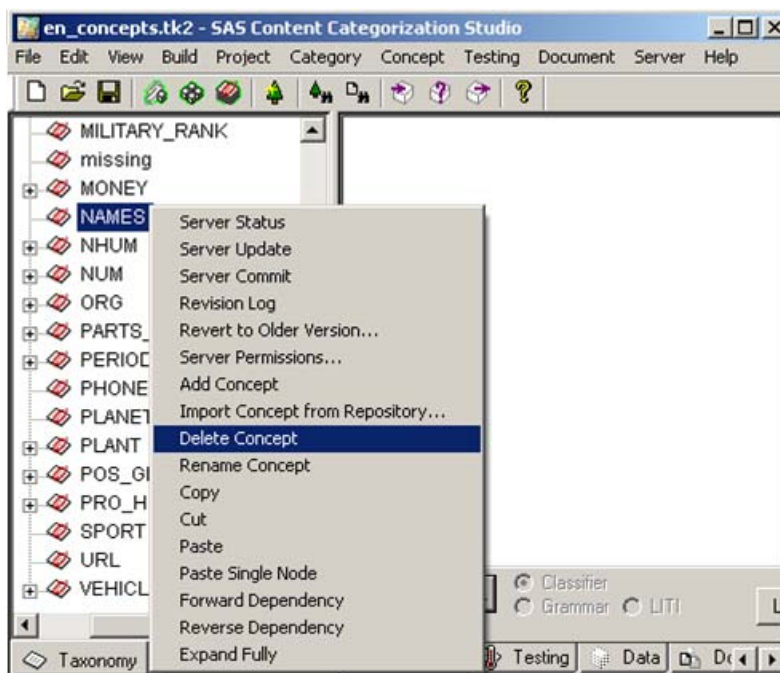
3. Click elsewhere in the **Taxonomy** tab to make name change take effect.

### 16.5.5 Delete a Concept

Before you delete a concept, check the Dependencies window to see whether any other concepts have a forward dependency on this concept. For more information about dependencies, see Section 19.5.4 *Defining Dependencies in Grammar Rules* on page 487.

To delete a concept, complete these steps:

Right-click on the concept that you want to remove. In the drop-down menu that appears, select **Delete Concept**.



Click **Yes** in the SAS Content Categorization Studio confirmation message that appears.



The concept no longer appears in the **Taxonomy** tab.

---

## 16.5.6 Copy and Paste Concepts

The Copy and Paste operations enable you to create new concepts with the same name, metadata, and rules. These operations make it easy to edit your concepts.

To create a concept using the copy and paste operations, complete these steps:

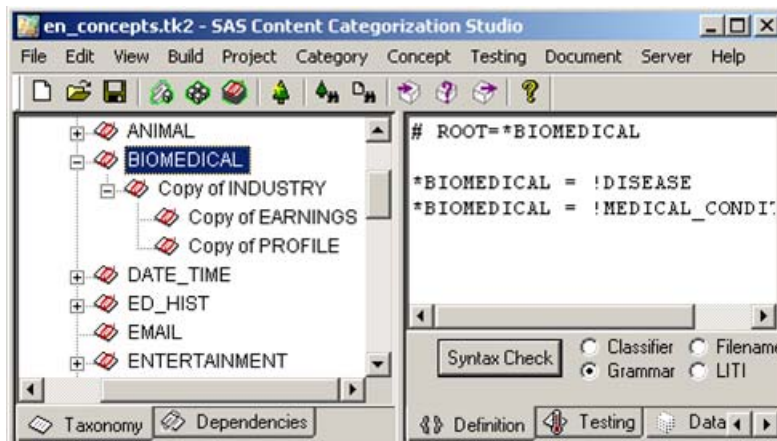
1. Right-click on a concept in the Taxonomy window. For example, select **Industry**. Select **Copy** from the drop-down menu that appears.



2. Right-click on another node in the Taxonomy window. For example, select **Biomedical**. Select **Paste** from the drop-down menu that appears. The **Industry** concept and its children are pasted below the

---

Biomedical concept. Each new node has the term *Copy of* appended to its name.



3. (Optional) Click the **Paste Single Node** component to add the parent concept, only, to your taxonomy.
4. Right-click on each of the copied concepts and type in their new names.
5. Edit the definition and metadata for each concept.

---

**Note:** If you repeat the copy and paste operations the words *Copy of* are appended to the name of the concept each time. For example, *Copy of INDUSTRY* becomes *Copy of Copy of INDUSTRY* that becomes *Copy of Copy of Copy of INDUSTRY*.

---

---

# Chapter: 17

## Defining Concepts

---

- *Overview of Defining Concepts*
- *Determining the Match Criteria*
- *Understanding Concept Types*
- *Use the Sample Concepts Project*
- *Write a Definition*
- *Use the Syntax Check Button*
- *Compile Concepts*

### 17.1 Overview of Defining Concepts

Concepts are the entities that SAS Content Categorization Studio extracts from input documents. These concepts identify the metadata that is located in your input documents. Metadata is information about information.

You can choose between two types of concepts. You can also specify various other settings to match concepts according to rule matches and other specifications.

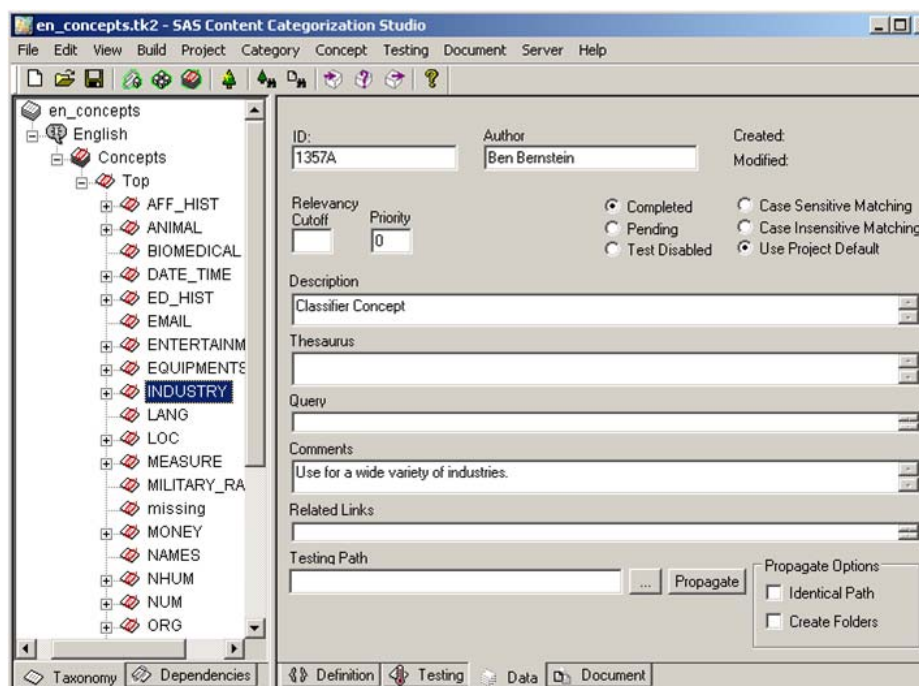
---

## 17.2 Determining the Match Criteria

### 17.2.1 Provide Identifying Information for Your Concepts

To specify information for a concept, complete these steps:

1. Click the **Data** tab.



2. (Optional) Type in the **ID** number. This is the unique identification number for this concept. If you want to enable duplicate identification numbers for your concepts, select **Allow Duplicate ID's** in the Project Settings - Concept window.
3. (Optional) Enter the name of the person creating the concept into the **Author** field.

The **Created** and **Modified** fields are automatically filled in for you.



- 
4. (Optional) Specify a **Relevancy Cutoff** setting if you want to override the **Default Relevancy Cutoff** setting in the Project Settings - Concept window that is set to zero (0) by default.
  5. (Optional) Specify a **Priority** setting and a match on this concept supersedes another match. This is true if an input document matches two or more concepts and no other determinant makes one concept a better match.
  6. Select a radio button to indicate the status of the concept definition:

**Completed**

(Default) Includes the concept in the compile and testing processes.

**Pending**

Specifies that this concept is incomplete and not included in the compile process. This specification does not affect the `<language>.concepts` file.

**Test Disabled**

Evaluates the selected concept but returns no matches for this concept. Other concepts can reference this concept by name using dependencies or symbolic links. Use this operation to build a taxonomy with helper concepts that are not exposed to the user. When you select this operation, the selected concepts node in the **Taxonomy** tab appears in a lighter font than normal.

No testing can be performed on this concept. These concepts are often used as helper concepts. For more information, see Section 5.7 *Evaluating a Referenced Category* on page 179.

7. Select the type of matching that is performed on classifier terms, only. If you select either of the first two selections, your choice overrides the default setting specified in the Project Settings - Concept window:

**Case Sensitive Matching**

Matches can occur only when the term that is located is an exact match on the case of the definition term.

**Case Insensitive Matching**

Matches can occur on terms where the case of the term is not a match.

**Use Project Default**

---

(Default) Use the case-sensitive setting is specified in the Project Settings - Concept window.

8. (Optional) Explain the concept in the **Description** field.
9. (Optional) Type any notes for this concept into the **Comments** field.

---

**Note:** The **Created** and **Modified** fields are automatically filled in for you.

---

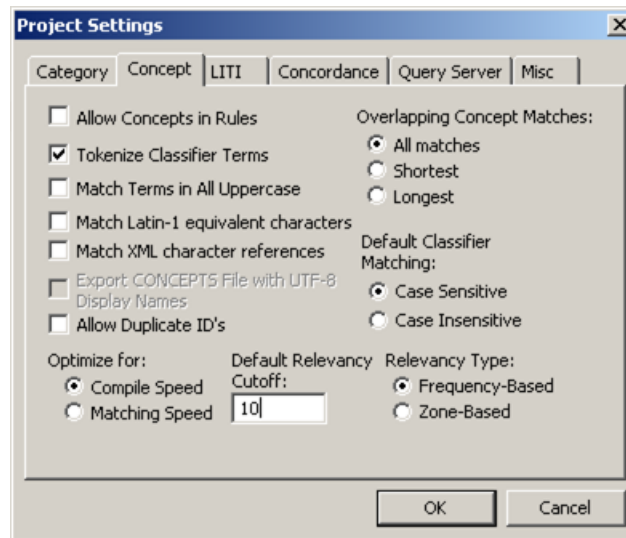
---

## 17.2.2 Specifying the Project Settings

Choose the settings that you require to define and test your concepts. Many of these specifications are set in the Project Settings - Concept window where they apply to the entire taxonomy of concepts.

To access and use the Project Settings - Concept window, complete these steps:

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



2. Select **Allow Concepts in Rules** to use classifier concepts in category rules in order to create dependencies. This check box is available only when you enable both categories and concepts.
3. (Default) Deselect **Tokenize Classifier Terms** if you do not want to enable SAS Content Categorization Studio to automatically break the definition text of classifier concepts into words. The default setting should always be maintained for new projects.

---

**Note:** Turn off this operation if you choose to use a backslash (\) instead of a space between terms in a concept definition.

---

4. (classifier concepts only) Select **Match Terms in All Uppercase** to add all uppercase versions of the specified rule terms to the classifier rule. For example, a rule containing the word *Cat* adds *CAT* to the concept rule.
5. (classifier concepts only) Select **Match Latin-1 equivalent characters** when you input documents in Latin-1 languages that contain accented characters. Choose to match the Latin-1 equivalent characters as if they were unaccented. For example, match *cana* as if it were *caña*.
6. (classifier concepts only) Select **Match XML character references** to match XML character references that appear in a document. For example, match `&amp;` for the ampersand character.
7. (Only enabled when you build a project using a UTF-8 language) Select **Export CONCEPTS File with UTF-8 Display Names** to create an additional concepts binary file where only UTF-8 display names appear. In other words, in addition to the `language.concepts` file, the `language.utf8.concepts` is also created.

The `language.concepts` file contains the Latin-1 internal names, while the `language.utf8.concepts` file enables you to see the taxonomy in the UTF-8 language that appears in the **Taxonomy** tab. For example, if you created a taxonomy structure of concepts using Japanese, you might see:

Top/学校

instead of Top/School

8. Select **Allow Duplicate ID's** if you want to set identical identification numbers in the Data window for two or more of your concepts.
9. Use the settings under **Overlapping Concept Matches** to determine the behavior of SAS Content Categorization Studio when an input document contains terms that match more than one concept. For more

---

information and an example, see Section 2.9.4.A *The Concept Tab* on page 57.

---

**Note:** If you specify **All Matches**, you might consider using the priority setting in each Data window to rank multiple input documents.

---

10. (default setting: classifier concepts, only) **Case Sensitive** under **Default Classifier Matching**. Select **Case Insensitive Matching** to change the default setting for all of the classifier concepts. When you make this change, the application locates all matching terms, regardless of case.
11. (default setting) **Compile Speed** under **Optimize for**. Select **Matching Speed** to make concept matching the priority.

---

**Notes:** Unless you are developing large binary files, there is little performance difference between these settings.

---

---

## 17.3 Understanding Concept Types

SAS Content Categorization Studio provides two types of concept definitions. Classifier and grammar definitions enable you to specify the strings or patterns that identify a match in an input document.

Classifier concepts specify the strings to be matched and also enable you to use regular expressions to locate patterns. In addition, you can write a Boolean disambiguation rule if you want to differentiate between the same form of a word that is used in two different contexts. For example, the word bush has two different meanings in the following contexts. President George Bush and a bush in the garden. For more information, see Chapter 18: *Writing Classifiers*.

Grammar concepts, on the other hand, rely on part-of-speech identification and symbols. This feature enables SAS Content Categorization Studio to extract precise matches. For example, locate all catering companies within a specified state. For information about grammar rules, see Chapter 19.

Whether you are writing or editing a definition, it is important to understand that you cannot mix classifiers, regular expressions, and grammar rules within the same definition. You can mix concepts that are defined by classifiers with those specified by grammar rules in the same taxonomy. You can also reference classifier definitions using grammar rules, but you cannot reference grammar rules using classifier concepts.

Concept types cannot be changed like categories.

To change a concept type, complete these steps:

1. Click the new definition type in the Data window. Choose either **Classifier** or **Grammar**.
2. Re-type the old definition into the Definition window for the new concept.
3. Click **Syntax Check** to check the definition.
4. Select **Build --> Compile Concepts**.

---

## 17.4 Use the Sample Concepts Project

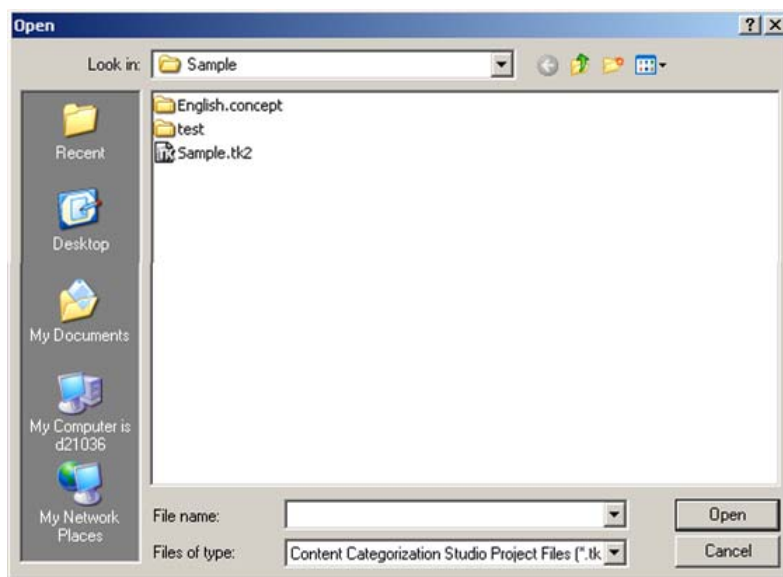
SAS Content Categorization Studio is installed with a sample project named `Sample.tk2`. This project enables you to familiarize yourself with the various elements of an existing project that include a prebuilt taxonomy, concepts, definitions, and testing documents.

The project files are saved in a Windows folder, named after the project. For example, the files for the project named `sample` could be stored in the following folder:

```
C:\Program Files\Teragram\tk240\Projects\Sample
```

To access the sample project that is shipped with the application, complete these steps:

1. Select **File --> Open Project**.
2. Double-click the `Sample.tk2` project in the Open window that appears.



- 
3. The Sample taxonomy appears in the **Taxonomy** tab.





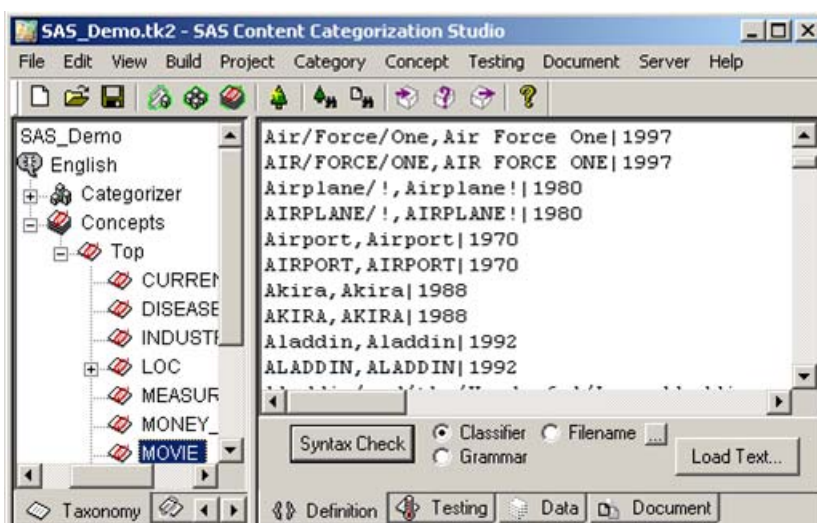
---

## 17.5 Write a Definition

Write a definition in the Definition window. You can also use this tab to enter a reference to a text file that contains the classifier definition that you want to use. The reference operation is the optimal solution when you write a long classifier definition. For example, if you develop a classifier definition with a million lines, reference this file (often a .txt file) to save compilation time within the application.


To write a definition, complete these steps:

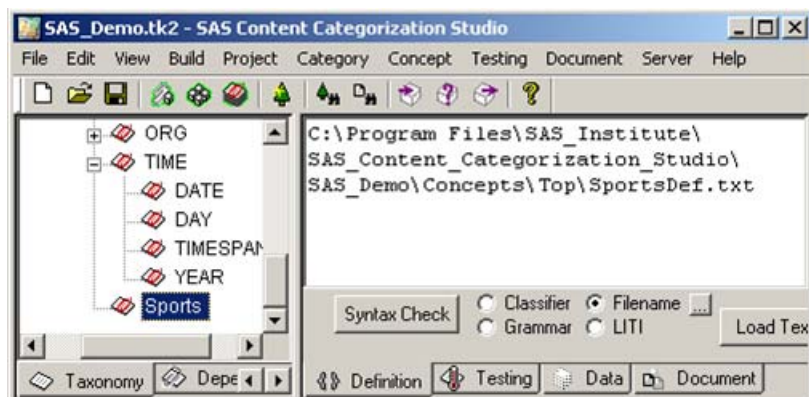
1. Select a concept in the Taxonomy window and click the **Definition** tab.



2. Select **Classifier** to type a classifier definition, or **Grammar** to enter a grammar rule.
3. Select **Filename** to reference a classifier concept definition file that resides in an external file.

---

Click  to load the path to the file instead of the definition that this file contains. The path to the selected file appears in the Definition window.



4. Select **Build --> Compile Concepts**.

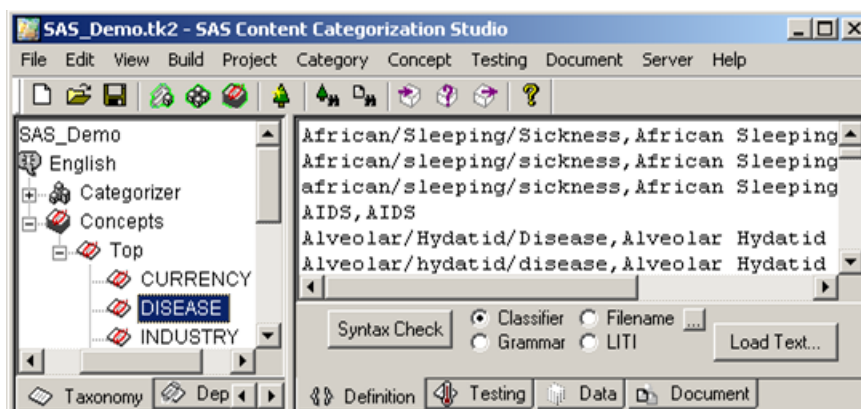
---

## 17.6 Use the Syntax Check Button

After you write a classifier definition for a concept, check its grammar. Perform this operation for every classifier concept before you compile the project.

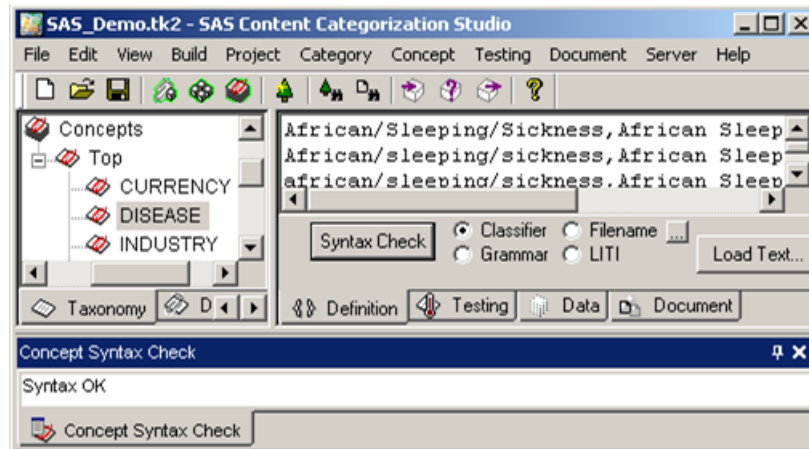
To check the grammar of a definition, complete these steps:

1. Select a classifier concept node in the Taxonomy window. For example, click `DISEASE`.

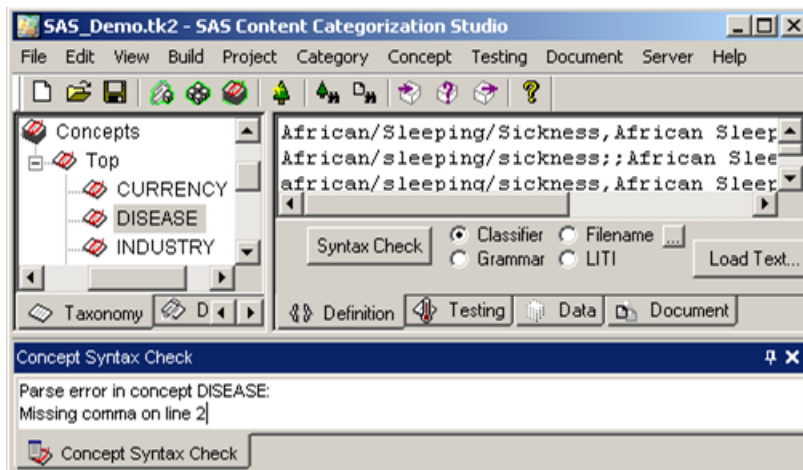


2. Click **Syntax Check**.

The Concept Syntax Check window appears and displays a status message. The status is either *Syntax OK* or an error message.



If the syntax is not OK, the Concept Syntax Check status window displays a message to help you edit this definition.



3. Click **X** to close the Concept Syntax Check window.

---

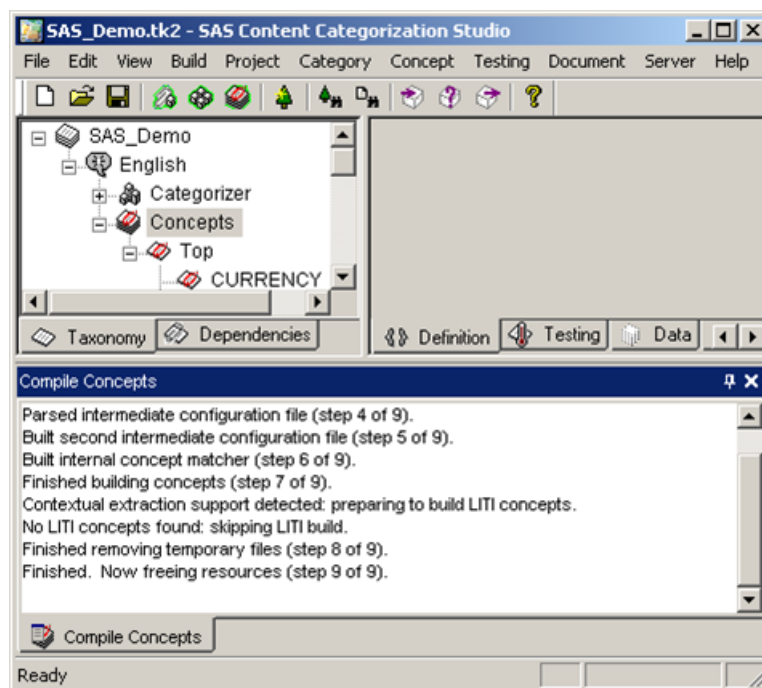
## 17.7 Compile Concepts

After you create and test the concepts, you can compile all of the concept definitions.

To compile your concepts, complete these steps:

1. Select the `Concepts` icon in the Taxonomy window.
2. Select **Build --> Compile Concepts**.

The Compile Concepts window appears at the bottom of the user interface. The progress and the results of the compilation are shown in this window.



3. Click **X** to close this window.



---

# Chapter: 18

## Writing Classifiers

---

- *Overview of Writing Classifiers*
- *Writing a Classifier Definition*
- *Writing Regular Expression Definitions*
- *Using Disambiguation to Increase Matching Precision*
- *Write a Definition in a Text File*
- *Generating Suggested Concepts*

### 18.1 Overview of Writing Classifiers

This chapter explains how to use classifiers to define simple concepts. A classifier can be a word, or a string. When you define a classifier concept, you enter a list of one or more terms. When there is a match in an input document on one or more of these terms, the text is a match for the concept.

Classifiers can define only autonomous pieces of information. These are snippets of information. If you want to identify relational data, use grammar-based concepts after you are familiar with the information presented in this chapter.

Classifier concepts can be either the parent or the child of a grammar-based concept in a hierarchical taxonomy. However, you cannot mix classifier and grammar rules within the same concept definition.

The definition for a classifier concept uses any of the following components:

Literal strings

A classifier concept definition can be comprised of one or more literal strings. For more information, see Section 18.2 *Writing a Classifier Definition* below.

Regular expressions

---

Specify a classifier concept definition using regular expressions. For more information, see Section 18.3 *Writing Regular Expression Definitions* on page 452.

#### Disambiguation

Disambiguate between documents that contain the same term in different contexts. For example, a document about flowering bushes should not be a match for the President Bush concept. For more information, see Section 18.4 *Using Disambiguation to Increase Matching Precision* on page 453.

You can also automatically generate a list of suggested concepts based on an input set of documents. Export and edit this list to create a classifier concept. Use the compile and syntax check operations to ensure that the rules are correct before you test them.

## 18.2 Writing a Classifier Definition

### 18.2.1 Format of Classifiers

A classifier concept can be defined by one or more words or by a literal string. However, you can also choose to specify a term to return when a match is located. In this case, the classifier definition uses the following format where `returned_information` is optional:

```
match_key, returned_information
```

The string that is matched in the input file is defined by the `match_key` part of the classifier concept. `returned_information` can be specified to modify the matched string. In other words, choose to return a string that is different from the matched string. (Returned information is used only when the `.concepts` file is applied by SAS Content Categorization Server to input documents.)



---

Display 18-1 Classifier Format Example



In the example above, every instance of:

- *weather* in an input document is returned as *weather*
- *FOX* in an input document is returned as *Fox news*
- *ABC* in an input document is returned as *ABC news*
- *CNN* in an input document is returned as *CNN news*

Matches can also be affected by case sensitivity. You can choose to make matches case sensitive, or case insensitive. For example, if **Case Sensitive** is selected in either the Project Settings - Concept window or the **Data** tab, matches on *FOX* are returned as a match on *ClassifierExample*. Any instances of *Fox*, *fox*, *FOx*, and so on, are not returned as matches.

---

**Note:** The settings in the Data window apply to the specific concept. For this reason, the case sensitivity setting in the Data window overrides the specification in the Project Settings - Concept window.

---

---

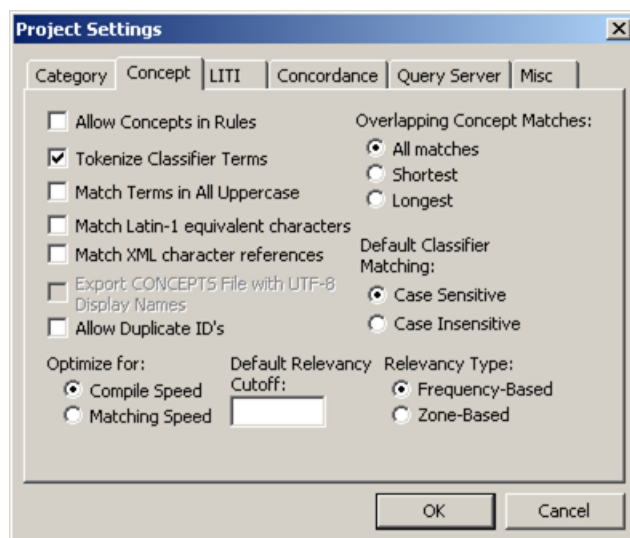
## 18.2.2 Before You Write Classifier Definitions

### 18.2.2.A Specifying Project Settings

There are several operations in the Project Settings - Concept window that you can use with classifier concepts. Specify these settings before you write your classifier definitions.

To access and use the Project Settings window, complete these steps:

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



The default settings are shown above.

2. Select **Allow Concepts in Rules** if you want to create dependencies with category rules for your classifier concepts.
3. (Default) Leave **Tokenize Classifier Terms** selected for new projects. The words that form a string in your classifier definition list are typically separated by spaces. This operation checks for spaces.

---

**Note:** The **Tokenize Classifier Terms** operation should always be selected, unless you choose to use a backslash (\) instead of spaces as the separator character.

---

4. (Optional) Select **Match Terms in All Uppercase** if you want to automatically add uppercase versions of the words in your definition to this definition.
5. (For Latin-1 languages that contain accented characters in their texts) Select **Match Latin-1 equivalent characters** to match the Latin-1 equivalent characters as if these characters are not accented.
6. Select **Match XML character references** to match XML character references that appear in a document. For example, match `&amp;` for the ampersand character.

### 18.2.2.B Case Sensitivity

By default, concepts are case sensitive. For this reason, the terms that you specify are matched in an input document if the matching letters appear in the same case specified in the `match_key`. For example, *Emergency* is matched only if the term *Emergency* is located in an input document. A match is not returned for any instances of *emergency*.

The case-sensitive setting for classifier concepts in the project is set in the Project Settings - Concept and Data windows. By default, **Case Sensitive** is

specified in the Project Settings - Concept window and **Use Project Default** in each Data window.



There are several combinations of these settings that you can use to change how case is used to return matches:

- Leave the default settings for the Project Settings - Concept window (**Case Sensitive**) and the Data windows (**Use Project Default**) selected. Matches are limited to the various combinations of upper- and lowercase letters that you specify. For example, write *Acura*, *acura*, and *ACURA*, to match any instances of exact matches on these terms.
- Select **Case Insensitive** in the Project Settings - Concept window and **Use Project Default** in the Data window for the selected concept. In this example, every match on a term regardless of its case, is returned. continuing with the example above, all instances of *Acura*, regardless of case, are returned.

- 
- Select **Case Sensitive** or **Case Insensitive** in the Project Settings - Concept window. Choose the opposite setting in the Data window of the selected concept to overwrite this match for this concept only.
- 

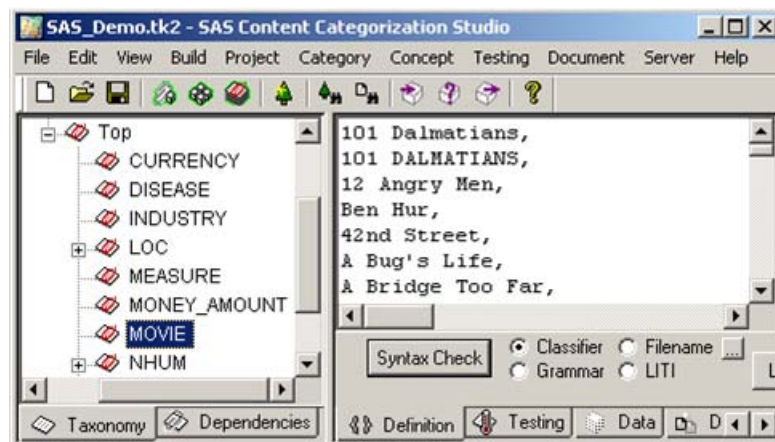
**Note:** The setting that you specify in the Data window overrides the specification in the Project Settings - Concept window.

---

### 18.2.3 Writing the match\_key

Begin writing concept definitions by specifying a list of the `match_key` entries, followed by a comma (,) in a list format. Enter each term on a separate line.

*Display 18-2 Match Key Terms*



When you write the match key part of the definition, do not use a single quotation mark (') as a standalone symbol. No matches are returned for this symbol.

---

## 18.2.4 Writing the Information String

The information string in a classifier definition is optional. Specify an information string when you want to return a different string than the matched terms. You can specify this string in mixed, upper-, or lowercase. For example, a match on the words *Abraham Lincoln* in an input document could return the text U.S. President Lincoln.

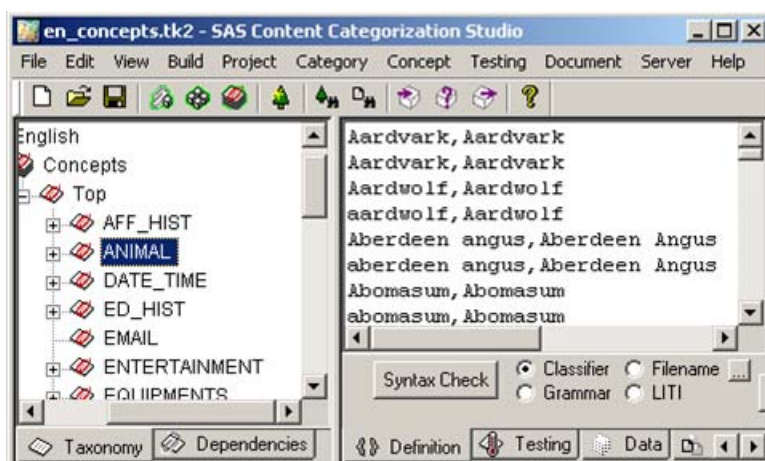
```
Abraham Lincoln, U.S. President Lincoln
```

The information string also enables you to specify the same returned string for different matches. For example, specify a definition that returns U.S. President Lincoln for several matched strings:

```
Mr. Lincoln, U.S. President Lincoln  
Mr. Abraham Lincoln, U.S. President Lincoln  
honest abe, U.S. President Lincoln  
President Abraham Lincoln, U.S. President Lincoln  
President Lincoln, U.S. President Lincoln
```

When the information string is present, it is delimited from the match string by a comma (,). You can use only one comma per line.

*Display 18-3 Information String Examples*



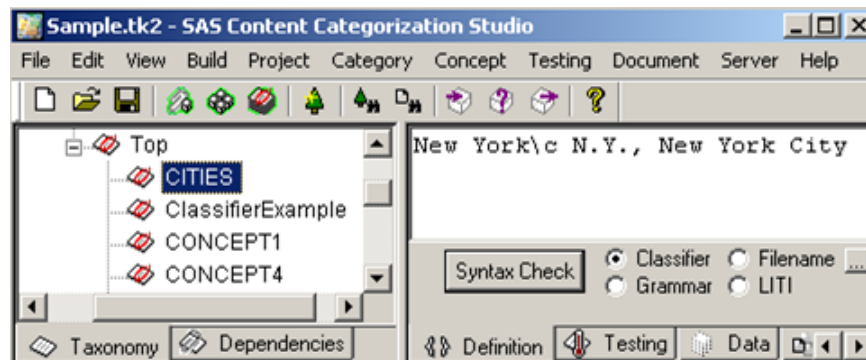
---

## 18.2.5 Matching the Comma Character

The comma character (,) is reserved for use as a separator character between the `match_key` entry and `returned_information` entry. The comma follows the `match_key` entry regardless of whether `returned_information` is specified. You can choose to match a comma using allowed characters for either the `match_key` or the `returned_information` string.

However, you can choose to match a comma (,) in an input document. To match a comma within the `match_key`, enter the backslash and lowercase c (\c) characters instead of a comma.

*Display 18-4 Matching the Comma Character Example*



This classifier concept definition example matches *New York, N.Y.* in an input document and returns *New York City* when applied by SAS Content Categorization Server.

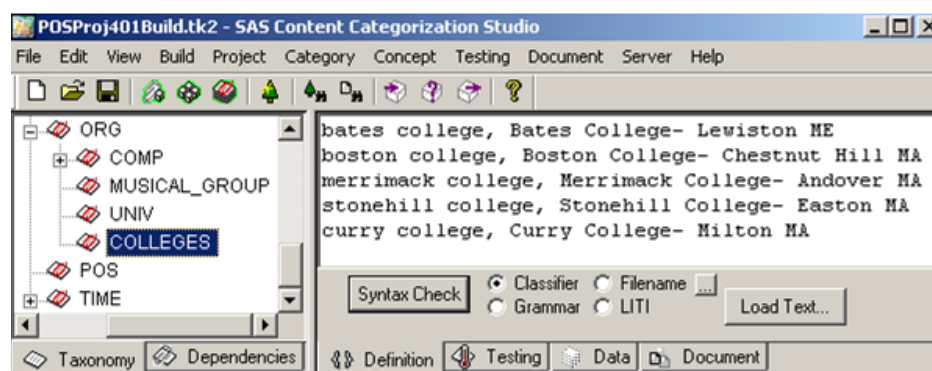
Display 18-5 A Returned Match for the Comma Character



Similar to the `match_key` field requirements, you substitute a different character when you want to specify a comma character in the `returned_information` string. You can use a hyphen (-) or a pipe (|) to serve as the separator character that is usually reserved for a comma.

For example, you could return the name and location of a college every time a match is located on the name of the college. To perform this operation, write a definition that is similar to the example shown below.

Display 18-6 A Hyphen Example



In SAS Content Categorization Studio the hyphen and pipe symbols have no special meaning. However, you can use a text parser to interpret these symbols according to your organization's requirements.



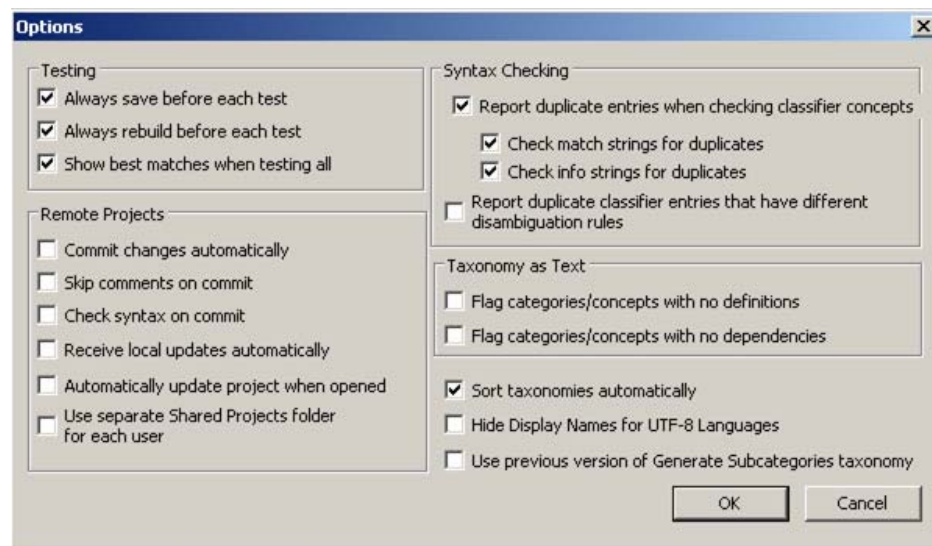
---

## 18.2.6 Locating Duplicates in the Match or Information Strings

When you write classifier definitions, you can set the **Syntax Checking** operations in the Options window by selecting **Report duplicate entries when checking classifier concepts**. This time-saving feature automatically checks for duplicate entries in your concept definitions.

If you choose this operation, select **Check match strings for duplicates**, **Check info strings for duplicates**, or both check boxes.

*Display 18-7 Choose Operations in the Options Window*



---

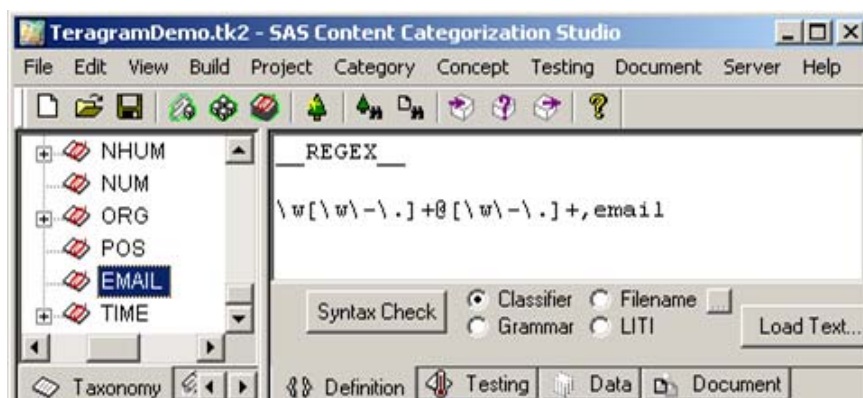
## 18.3 Writing Regular Expression Definitions

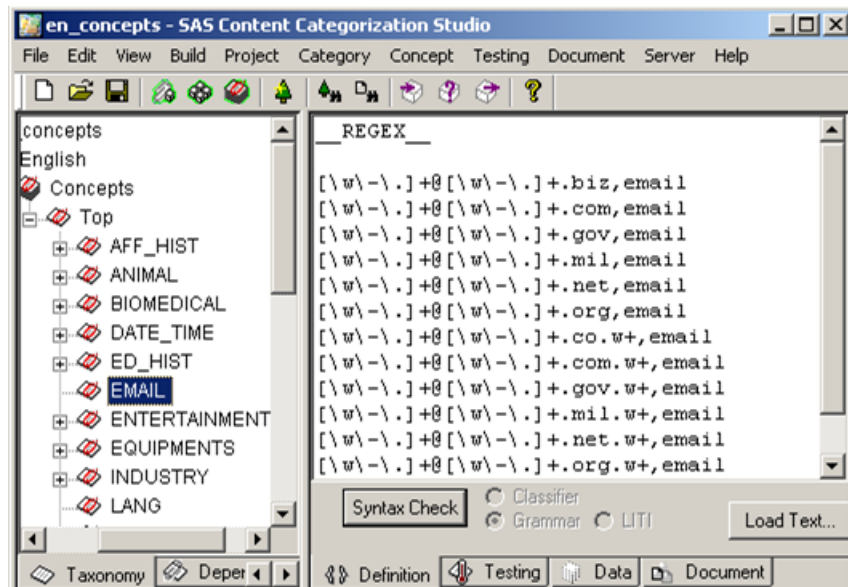
Use regular expressions when you want to locate matches on text that has recognizable patterns such as e-mail, phone numbers, and street addresses. In this case, you can define a classifier concept using regular expressions. Regular expressions enable you to specify the known formats and their variations and to return all of these matches without specifying individual strings.

The first line of a regular expression definition is `__REGEX__`. When you type this line, make sure that two underscore characters (`_`) precede, and two underscore characters follow, the word `REGEX`. Type only the term `REGEX` in uppercase letters.

This section is for advanced users who are experienced with writing regular expressions. For this reason, this section contains only two examples. For more information about how to write a regular expression, see Appendix A: *Regex Syntax and Part-of-Speech Tags*.

*Display 18-8 Short E-Mail Definition*





## 18.4 Using Disambiguation to Increase Matching Precision

### 18.4.1 Overview of Disambiguation

Precise concept matching relies on the ability to disambiguate between documents that contain the same term in different contexts. For this reason, specify a Boolean rule as part of the entity string of a classifier concept. Use `__TGIF` or `__TGUNLESS` to determine whether the concept needs to match the Boolean rule. In other words, you can use `__TGIF` to specify that the concepts only match if the document matches the Boolean rule. Or you can specify `__TGUNLESS` to determine that a concept is a match only in cases where the text does *not* match the rule.

---

**Note:** Specify two underscores (\_\_) before each TGIF or TGUNLESS term.

---

## 18.4.2 Before You Write Disambiguation Definitions

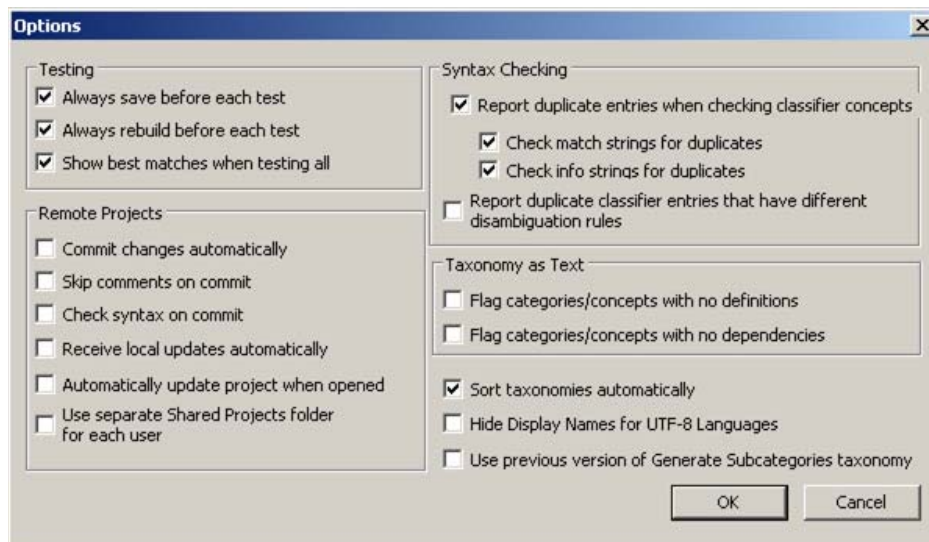
When you write rules using disambiguation, select **Report duplicate classifier entries that have different disambiguation rules** in the Options window.

---

**Note:** When you select **Report duplicate classifier entries that have different disambiguation rules**, also select **Check match strings for duplicates**, **Check info strings for duplicates**, or both.

---

*Display 18-10 Select Duplicate Operations*



---

### 18.4.3 Disambiguation Definition Examples

For example, if you want to extract matches for the *Giants football team*, you could specify:

*Display 18-11 Giants Disambiguation Example*



However, this definition is ambiguous because both the *San Francisco Giants* baseball team and the *New York Giants* football team are referred to as the *Giants*. In this case, you might want to disambiguate between sports documents where the topic is *football* and texts covering *baseball*. You can specify this differentiation by using either TGIF or TGUNLESS.

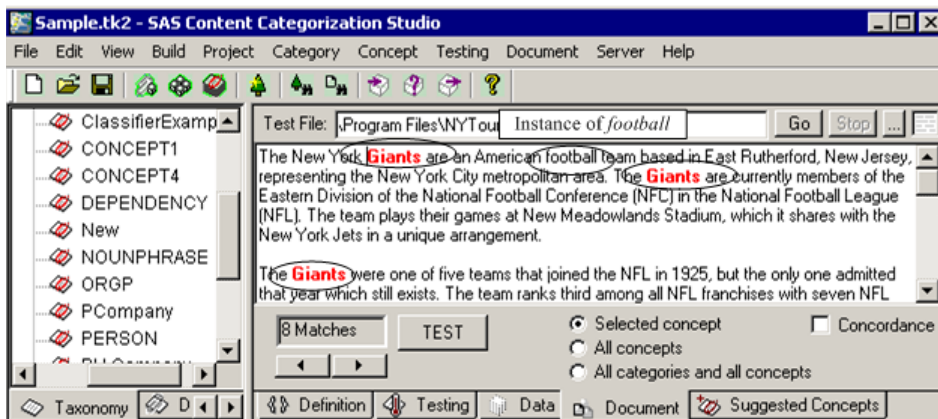
In this case, you can write the following line as the definition for the Giants football team:

*Display 18-12 Giants Football Team Disambiguation Example*



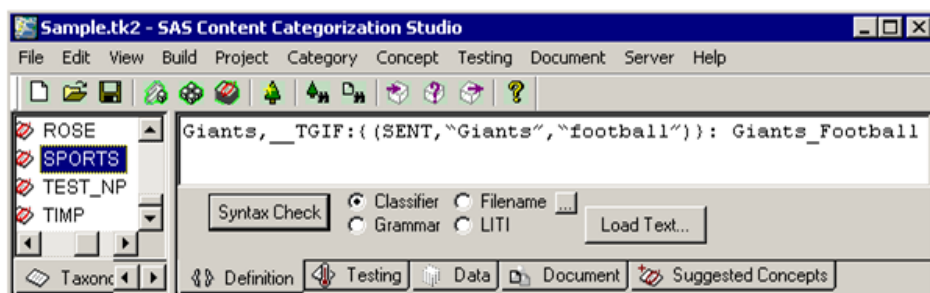
In this case, if the word *football* is located in an input text, all occurrences of the specified term *Giants* in this document are matched. The specified entity string `Giants_Football` is assigned to each occurrence of the matched term.

Figure 18-1 Matches on The Giants Disambiguation Definition



You could also choose to write a more restrictive Boolean rule. For example, you could write the following rule:

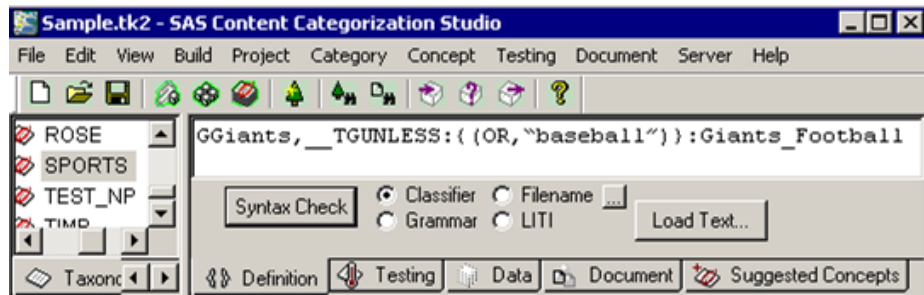
Display 18-2 A Second Giants Disambiguation Definition



In the example above, a match only occurs when the words *Giants* and *football* both occur in the same sentence. When both of these terms occur in the same sentence, every occurrence of the word *Giants* in the document is assigned the specified entity string, `Giants_Football`. For an example of this match, see Figure 18-1 above.

Also choose to use \_\_TGUNLESS. For example, when the word *Giants* in your input texts usually refers to *football* use \_\_TGUNLESS.

Display 18-3 Giants TGUNLESS Example



In the definition above, *Giants* is matched if the document does *not* match the Boolean rule meaning that it does not contain the word *baseball*.

Display 18-4 Giants Baseball Unmatched Example





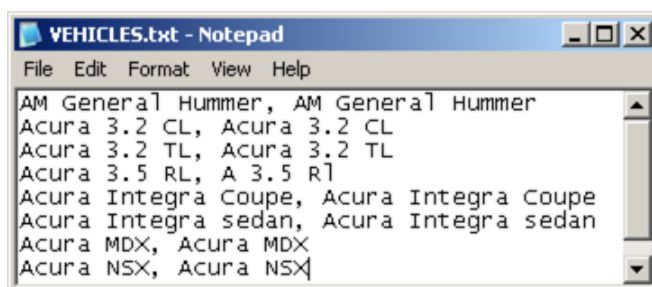
---

## 18.5 Write a Definition in a Text File

When you develop a long classifier definition, you can write all of the lines into a text document. For example, if you write a classifier definition that has a million lines, choose this operation and save build time in SAS Content Categorization Studio. After you write the definition, edit and save the .txt document. After you write the concept definition, you can import this definition using the Load Text operation that is available in the Definition window.

To write a classifier definition in a text file and to import this definition, complete these steps:

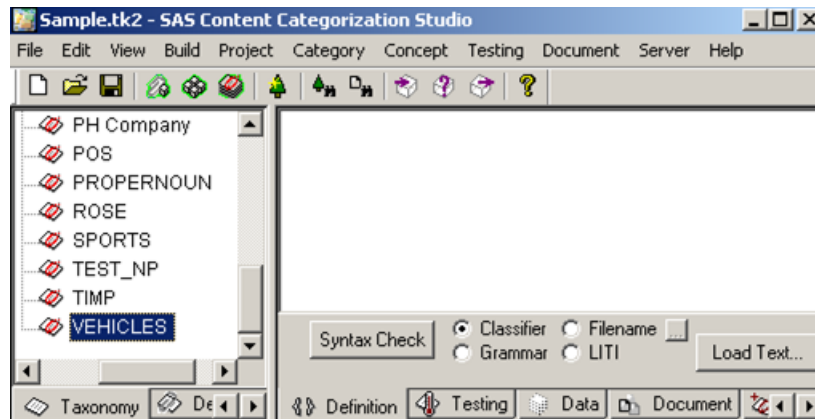
1. Access a text editing program. For example, access *Notepad*.



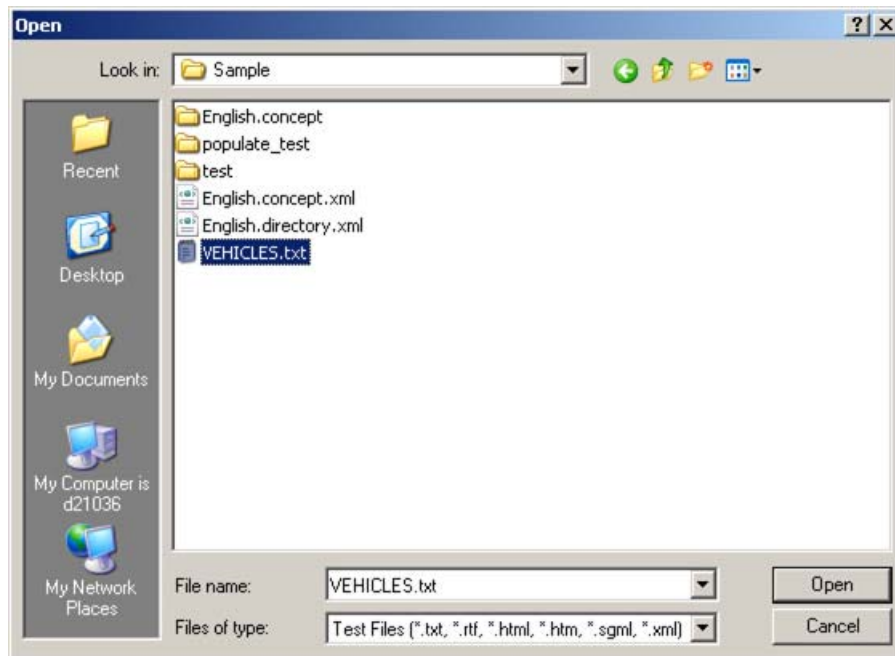
2. Type the concept definition into the new file.
3. Save as a .txt file.



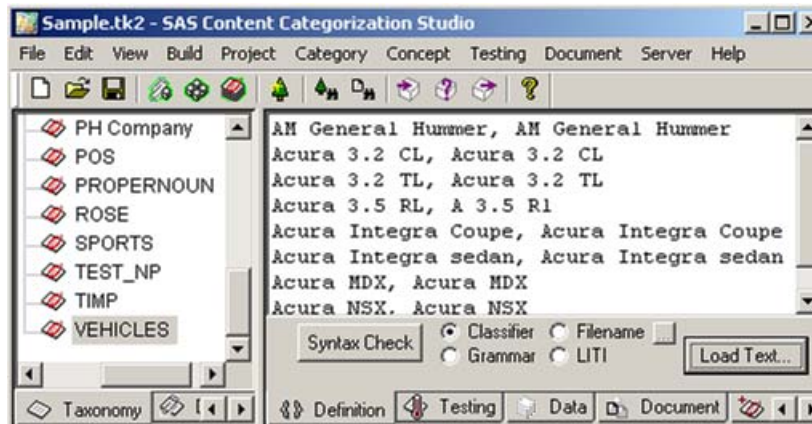
4. Select the concept node in the **Taxonomy** tab that this file defines and click the **Definition** tab.



5. Click **Load Text**.
6. Select the saved `.txt` file in the Open window that appears.



7. Click **Open**.
8. The classifier definition is loaded into the Definition window.



9. Click **Syntax Check** to check the definition.
10. If the syntax is OK, select **Build --> Compile Concepts**.  
If the syntax is not OK, edit the definition and repeat Step 9.

---

## 18.6 Generating Suggested Concepts

### 18.6.1 Overview of Generating Suggested Concepts

Use SAS Content Categorization Studio to suggest terms that might be appropriate for a concept that is defined using classifier terms only. This suggested concepts operation is performed by enabling SAS Content Categorization Studio to import terms from a matching concept in another project into your current project.

For explanatory purposes, the *new project* is the project that contains a classifier concept that uses terms from another classifier concept. The term *original project* is used to specify the project that contains the concept with the terms that are being exported into the *new project*.

The Generate Suggested Concepts operation can be performed only under the following specific conditions:

- Information strings are the only strings that are imported.

---

**Note:** Add a comma (,) to the right of each term. At this time the comma is not imported. Without a comma your concepts do not compile. For this reason, you add commas.

---

- Information strings can be imported only from a concept that has the same name as the concept that is adding these terms.
- The only terms that are imported are those terms that match the documents in the testing folder for the original project. If you want to generate all of the terms from the definition in the original project, make sure you have test documents that include all of these terms.
- Use the `.tk2` file that is automatically generated with each project that you develop.

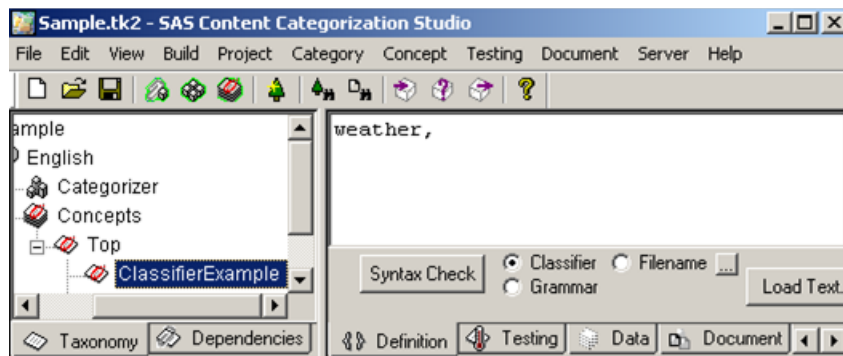
### 18.6.2 Generate Suggested Concepts

When you develop a project and want to import the classifier terms in another project, use the generate suggested concepts operation.

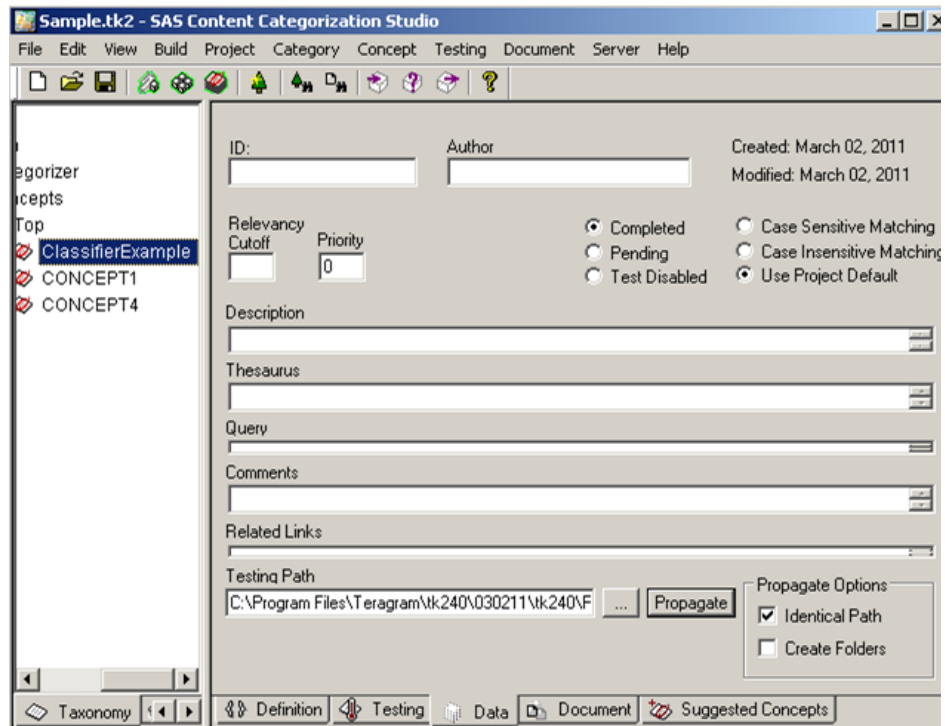
---

To Generate Suggested Concepts, complete these steps:

1. If you have not already done so, access a project and check the definition for the classifier concept that you want to expand. For example, access the `Sample.tk2` project and check the `ClassifierExample` definition.



- 
2. Make sure that the testing path in the **Data** tab is set to the same file of test documents that is specified in the original project.



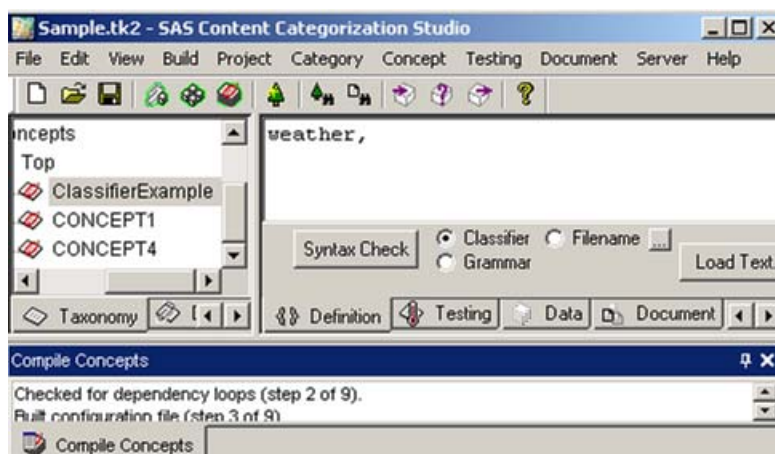
---

**Hint:** You do not need to test these documents.

---

3. Select **Build --> Compile Concepts**.

The Compile Concepts window appears at the bottom of the user interface.

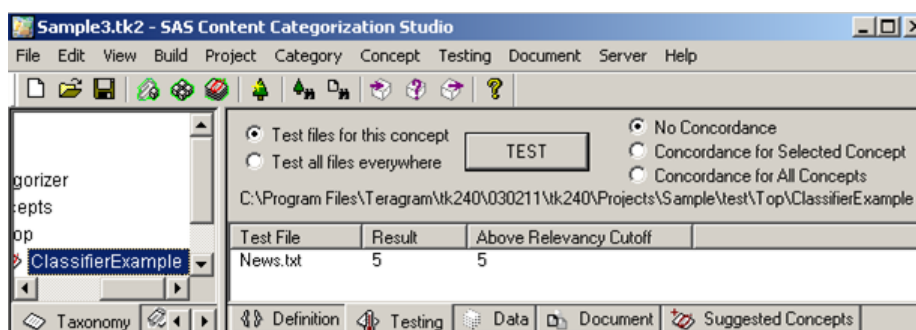


4. (Optional) Click **X** in the Compile Concepts window.
5. Access the original project that contains a concept node with the same name as the concept in the original project. For example, access Sample3.

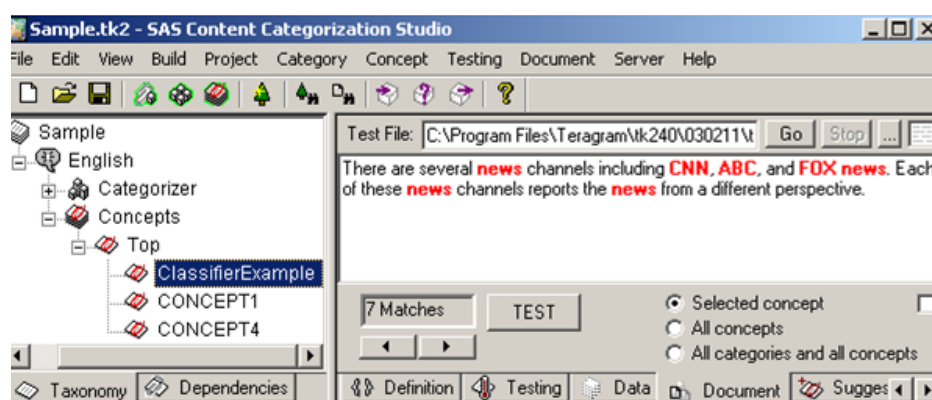


6. Check the matching concept definition. For example, check the matching terms for the ClassifierExample concept.
7. Complete Step 3 on 463 through Step 4 on 464.

- 
8. (Optional) Click the **Testing** tab and click **TEST** to see the number of matching instances for the terms in the definition.



9. (Optional) Double-click a testing document to access the text in the Document window. For example, double-click on News . txt.



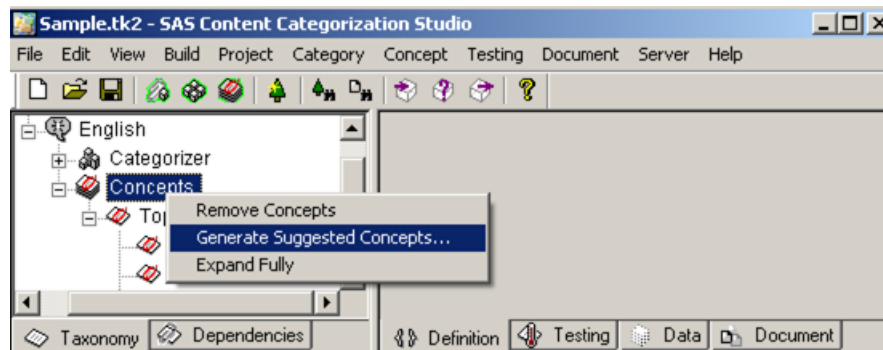
---

**Hint:** If you do not see the matches that you expect, select **Project Settings --> Concepts** and check the **Overlapping Project Settings** selection.

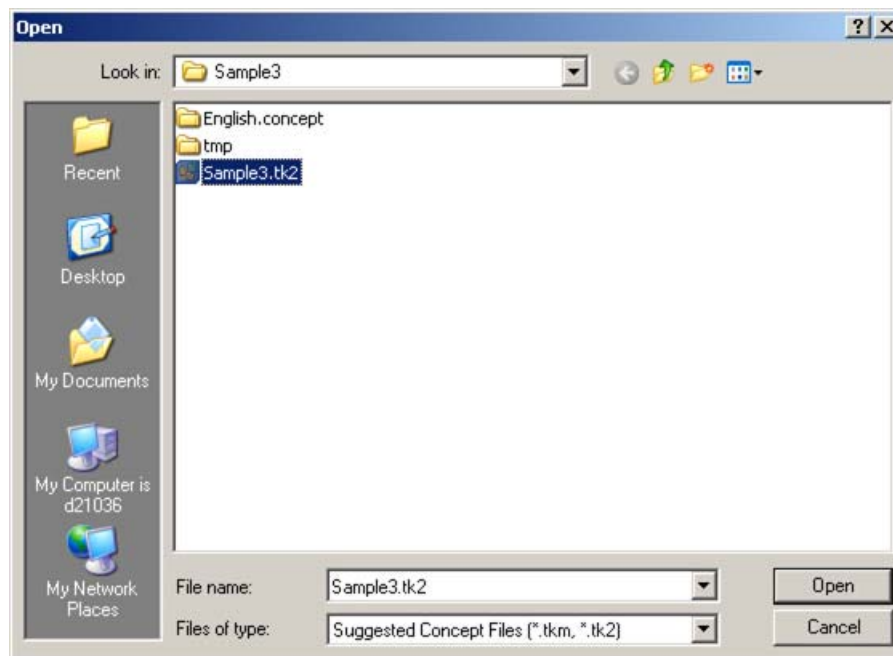
---

10. (Optional) Compare the definition terms with the matches that appear in the testing document. This operation enables you to see the definition terms that are available for the Generate Suggested Concepts operation.

11. Access the new project. For example, access `Sample.tk2`.
12. Right-click on `Concepts` in the Taxonomy window of the new project. Select **Generate Suggested Concepts** from the drop-down menu that appears.

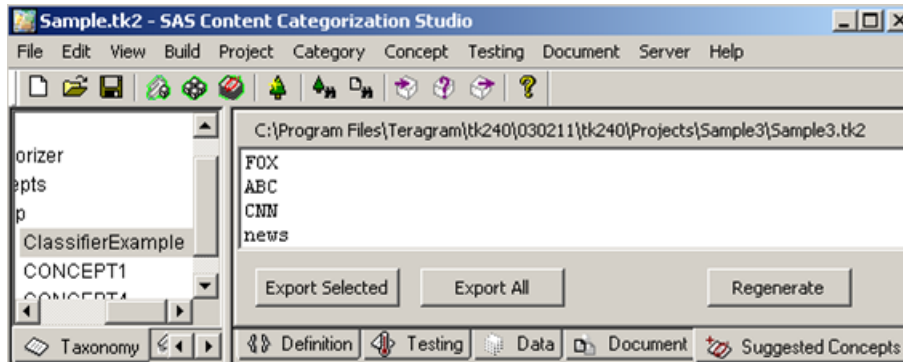


13. The Open window appears. Select a `.tk2` file. For example, select the `Sample.tk2` file.





14. Click **Open**.
15. Select the matched concept. For example, select `ClassifierExample`.
16. Click the **Suggested Concepts** tab to see the list of imported concept terms.



17. Double-click a concept node in the Taxonomy window and click the **Suggested Concepts** tab. The terms that appear are the terms that are not currently part of the definition, but are found in the testing files shared by both projects.
18. Select one of the following operations in the **Suggested Concepts** tab:

**Export Selected**

use this operation after you select one or more classifiers. These terms are exported into the **Definition** tab for the selected concept. The selected term appears at the bottom of the list in the Definition tab.

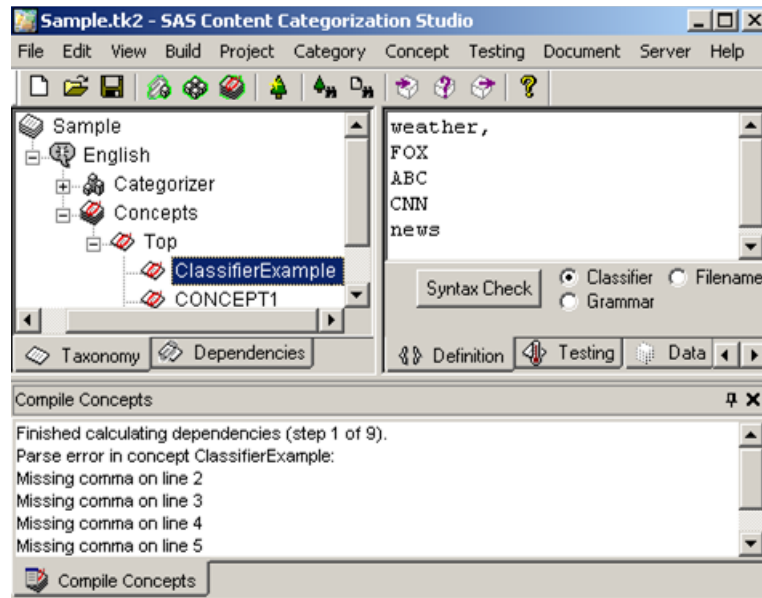
**Export All**

export all of the classifiers into the Definition window for the selected concept. The exported classifiers appear at the bottom of the list in the Definition window for the selected concept.

**Regenerate**

see a new list of suggested concepts.

- 
19. Click the **Definition** tab to see the exported terms. For example, see all of the terms if you selected **Export All**.



20. Test using the **Testing** and the **Document** tabs to make sure that the returned matches are the matches that you expect to see.

---

# Chapter: 19

## Writing Grammar Rules

---

- *Overview of Writing Grammar Rules*
- *Specifying Project Settings and Options*
- *Specifying Terminal Symbols or Strings*
- *Using Nonterminal Symbols*
- *Writing Grammar Rules*

### 19.1 Overview of Writing Grammar Rules

This chapter describes how to write rules that define grammar concepts. Grammar is defined as the set of rules and conventions that govern the way that words are used and sentences are constructed in any given language.

Deploying advanced linguistic technologies, grammar concepts enable you to identify entities and the existing relationships between these entities. This is true, in cases where you might not know about these relationships before you write your rules. For example, you could define the concept CITY with classifier rules and then use grammar rules to define a higher-level concept such as LOCATION that references the CITY concept.

Grammar rules are case-sensitive. For this reason, you can use special symbols and refer to other concepts to match terms such as *organizations*. Unlike classifier terms, you cannot specify case sensitivity in the Project Settings or Data windows for grammar concepts. Matches are returned in a case-sensitive manner for the terms that you specify.

When you write your grammar rules, use both terminal and nonterminal symbols. Nonterminal symbols include part of speech tags and symbols that are used to locate matches in input documents. For more information, see Section 19.3 *Specifying Terminal Symbols or Strings* on page 474 and Section 19.4 *Using Nonterminal Symbols* on page 476.

---

**Note:** A known issue to be addressed in a future release, is the fact that one grammar concept can mask another. In other words, one grammar definition can preclude another grammar definition from returning a match. You can comment out rules, or eliminate concepts, to locate the offending definition if you suspect this issue.

---

### 19.1.1 The Features and Benefits of Grammar Concepts

Grammar concepts, unlike classifier concepts, enable SAS Content Categorization Studio to locate concepts that are not specifically listed by name. For example, use grammar concepts to locate street names, animal species, or drugs made by a specific manufacturer. Grammar concepts identify this information by identifying the terms in their grammatical context.

You can also use grammar concepts to identify relationships between pieces of known or unknown information. For example, locate all of the officers in a specific company. You can locate this data when you specify the grammar that identifies the entities and the relationship between these terms.

When you choose to write grammar definitions, you gain the following benefits:

- Return information based on grammatical relationships.
- Use part-of-speech tags to identify matches in the context of the grammar used to construct the original document. For example, `N` is used to identify nouns in the English language. When you specify this part-of-speech tag, SAS Content Categorization Studio identifies all of the nouns in an input document. For more information, see Section A.2 *Part-of-Speech Tags* on page 508.
- Add other symbols, such as `#cap` and `#w` to specify the case of the string matches. For more information about these symbols and language sensitivity, see Section 19.4.4 *Using the #cap and #w Symbols* on page 480.
- Simplify grammar rule development when you define intermediate concepts within the definition of a broader concept. For more

---

information, see Section 19.5.3 *Writing Intermediate Concepts* on page 485.

- Create dependencies between grammar concepts and classifier concepts, or between grammar concepts. For more information, see Section 19.5.4 *Defining Dependencies in Grammar Rules* on page 487.
- Insert comments into grammar definitions. Use these notes to track definition development. These notes are not compiled in the definition, but they are available for reference purposes. For more information, see Section 19.5.2 *Inserting Comment Lines into the Grammar* on page 484.

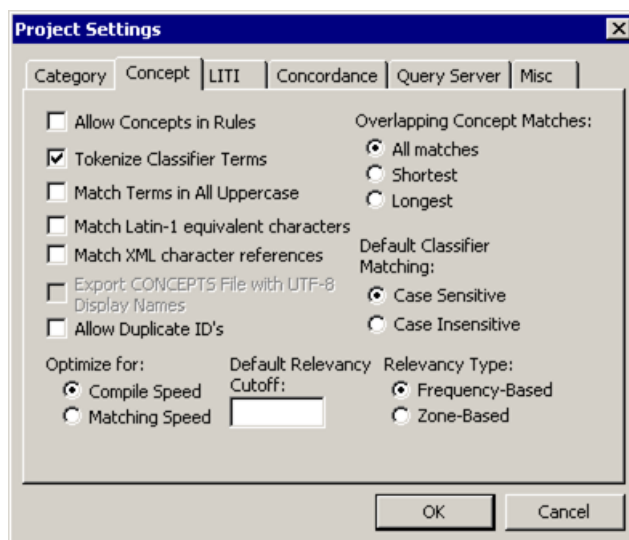
## 19.2 Specifying Project Settings and Options

Some of the Project Settings apply to grammar definitions, but none of the settings in the Options window are only for grammar concepts. For this reason, use this section to set your project settings and see Section 2.8 *The Options Window* on page 47 to specify your options.

Specify the Project Settings before you write your definitions. Project settings apply across the entire project. However, if you specify a setting in the Data window, this specification overrides the setting in the Project Settings window.

To access and use the Project Settings - Concept window, complete these steps:

1. Select **Project --> Project Settings**.



2. Click the **Concept** tab.
3. (Optional) Select a different selection under **Overlapping Concept Matches**. Specify one of the following operations:  
(Default) **All matches**  
return all matched terms.  
**Shortest**  
return the shortest match.  
**Longest**  
(default setting) return the longest match.
4. (Optional) Use **Export CONCEPTS File with UTF-8 Display Names** if you build a project using a UTF-8 language. This setting builds an additional concepts binary file where only UTF-8 display names appear. In other words, an additional file `language.concepts` is created. This file is `language.utf8.concepts`.

The `language.concepts` file contains the Latin-1 internal names, while the `language.utf8.concepts` file enables you to see the taxonomy in the UTF-8 language that appears in the **Taxonomy** tab.

---

For example, if you created a taxonomy structure of concepts using Japanese, you might see:

Top/学校

instead of Top/School

5. (Optional) **Allow Duplicate ID's** make it possible to assign duplicate identification numbers for two or more categories. You can enter these numbers in the **Data** tab for each affected concept.
6. Click **OK**.

## 19.2.1 Format of Grammar Rules

A grammar-based concept is defined by one or more grammar rules. Each rule is written into the Definition window using the following format:

\*CONCEPT\_SYMBOL = match symbols

The `CONCEPT_SYMBOL` can be one of the following types:

### Concept

The name of the concept, as it is listed in the **Taxonomy** tab can be specified as the `CONCEPT_SYMBOL`. This name can also be whatever you choose. For example, for a *Corporation* concept, you can specify `CEO`. If you are specifying the rule in an intermediate concept, the `CONCEPT_SYMBOL` specifies the name of this concept followed by a rule.

### Intermediate Concept

An intermediate concept is referenced only within the grammar specification of the referencing concept. This name does not exist outside of the body of this concept. For example, the *Title* intermediate concept might be defined within the *Corporation* definition.

An intermediate concept cannot be referenced directly by another concept in the taxonomy. If another concept references the grammar definition where the intermediate concept resides, the intermediate concept is indirectly referenced.

For example, the *Presidents* concept could reference the *Dog* concept that includes the `Breeds` intermediate concept. However, the *Presidents* concept cannot directly reference any specific lines in the `Breeds`

---

definition. For more information, see Section 19.5.4 *Defining Dependencies in Grammar Rules* on page 487.

The `match symbols`, or rule that returns a match on this concept, can use any of the following components:

#### Part-of speech tag

These classification tags match one or more words based on their grammatical function. Continuing with the example above, specify `N` in English to match a noun such as *dog*. For more information, see Section A.2 *Part-of-Speech Tags* on page 508.

#### :PN tag

This tag specifies that a proper noun is matched. This is true, whether the matched noun occurs in the dictionary that is shipped with this application.

Note: The `:PN` tag does not match

#### String

A specified term, or string of words, is also known as a non-terminal symbol. Matches can occur when the string is matched in a case-sensitive manner within the input document. For example, *red* is not matched in the following sentence. *Red lights mean stop!*

#### Non-terminal symbol

This symbol either stands alone or with another string that is used to match a category of terms. For example, `#cap` specifies that a match occurs only on a word that begins with an uppercase letter. For more information, see Section 19.4 *Using Nonterminal Symbols* on page 476.

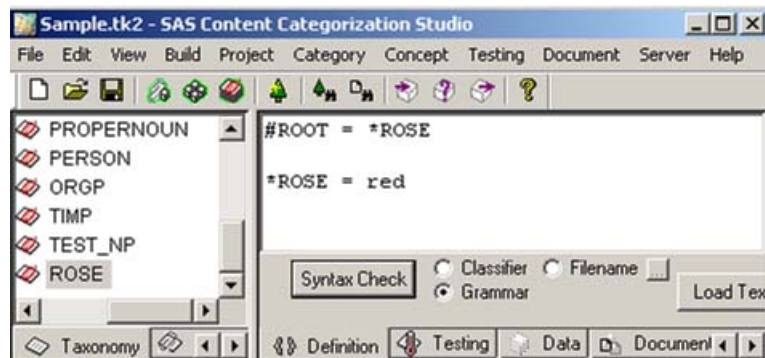
The `CONCEPT_SYMBOL` and `match symbols` are always separated by an equal sign (=). Together these specifications form a line in a concept definition. Each rule in the grammar specification is written on a separate line.

## 19.3 Specifying Terminal Symbols or Strings

You can specify strings, more formally known as terminal symbols, in your grammar definitions. These strings, or terms, are similar to the terms used for classifier concepts. Strings match only themselves in the input document.

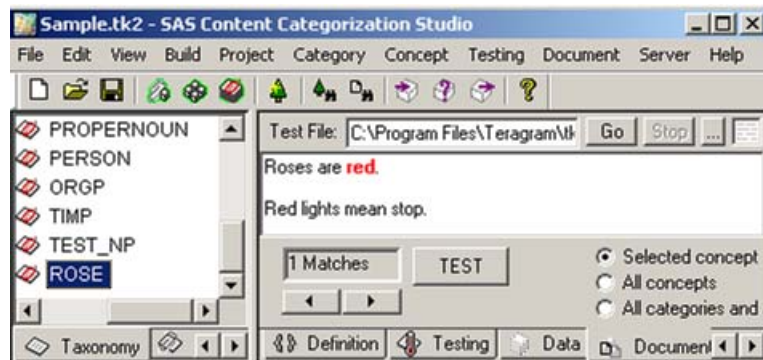


Display 19-1 String Definition Example



In this example the word *red* is specified as the string to be matched.

Display 19-2 String Match Example



This rule matches the term *red* in the sentence *Roses are red*. However, no match is returned for the occurrence of *red* in the sentence *Red lights mean stop*. The uppercase letter *R* in the term *Red* eliminates the possibility of a match on the word *Red* due to case sensitivity.

Specify non-terminal symbols such as characters and part-of-speech tags to modify the context where a match for these terms occurs.

---

**Note:** Although colons (:) precede part-of-speech tags, there are no colons before strings.

---

---

## 19.4 Using Nonterminal Symbols

### 19.4.1 Understanding Non-Terminal Symbols

Nonterminal, unlike terminal, symbols match an entity other than themselves. In other words, nonterminal symbols such as `:PN`, `N`, and `#cap` do not match `:PN`, `N`, and `#cap` literally. Instead, nonterminal symbols represent a type of match to be located in input documents.

### 19.4.2 Using Characters

There are several symbols that you can use in your grammar concepts, depending on their construction. Not all of these characters are nonterminal characters, but they are used in definitions with nonterminals.

Table 19-1: Characters Used in Grammar Rules

Symbol	Usage
:	Specify the colon before a part-of-speech tag. This symbol specifies that the following part- of-speech is defined in a SAS Content Categorization Studio syntactical file.
#	Specify a pound or hash sign, followed by the word <code>ROOT</code> , to begin each grammar concept definition. Also use the hash sign to indicate that a comment that follows this sign is not compiled.
*	Use the asterisk to reference an intermediate concept. An intermediate concept is a concept that is defined within the same root.
!	Specify the exclamation point to refer to a concept definition that is located outside of this file.

---

## 19.4.3 Specifying Part-of-Speech Tags

### 19.4.3.A Overview of Part-of-Speech Tags

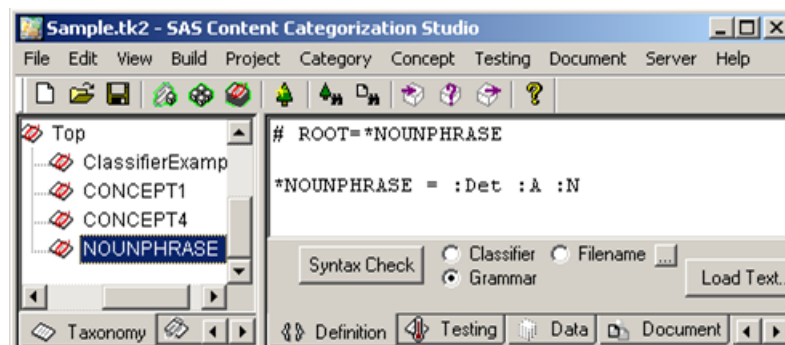
Parts of speech define the function of words in an input document. When you choose to specify parts of speech in a concept definition, you enable SAS Content Categorization Studio to locate matches that you might not know when you write your definitions. When you specify a part-of-speech tag, place a colon (:) before the tag.

### 19.4.3.B Part-of-Speech Tags in the English Dictionary

You can use part-of-speech codes to develop grammar rules. See the examples of the commonly used codes for English in Section A.2 *Part-of-Speech Tags* on page 508. Although the tags for each language differ, they are all case-sensitive. These tags make it possible for SAS Content Categorization Studio to identify previously unknown information according to the pattern that you specify.

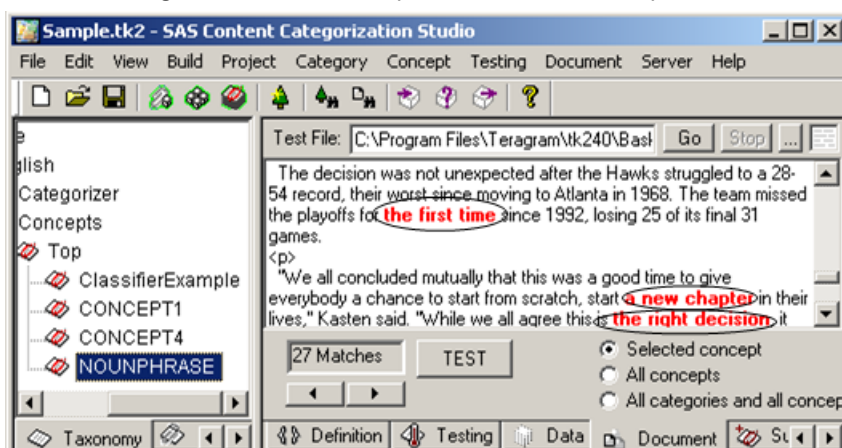
The following example specifies a grammar rule for a noun-phrase that can be matched when it is located in an input text:

*Display 19-3 Part-of-Speech Definition Example*



The determinant part-of-speech tag (`Det`) specifies a match on words such as *the* and *a*. The adjective part-of-speech tag returns matches on words such as *blue*, *third*, and *new*. The noun part-of-speech tag returns matches on words such as *boat*, *plant*, and *lion*.

Figure 19-1 Part-of-Speech Matches Example



### 19.4.3.C PN Tag

The :PN, or proper noun tag, locates words that begin with an uppercase letter, whether these nouns are included in the dictionary that is shipped with this application. The :PN tag shortens the rule writing process for you and enables you to identify all proper nouns in your input documents.

Like the #cap symbol, the PN tag matches each instance of a word that appears in uppercase. The only difference between th

If you choose to use the #cap symbol, specify #cap for each occurrence of a matched term. For example, if you choose to match a proper noun that has three words without specifying :PN, specify a string of three instances of #cap. For more information, see Section 19.4.4.B *Specifying the #cap Symbol* on page 480.

---

**Notes:** The :PN tag applies to proper nouns that are, or are not, found in the dictionary that is shipped with the application.  
A known issue, to be addressed in a future release, is the fact that the PN tag can return a match on a word that is not a proper noun. For example, a word that

---

begins a sentence might be matched as a proper noun because the word begins with an uppercase letter.

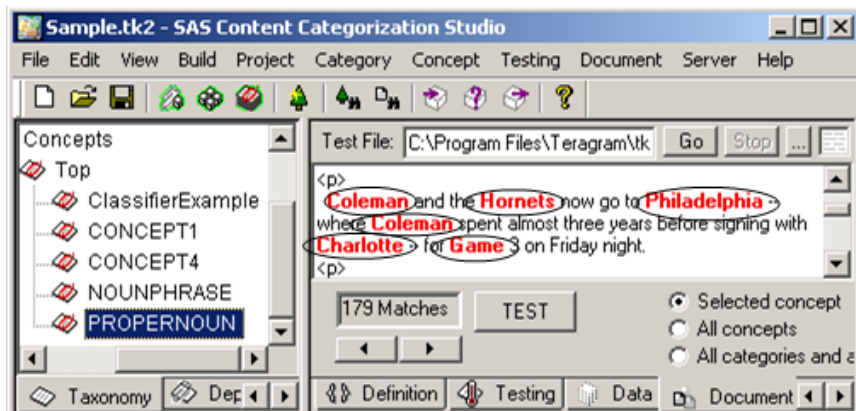
---

*Display 19-4 Proper Noun Definition Example*



This concept extracts any proper noun. For example, use the PN symbol to locate Bangladesh, India, or John F. Kennedy.

*Figure 19-2 Proper Noun Matches Example*



---

## 19.4.4 Using the #cap and #w Symbols

### 19.4.4.A Available Languages and Case Sensitivity

When referencing words in a grammar rule, you can use either the #cap or #w symbol. The #cap and #w symbols are available only for English Latin-1 and are case-sensitive. Unlike #cap, the #w symbol can be used to match nonwhitespace characters.

### 19.4.4.B Specifying the #cap Symbol

The #cap symbol matches any word that begins with an uppercase letter. For example, #cap matches Joseph Kennedy, Stop, and so on. However, the #cap symbol also matches *The* if *The* appears at the beginning of a sentence. For this reason, this symbol is typically used to match all instances of words that begin with an uppercase letter.

---

**Hint:** You can also use the :PN symbol to match a proper noun that is, or is not, located in the dictionary shipped with the application. For more information, see Section 19.4.3.C *PN Tag* on page 478.

---

For example, write a rule using the cap symbol when you have a file that lists names, scores, and events. In this case, you might want to match only the names and events and not the information about the respective scores.

*Display 19-5 Cap Symbol Definition Example*



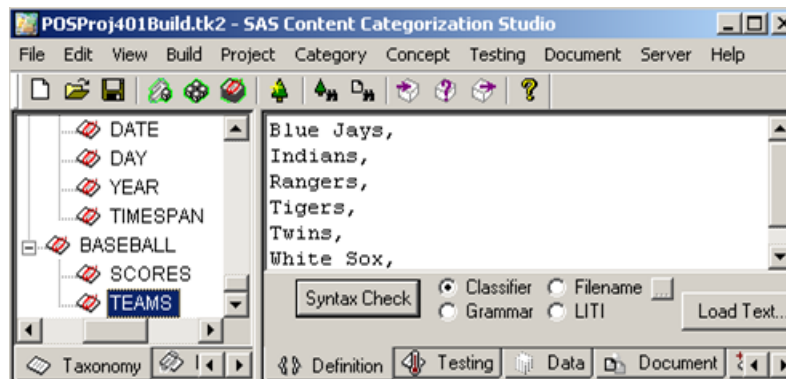
---

#### 19.4.4.C Specifying the #w Symbol

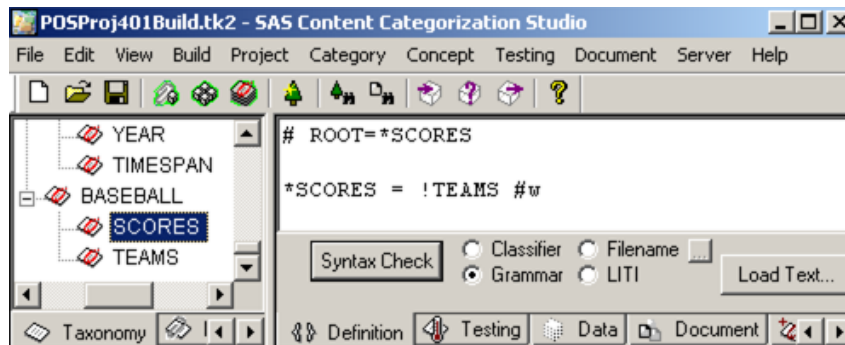
The #w symbol stands for one, or more, nonwhitespace characters and is case-insensitive. The #w symbol matches any single word or term. A term can consist of alphabetic or non-alphabetic characters. For example, <p>, <, Web, 1.0, and so on. This symbol is typically used to return a match on an unknown word that is located in a predefined location.

For example, you might want to return the results of recent baseball games. To create this project, complete these steps:

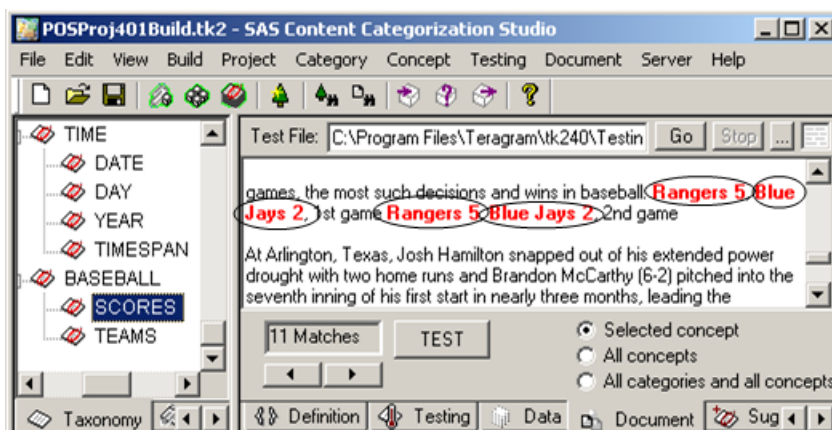
1. Develop a classifier concept that specifies the names of the baseball teams. For more information, see Section 18.2 *Writing a Classifier Definition* on page 442.



2. Develop a grammar concept rule that references the classifier concept and specifies the #w symbol. For more information, see Section 19.5.3 *Writing Intermediate Concepts* on page 485.



3. Test an input document and check the results. For more information, see Chapter 20: *Testing Concepts*.



**Note:** A text might also contain the name of a matching team followed by a nonwhitespace character, or a term. In this case, the character or term along with the name of the team, could be returned as a match.



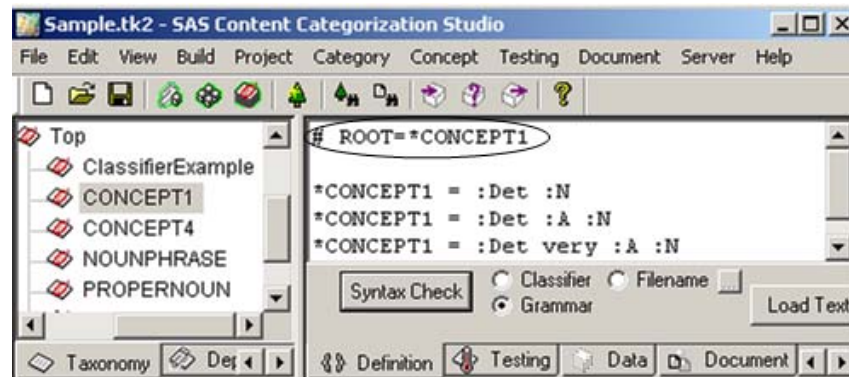
---

## 19.5 Writing Grammar Rules

### 19.5.1 Specifying the Root of the Grammar

A concept that is defined with grammar rules begins with the `ROOT` line followed by the name of the concept prefixed with an asterisk (\*).

*Figure 19-3 Grammar Root Example*



The `ROOT` line indicates that this is the base of the grammar for the concept `CONCEPT1`. Before you define your concept, enter `# ROOT=`. The defined concept is always preceded by an asterisk (\*). Any matches on one concept definition that is specified by the asterisk match this definition. In other words, every line does not require a match. A match on one concept definition returns a match for the concept.

---

**Note:** The pound sign (#) usually indicates a comment line, except when it is followed by the word `ROOT` in the first line of a grammar concept definition.

---

The specifications following the `ROOT` line define the grammar. Each concept definition line begins with the `*CONCEPT1` format.

---

## 19.5.2 Inserting Comment Lines into the Grammar

Use comment lines to enter notes as you develop your grammar concepts. These notes enable you, or another subject matter expert, to track definition development and to avoid errors. The hash tag (#) is specified at the beginning of a comment line to instruct the concepts compiler to ignore this line in the definition.

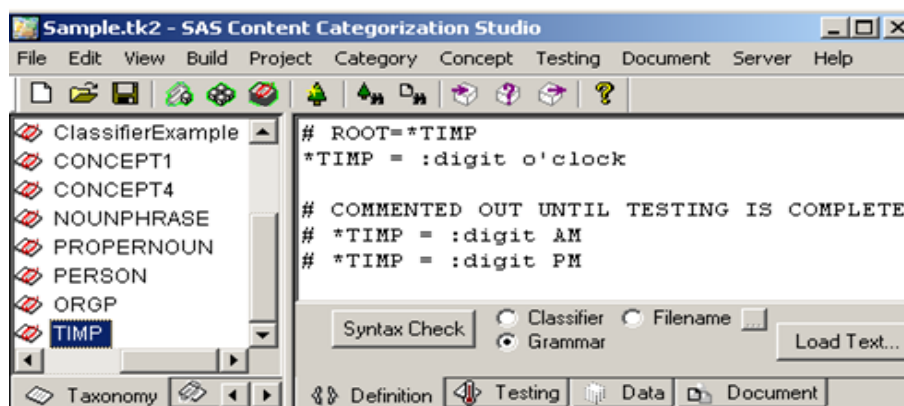
---

**Note:** The # ROOT line that begins a grammar definition is an exception. When the word ROOT is preceded by the pound sign, the line is read and processed by the concepts compiler.

---

The example below provides a sample of definition lines that are commented out of the grammar concept definition.

*Example 19-1: Viewing Comment Lines*



The first two lines in this example are processed by the compiler, but not the last three lines.

---

**Hint:** This example uses the intermediate concept. For more information, see Section 19.5.3 *Writing Intermediate Concepts* on page 485.

---

---

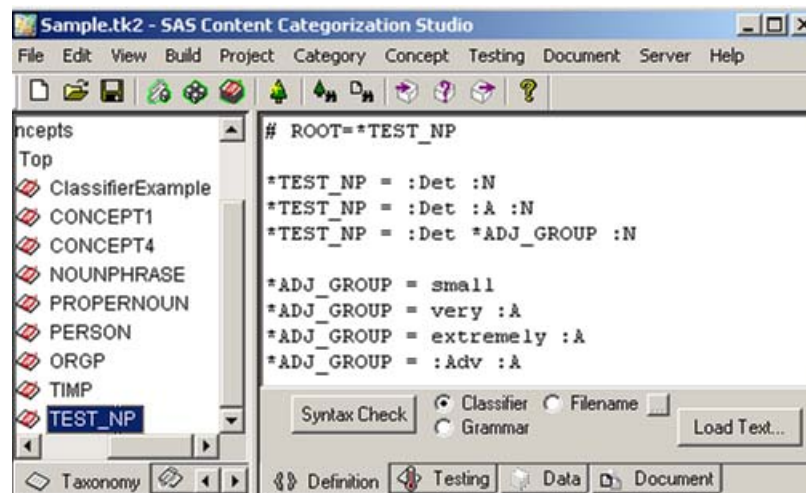
### 19.5.3 Writing Intermediate Concepts

An intermediate concept is a concept that is defined in a grammar definition. Write an intermediate concept into the definition of a concept where it can also be referenced by other intermediate concepts that are written into the same definition. (Although you can reference a concept, intermediate concepts cannot be directly referenced by other concepts in the taxonomy.)

Place an asterisk (\*) before a reference to an intermediate concept. In the example below, \*ADJ\_GROUP is an intermediate concept. ADJ\_GROUP is referenced by the asterisk.

- Although colons (:) precede part-of-speech tags, there are no colons before strings. For example, enter the terms small, very, and extremely, as shown in the TEST\_NP grammar concept definition below. There are no symbols appended to these strings because they are matched as literals.

*Display 19-6 Intermediate Concepts*



In the example above, two intermediate concepts are specified. The first intermediate concept TEST\_NP references the second intermediate concept ADJ\_GROUP.

The following line:

---

```
*TEST_NP = :Det *ADJ_GROUP :N
```

refers to the \*ADJ\_GROUP intermediate concept. \*ADJ\_GROUP represents the adjective matches that can be made after a determiner and before a noun. (A determinant is a noun modifier such as *the*, *an*, or *a*.) This intermediate concept is specified using the following lines:

```
*ADJ_GROUP = small
```

The line above specifies a match on the term *small*.

```
*ADJ_GROUP = very :A
```

This line above specifies a match on the term *very* and the adjective that follows *very*.

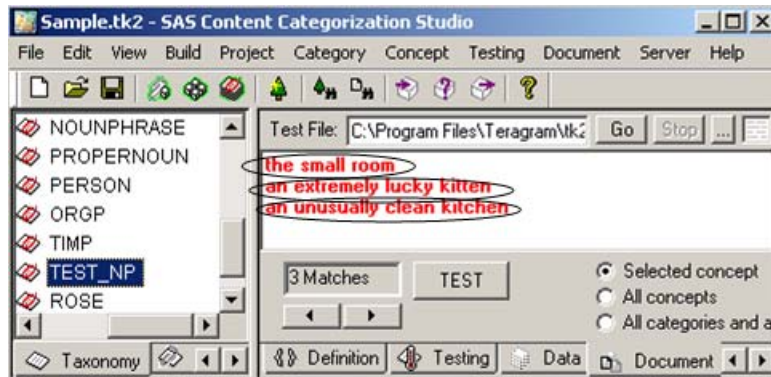
```
*ADJ_GROUP = extremely :A
```

This line above specifies that the term *extremely* is matched if *extremely* is followed by a match on the part-of-speech tag adjective.

```
*ADJ_GROUP = :Adv :A
```

The line above specifies that the part-of-speech tag Adv returns a match on an adverb if an adjective follows the adverb.

Figure 19-4 Intermediate Concept Examples



In summary, use intermediate concepts, to progressively build grammar rules.

---

## 19.5.4 Defining Dependencies in Grammar Rules

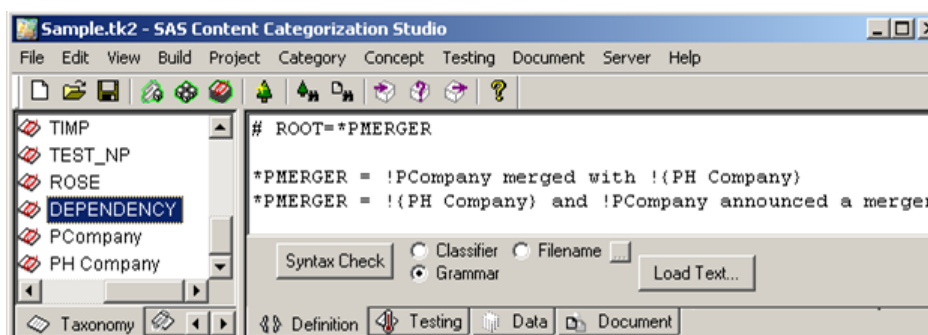
Use the definition of one or more concepts in the grammar rule that you are writing, whether these are grammar or classifier concepts. The new rule uses the entire definition of the source concept as part of its own definition. This is true whether the source definition is for a classifier, grammar, or both types of concepts.

In order to reference a source concept, place an exclamation mark (!) before the name of a referencing, or target, concept such as !PCompany. If there is a space in the name of the referenced concept, you should specify an exclamation mark followed by a closed set of curly brackets surrounding the concept name (!{ }). For example, write !{PH Company}.

Continuing with the !PCompany example above, the following two concepts are defined as follows. The classifier concept, PCompany, lists publicly traded companies while the PH Company concept lists privately held companies.

The following grammar concept, PMERGER, defines a relationship between itself and these two concepts and uses the PCompany and PH Company definitions as part of its own definition.

*Display 19-7 Creating a Dependency*



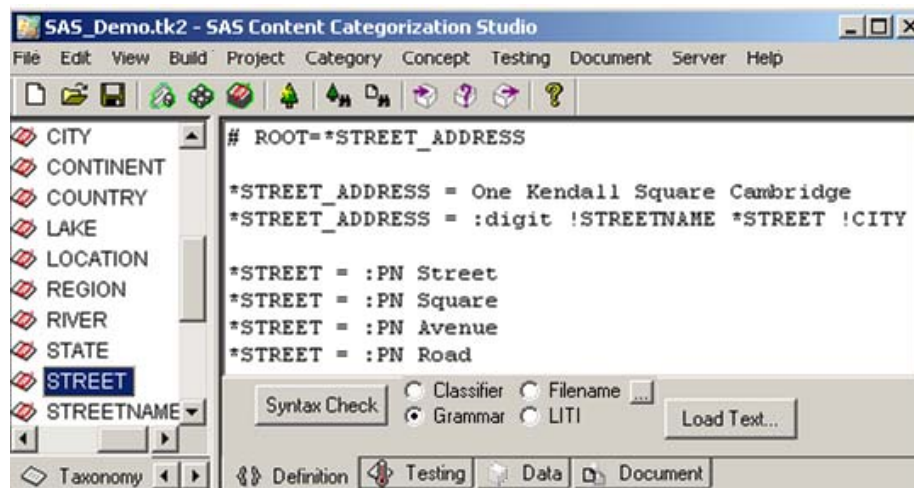
In this example, mergers between publicly traded and privately traded companies are matched. Any changes to the definitions of either the PH Company or the PCompany can also affect the results returned to the PMERGER company.

---

## 19.5.5 Write a Complete Grammar Rule

After you read the preceding subsections, write your complete grammar concept definition. See the example shown below:

*Display 19-8 Sample Grammar Concept Definition*



To write your grammar concept, use the following steps:

1. Click **Grammar** in the Definition window.
2. Type in # ROOT=\* and specify the name of the concept.
3. Enter the syntax for the rule. In the example shown above, STREET\_ADDRESS is defined twice:
  - First, a match is specified as an exact match on the string One Kendall Square Cambridge.
  - Alternatively, a match can occur on the second STREET\_ADDRESS line. This line specifies that a number (digit) is followed by a reference to another concept in this taxonomy (STREETNAME). These matching terms are followed by a match on the intermediate concept STREET and followed by a match on the CITY concept in this taxonomy.

---

**Note:** Specify case-sensitive terms, or reference classifier definitions that list the forms of these words that you want to match. Alternatively, enter the `:PN` tag for matches on proper nouns. (For more information, see Section 19.4.3.C *PN Tag* on page 478.)

---

4. (Optional) Write any intermediate concept definitions such as `STREET`. for more information, see Section 19.5.3 *Writing Intermediate Concepts* on page 485.
5. (Optional) If you have not already defined your classifier concepts referenced by this grammar rule, write these now. For more information, see Section 19.5.4 *Defining Dependencies in Grammar Rules* on page 487.



Here are some examples of possible matches on this grammar concept:

- One Kendall Square Cambridge
- 11 Park Avenue Brighton
- 87 Huron Road Newton

The following strings are not matched:

- *Columbus Avenue*. A number does not precede this string, nor does a city follow this match.
- *Quincy* matches the Classifier concept `CITY`. However, no other rule requirements are met.

- 
- *101 Broadway New York* does not match because *Broadway* is not a match for the intermediate concept `STREET`.
  - *One Kendall square Cambridge* does not match because the word *square* begins with a lower- and not an uppercase letter.

The example below defines a rule that specifies various matches on noun-phrases in input documents.

*Example 19-2: Extracting a Noun Phrase*

```
# ROOT= *SAMPLE_NP
*SAMPLE_NP = :Det :N
*SAMPLE_NP = :Det :A :N
*SAMPLE_NP = :Det very :A :N
```

The `ROOT` line indicates the name of the concept. The right-hand side of each rule contains strings and parts of speech. The lines that define `*SAMPLE_NP` are explained below:

```
*SAMPLE_NP = :Det :N
```

The line above specifies that `SAMPLE_NP` can extract a determiner (`Det`) followed by a noun (`N`). For example, the noun-phrase *the house* is matched in an input document.

```
*SAMPLE_NP = :Det :A :N
```

The line above specifies that a match for `SAMPLE_NP` can also be a determiner (`Det`), followed by an adjective (`A`), and followed by a noun (`N`). In this example, the noun-phrase *the big house* returns a match.

```
*SAMPLE_NP = :Det very :A :N
```

The line above specifies that a match for `SAMPLE_NP` can also be a determiner (`Det`) followed by a match on the string `very`. If this match occurs and if the match is followed by an adjective (`A`) that is followed by a noun (`N`), a match for the concept is returned. In this example, the noun-phrase *the very big house* returns a match.



---

# Chapter: 20

## Testing Concepts

---

- *Overview of Testing Concepts*
- *Understanding Testing Results*
- *Setting the Priorities*
- *Testing with the Concordance in the Document Tab*
- *Testing with the Concordance in the Testing Tab*

### 20.1 Overview of Testing Concepts

This chapter explains the testing operations and settings that differ from those used for categories. This chapter assumes that you understand the contents of the following chapters that specify the testing operations for categories:

- Chapter 12: *Assembling Testing Sets*
- Chapter 13: *Batch Testing*
- Chapter 14: *Testing with One Document*
- Chapter 15: *Other Testing Operations*

Use the information in these four chapters to assemble your testing directory and to test your documents against the concepts branch of the taxonomy. You can then use these operations with the features that are specific to concepts.

---

## 20.2 Understanding Testing Results

When you test a concept using a testing folder, the following information is available:

### Test File

the name of the test file

### Result

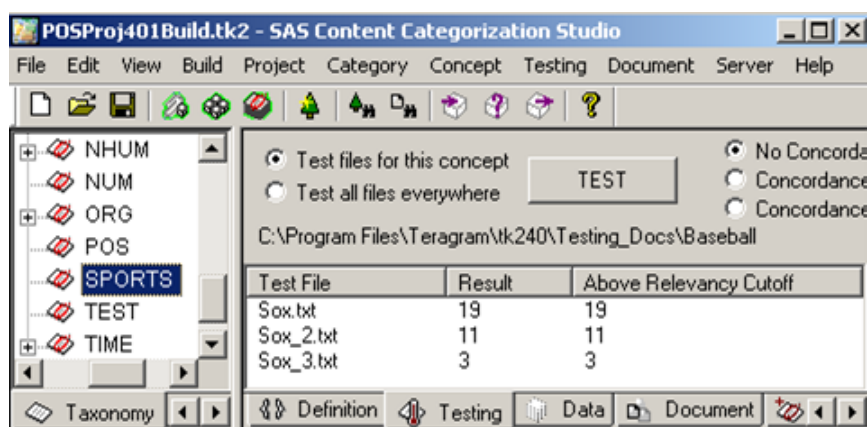
the number of matching terms

### Above Relevancy Cutoff

the number of instances of matched terms that exceed the relevancy cutoff. This is true whether this number is derived from the **Default Relevancy Cutoff** setting in the Project Settings - Concept window or the **Relevancy Cutoff** setting in the Data window. If the **Relevancy Cutoff** setting is specified, this concept-specific setting overrides the **Default Relevancy Cutoff** setting for all concepts.

In the following example, the **Default Relevancy Cutoff** and the **Relevancy Cutoff** settings are both set to the default specifications of 0.

*Display 20-1 Test Results*



---

**Note:** These matches include all instances. You can see the matching terms highlighted in red in the Document window.

---

## 20.3 Setting the Priorities

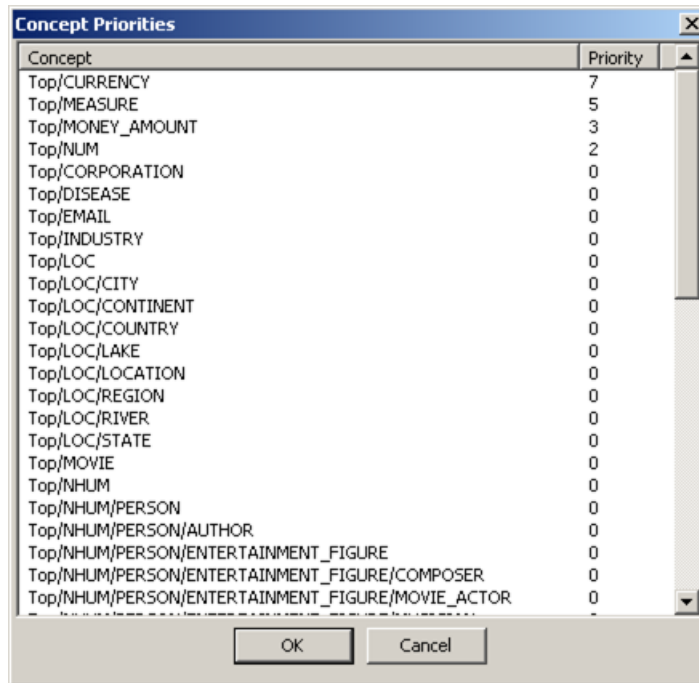
Specify priorities when you want to match one concept over another in the case of duplicate definition matches. For example, you might have several concepts that use numbers as part of the definition. In this case, you can choose to return a match on CURRENCY even when the document also matches MONEY and MEASURE.

When you specify individual priority settings for one or more concepts in the Data window, these results are displayed in the Concept Priorities window. This window lists the priority settings so that you know what concepts are prioritized when an input document matches two or more concepts.

To use the Concept Priorities window, complete these steps:

1. Specify a priority for each concept when you have two or more concepts that could match, but should not.

- 
2. Select **Concept --> Priorities**. The Concept Priorities window appears.



3. See a ranked list of concepts according to the priorities that you specified in the Data window. (If you did not specify any priorities, this window does not display any concepts.)
4. (Optional) Click **Concept** to see the list of the concepts listed from A - Z.
5. (Optional) Click **Priority** to rank the concepts according to matching priority.
6. Click **OK** to close this window.

---

## 20.4 Testing with the Concordance in the Document Tab

### 20.4.1 An Overview of the Concordance

The concordance feature enables you to see a list of the matched terms, highlighted in red, in an input document. Select the **Concordance** check box in the **Document** tab and make one of the following selections:

**Selected concept**

See all of the matches for the selected concept.

**All concepts**

See all of the matches for all of the concepts in this project.

**All categories and all concepts**

See all of the matches for the entire taxonomy.

The results are displayed according to the selections that you specify. These selections include the operations specified in the following windows:

**Project Settings - Concepts**

Specify settings for the entire project.

**Data** tab for each concept

Override the **Default Relevancy Cutoff** setting in the Project Settings - Concepts window for the selected concept using the **Relevancy Cutoff** field.

**Project Settings - Concordance**

Set the display for the Concordance window. (The Concordance window appears in the **Document** tab.)

### 20.4.2 Determine How the Concordance Is Displayed

To set up the display for the concordance, complete the following steps:

1. Select **Project --> Settings**.

- 
2. Click the **Concordance** tab.



3. Under **For each match show**, choose one of the following settings:

**Before**

(Default: 25) Specify how many characters, words, or sentences to display before the match.

**After**

(Default: 25) Choose how many characters, words, or sentences to display after the match.

**characters**

(Default) Show the specified number of characters with the match.

**words**

Show the specified number of words with the match.

**sentences**

Show the specified number of sentences with the match.

- 
4. Specify the selections that determine the sorting order for matching terms:
    - Document Order**  
(Default) See the matching concepts displayed in the order in which they occur in the document.
    - Matched Text**  
Sort the matches alphabetically.
    - Concept Name, then Document Order**  
Sort by concept name first. Then sort by the order of appearance in the text.
    - Concept Name, then Matched Text**  
Sort the matches by concept name and then alphabetically.
    - Relevancy, then Document Order**  
Sort results by relevancy and then in the order in which they appear in the input text.
    - Relevancy, then Matched Text**  
Sort results so that those that are the most relevant are displayed first and the remainder appear in alphabetical ordering.
  5. Determine how to **Test multiple files**:
    - Hide Filenames**  
(Default) Do not display the names of the files that match in the concordance view.
    - Show Filename**  
Display the test results, and to the right of these results, the name of the file.
    - Show Full Path**  
Display the test results with the name of the file. The full path of this file appears to the right of the results.
  6. Select **Insert text markers** to display text markers in the concordance view of the **Document** tab. These tags appear when you test a single file against multiple concepts. The match text fields display the concept that is the best match for the matched term that is returned. For example,

---

the following tags might be inserted around the matched text,  
<CONCEPT1> matched text </CONCEPT1>.

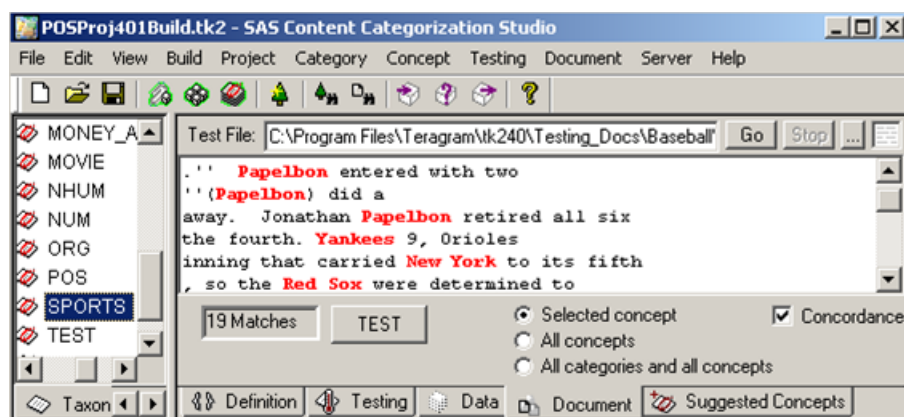
7. Select **Show Relevancy** to display the relevancy of each matched term. Matches exceed the Relevancy Cutoff specification are marked PASS and those that do not are marked FAIL.

### 20.4.3 See the Concordance Terms for a Selected Concept

Use the concordance to see a list of the terms in the input document that match only the selected concept.

To see a list of matching terms for a selected concept, complete these steps:

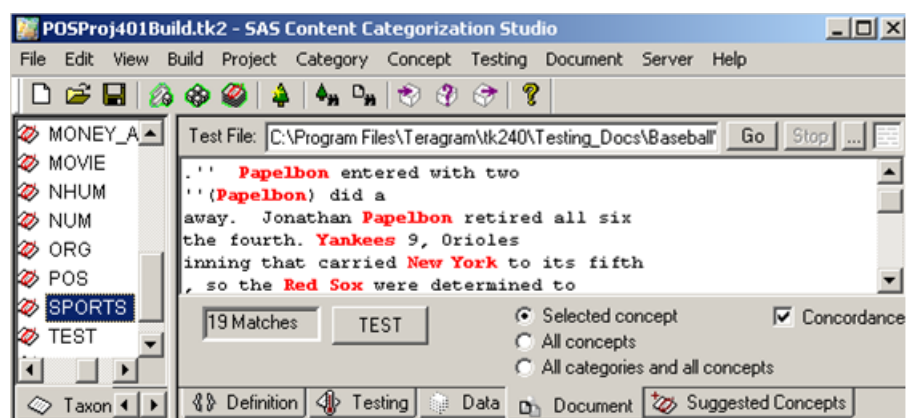
1. Test the testing documents for a selected concept in the **Testing** tab.
2. Double-click a tested document. The text appears in the **Document** tab.



3. If you selected **Show Relevancy** in the Project Settings - Concordance window, the matches that exceed the Relevancy Cutoff are displayed.
4. By default, **Selected Concept** is selected. If not, select **Selected Concept**.
5. Select **Concordance**.



6. See the matching terms highlighted in red.



#### 20.4.4 See the Concordance Terms for All

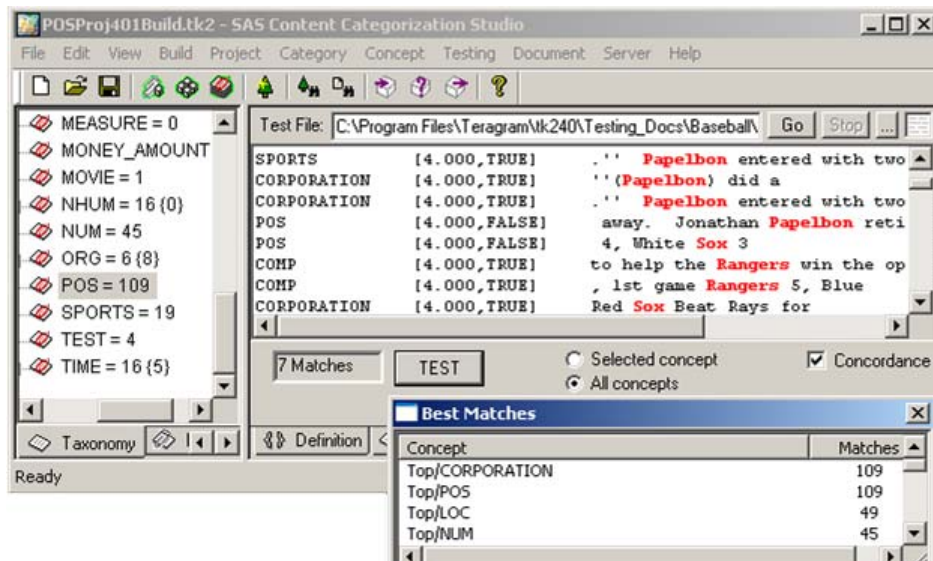
You can choose to see all of the matching terms in an input document for all of the concepts, or for all of the concepts and categories. When you select either of these operations, you can also see the results in the Best Matches window.

The steps that are necessary for the **All concepts** operation are explained in this section. If you want to use the **All categories and concepts** selection, modify these steps as necessary.

To see a list of matching terms for all of the concepts in the taxonomy, complete these steps:

1. (Optional) If a test document is not displayed in the **Document** tab, complete Step 1. and Step 2 on page 498.
2. Select **All concepts**.
3. Select **Concordance**.
4. See the following information from left to right in the Concordance view of the **Definition** tab:
  - The matched concept is displayed. For example, see **SPORTS**, **CORPORATION**, and **POS**.
  - The number of matches are displayed. For example, see 4 . 000.

- If the number of matches exceeds the Relevancy Cutoff specification, see **TRUE**. Otherwise, see **FALSE**. The relevancy is determined by the **Default Relevancy Cutoff** setting in the Project Settings - Concept window, unless the **Relevancy Cutoff** in the Data window is specified.
  - The matching terms are highlighted in red.
  - The number of words, characters, or sentences specified in the Project Settings - Concept window are displayed.
5. Use the Best Matches window that appears.
- See the path to the matched concept under **Concept** and the number of instances of matched terms under **Matches**



- (Optional) Click **Concept** to see the concepts in an alphabetized list.
  - (Optional) Click **Matches** to see the number of matches from highest to lowest.
6. Click **X** to close the Best Matches window.

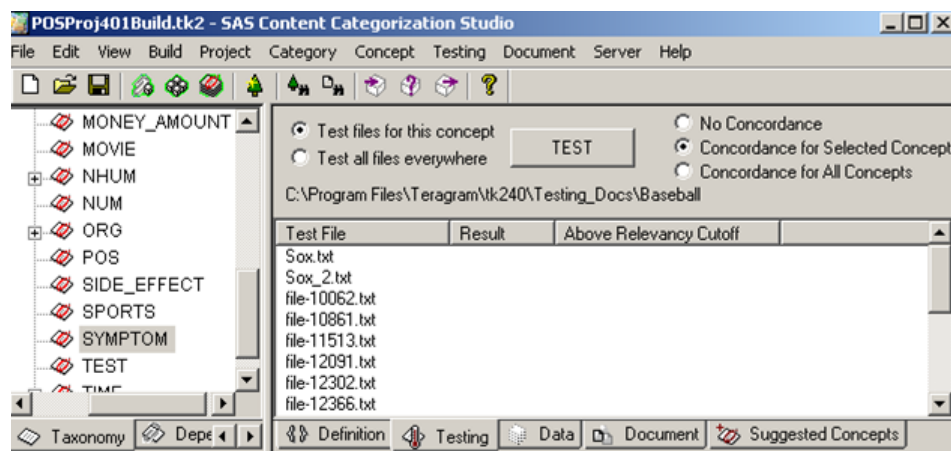
---

## 20.5 Testing with the Concordance in the Testing Tab

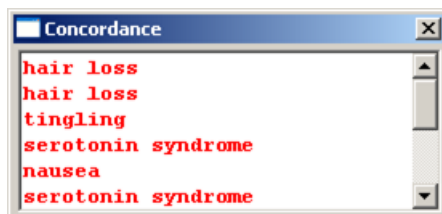
You can use the Concordance windows that are available in the Testing tab to see similar results to those that you see when you use the Document tab. For this reason, see Section 20.4 *Testing with the Concordance in the Document Tab* on page 495 before you use this section.

To use the Concordance operations in the Document tab, complete these steps:

1. Select a concordance operation in the Testing tab. (Make sure that documents are loaded into this window.) For example, select **Concordance for Selected Concept**.



2. Click **TEST** and the Concordance window appears.



3. Click **X** to close the Concordance window.

---

**Hint:** For this example, 0 is specified in the **Before** and **After** fields of the Project Settings - Concordance window.

---

---

# Appendixes

---

- Appendix A: *Regex Syntax and Part-of-Speech Tags on page 505*
- Appendix B: *Program Files on page 511*
- Appendix C: *Recommended Reading on page 535*
- Appendix D: *Glossary on page 537*



---

# Appendix: A

## Regex Syntax and Part-of-Speech Tags

---

- *Regular Expressions*
- *Part-of-Speech Tags*

### A.1 Regular Expressions

#### A.1.1 Rules and Restrictions

The following rules and restrictions apply to regular expressions:

- Any single character **a** (ASCII 1 through 255, subject to escaping restrictions in 14 below) is a regular expression, and it matches precisely that character.
- A character class is a regular expression. One or more characters inside square brackets (**[ ]**), match any of the characters specified inside of the square brackets. For example, **[abc]** matches **abc**. A range inside a character class such as **a-z** matches any ASCII character whose value is between **a** through **z**, inclusive. Any character, including special characters, can appear in a character class. However, **\** (backslash), **-** (hyphen), **[** and **]** (open and closed brackets) are preceded by a backslash. If you want to return a literal match on these characters, see Section A.1.3 *Special Cases* on page 508.
- A negated character class is a regular expression. One or more characters are inside square brackets, with **^** (caret) being the first character to indicate negation. For example, **[^abc]** matches any character except **a**, **b**, or **c**. (If you want to return a literal match on a caret, precede the caret with a backslash.)

---

Also see the table below for more information about the rules and restrictions for regular expressions.

Table A-1: More Rules and Restrictions

If Statement	Explanation
If <b>a</b> and <b>b</b> are regular expressions	then so is <b>ab</b> that matches whatever <b>a</b> matches followed by whatever <b>b</b> matches (concatenation)
	then so is <b>a b</b> that matches either whatever <b>a</b> matches or whatever <b>b</b> matches
If <b>a</b> is a regular expression	then so is <b>(?:a)</b> that simply serves as a grouping mechanism without remembering what it was grouping. For example <b>(?:ababb) b</b> matches either <b>abaab</b> or <b>b</b> . This would be difficult to express without the grouping mechanism.
	then so is <b>a*</b> that matches 0 or more occurrences of whatever <b>a</b> matches
	then so is <b>a+</b> that matches 1 or more occurrences of whatever <b>a</b> matches
	then so is <b>a?</b> that matches 0 or 1 occurrences of whatever <b>a</b> matches
	then so is <b>a{n,m}</b> that matches at least <b>n</b> but no more than <b>m</b> concatenated occurrences of whatever <b>a</b> matches
	then so is <b>a{n,}</b> that matches at least <b>n</b> concatenated occurrences of whatever <b>a</b> matches
	then so is <b>a(n)</b> that matches exactly <b>n</b> concatenated occurrences of whatever <b>a</b> matches



---

## A.1.2 Special Characters

The table below lists, and gives extended meaning to, special characters that are used with regular expressions.

Table A-2: 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 <b>[0-9]</b> )
\D	Not a digit (same as <b>[^0-9]</b> )
\w	Word character (same as <b>[a-zA-Z_0-9]</b> )
\W	Non-word character (same as <b>[^a-zA-Z_0-9]</b> )
\s	Whitespace character (same as <b>[\t\n\r\f]</b> )
\S	Non-whitespace character (same as <b>[^\t\n\r\f]</b> )
.	Wildcard (matches any character)
\xh	Hexadecimal number, where <b>h</b> is a hexadecimal character
\xhh	Hexadecimal number, where <b>h</b> is a hexadecimal character
\0o	Octal number, where <b>o</b> is an octal digit
\0oo	Octal number, where <b>o</b> is an octal digit

---

### A.1.3 Special Cases

There are several special cases for regular expressions. These cases include:

[.,(,),?,\*,+,,-,\\,|

for metacharacters such as these to have literal meaning, these metacharacters need to be escaped with a backslash (\). If inside a character class, however, only those metacharacters that are explicitly mentioned need escaping.

No support is provided for the following:

backward references

O as a remembering grouping mechanism.

^ as the beginning-of-line zero-width assertion

\$ as the end-of-line zero-width assertion

---

**Note:** Unlike Perl regular expressions, the ^ and \$ markers are implicitly assumed.

---

## A.2 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 Chapter 19: *Writing Grammar Rules*. Also see the language book for each language that you purchased.

Table A-3: Part-of-Speech Morphological Features

Code	Part-of-Speech	Example
A	adjective	The sky is <i>azure</i> .
ABBREV	abbreviation	etc.
Adv	adverb	He is <i>easily</i> the best candidate.
Asup	superlative adjective	He cooked the <i>best</i> dish.

---

Table A-3: Part-of-Speech Morphological Features (Continued)

Code	Part-of-Speech	Example
C	conjunction	Say nothing of former informers <i>and</i> spies.
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	
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.
Num	number	She just turned <i>seventeen</i> years old.
PN	proper noun	We are going to <i>England</i> for vacation.
PossDet	possessive determinant	It is <i>her</i> choice.
PossPro	possessive pronoun	The choice is <i>hers</i> alone.
PreDet	<i>pre</i> determinant	<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> .

---

Table A-3: Part-of-Speech Morphological Features (Continued)

Code	Part-of-Speech	Example
ProMD	pronoun contracted with modal	If it weren't for him, <i>we'd</i> still be here.
ProV	pronoun contracted with a verb	<i>We're</i> ready.
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	;;,;,.,.
V	verb	You should <i>verbalize</i> your wishes.
V3sg	verb, 3 <sup>rd</sup> person singular	The boy <i>amuses</i> himself throwing rocks.
V3sgn't	verb, 3 <sup>rd</sup> 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..."
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: B

## Program Files

---

- *Overview of the Program Files*
- *The Projects Folder*
- *The Data Folder*
- *Configuration Examples*
- *The Categorization XML File Format*
- *The Concepts XML File Format*

### B.1 Overview of the Program Files

This appendix covers the folders, files, tags, and directives that comprise the SAS Content Categorization Studio application. Use this chapter to specify the settings for your SAS Content Categorization Studio project.

By default, the SAS Content Categorization Studio application is installed in the following folder:

```
C:\Program Files\Teragram\tk240
```

The `tk240` folder has two subfolders. These folders are described in the following sections.

### B.2 The Projects Folder

#### B.2.1 About the Projects Folder

The `Projects` folder contains subfolders for each project. The name of each subfolder has the same name as the project and contains the configuration and binary files that define the project. Most filenames use a language as a prefix.

---

For example, `French.rb.cat` represents the French version of the rule-based categorizer.

---

**Hint:** You can create projects in any folder.

---

The table below provides a brief description of these files:

Table B-1: Program Files in the Projects Folder

Filename	Description
<code>projectname.tk2</code>	This file is the project configuration file where the project settings are specified. For example, the name of the project, languages, names of the categorizer binary data files, name of the categorizer, and the names of the XML files, are stored here.
<code>language.admin.log</code>	This file is an administration log for SAS Content Categorization Studio. This log lists administrative operations that include adding or deleting categories and concepts, building categorizers, and document testing. <b>Note:</b> The log is used for the latest version of the project. The previous contents are deleted when you restart SAS Content Categorization Studio.
<code>language.concept.xml</code>	The definitions for each language-specific concepts taxonomy are stored here in XML format. (The definitions are stored in the <code>&lt;lang&gt;.concept</code> folder.)
<code>language.concepts</code>	This binary data file is generated when you compile the concepts. SAS Content Categorization Studio uses this file to perform concepts extraction on input documents.
<code>language.tx</code>	This file is a binary data file that is generated when you compile the concepts.
<code>language.directory.xml</code>	This definition file is in XML format, of the categorization taxonomy for a specific language.
<code>language.mco</code>	This file is the categorization binary file. Category macro names are stored here.
<code>language.rb.cat</code>	This binary data file is generated and used by the rule-based categorizer.

---

Table B-1: Program Files in the Projects Folder (Continued)

<i>language.stat.cat</i>	This binary data file is generated and used by the statistical categorizer.
<i>language.test.save</i>	This is a list of test results for categories.
<i>language.train.txt</i>	This is a list of the pathnames of the training files used by the automatic rule generator tool.
<i>conceptname.n.def</i>	<p>This is a definition file for a user-created concept. The <i>n</i> specifies the number that indicates the directory level of the concept. See the following examples:</p> <ul style="list-style-type: none"> <li>- 0: Top-level concept, in other words, subordinate only to Top</li> <li>- 1: Subconcept of a top-level concept</li> <li>- 2: Subconcept of a subconcept</li> </ul> <p>Each file contains the list of classifiers, or the set of grammar rules that define the concept. These files are located in the <i>language.concept</i> subfolder.</p> <p><b>Note:</b> These values only apply to projects that have a single-user.</p>
<i>conceptname.n.def.miss</i>	This is the file that contains the definitions for classifier concepts that use <i>Suggested Concepts</i> . This file is located in the <i>language.concept</i> subfolder.

---

**Note:** The taxonomies of both categories and concepts are defined in XML format.

---

---

## B.2.2 SAS Content Categorization Studio File Format

The `projectname.tk2` file is a text file that describes the various settings for the SAS Content Categorization Studio project. The list of directives shown in the table below are supported for `.tk2` files. Any other directive is ignored by SAS Content Categorization Studio.

Table B-2: Core Directives

Directive	Description
<code>project=&lt;project_name&gt;</code>	This directive specifies the name of the SAS Content Categorization Studio project.
<code>language=&lt;language&gt;</code>	<p>This directive specifies the language, or languages, used by a SAS Content Categorization Studio project. The format of <code>&lt;language&gt;</code> is a language name with the first letter capitalized. For example, specify <code>&lt;English&gt;</code>. The specified language also has supporting data files. For more information, see Section B.3 <i>The Data Folder</i> on page 515. These files are usually found in the directory <code>data\&lt;language&gt;</code>. This directory is located underneath the <code>tk240</code> directory. If no supporting data files are present for the specified language, an error is returned when the <code>.tk2</code> file is parsed.</p> <p>(You can create a SAS Content Categorization Studio project that does not contain a language, but this operation is not recommended.)</p> <p><b>Note:</b> Directives that follow a <code>language=</code> tag operate in the context of the preceding language specification.</p>
<code>ParagraphFPAT</code>	Specify this tag for Boolean rules that use the PAR operators such as PAR, MAXPAR, and PARPOS. This setting specifies the markers that define a paragraph break in input documents.
<code>DefaultField</code>	Specify the default XML fields to search when XML documents are processed.
<code>FieldsToIgnore</code>	Specify the XML fields that are ignored when XML documents are processed.
<code>Compatibility Date</code>	Specify this date to make the <code>.mco</code> file built by SAS Content Categorization Studio correspond to an older version of the file format for compatibility purposes. The value for this tag is in the format <code>YYYYMMDD</code> .



Table B-2: Core Directives (Continued)

Directive	Description
<code>categorizer=</code> <code>&lt;categorizer_xml&gt;</code>	<p>This directive specifies the name of the categorizer XML file for a specific language project. A <code>language=</code> tag precedes this directive in the <code>.tk2</code> file or an error is returned.</p> <p>Only the <code>.tk2</code> files that contain a directive for categorization specify this directive. The categorizer XML file is typically named <code>&lt;language&gt;.directory.xml</code>. For example, the <code>.xml</code> file could be named <code>English.directory.xml</code>. This naming convention is not required. You can choose another name for this file.</p>
<code>concept=</code> <code>&lt;concept_xml&gt;</code>	<p>This directive specifies the name of the concept XML file for a specific language project. A <code>language=</code> tag precedes this directive in the <code>.tk2</code> file or an error is returned.</p> <p>Only the <code>.tk2</code> files that contain a directive for concept extraction specify this directive. The concept XML file is typically named <code>&lt;language&gt;.concept.xml</code>. For example, <code>.xml</code> file could be named <code>English.concept.xml</code>. This naming convention is not required. You can choose another name for this file.</p>
<p><b>Note:</b> Not all <code>.tk2</code> files contain the categorizer and concept directives. You could also have a language project that contains neither categorization nor concept extraction, although this type of project is not recommended.</p>	

## B.3 The Data Folder

The `data` folder contains one or more language folders such as `English` or `French`, depending on the languages that you purchased from SAS. Use the tags in the following two tables for categories and concepts. The third table in this section describes the deprecated directives for concept extraction.

---

To configure your categorizer, use the directives in the table below. These directives operate in the context of a `categorizer=` directive.

Table B-3: Categorization Configuration Directives

Directive	Description
<code>stat_cat=</code> <code>&lt;stat_cat_file&gt;</code>	Specifies the statistical categorization file that SAS Content Categorization Studio uses during this session.
<code>use_leaf_macros</code>	Specify macro names in a SAS Content Categorization Studio project and enable them to reference the leaf name of a category. For example, reference <code>Football</code> , instead of a full pathname such as <code>Top/Recreation/Sports/Football</code> .
<code>export_short_mco</code>	Export an additional <code>.mco</code> file named <code>&lt;language&gt;.short.mco</code> when you build a rule-based categorizer. This <code>short.mco</code> file returns category leaf names instead of full paths. For example, return <i>Baseball</i> instead of <code>Top/Recreation/Sports/Football</code> .
<code>export_utf8_mco</code>	Export an additional <code>.mco</code> file named <code>&lt;language&gt;.utf8.mco</code> for UTF-8 languages. This file contains the category names in UTF-8 characters instead of the Latin-1 display names used internally by SAS Content Categorization Studio.
<code>uses_concepts</code>	Reference concepts with the categories in a SAS Content Categorization Studio categorization project.
<code>never_expand</code>	Do not expand Boolean rule terms in categorization projects that end with <code>@</code> , <code>@N</code> , or <code>@V</code> . Instead, literally match each term in the rule. If you do not specify this operation, these words are automatically inflected when these suffixes are present.
<code>expand_at</code> <code>_compile</code>	Expand rule terms in categorization projects that end with <code>@</code> , <code>@N</code> , or <code>@V</code> in the <code>.mco</code> file.
<code>expand_all</code>	Expand all rule terms in categorization projects.
<code>use_auto_save</code>	Automatically save the category XML file before you build the categorizer.
<code>has_auto_rules</code>	Use the automatic rule generator tool for a categorization project.
<code>should_rebuild</code> <code>_categories</code>	Rebuild the <code>.mco</code> file before any testing operations are performed.

Table B-3: Categorization Configuration Directives (Continued)

Directive	Description
<code>should_rebuild_stat_cat</code>	Rebuild the statistical categorization file before the user performs any testing operations.
<code>active_cat=rule stat</code>	Specify the currently active categorizer.

The following directives operate in the context of a `concept=` directive. In other words, this directive must precede the usage of any of the following directives:

Table B-4: Concept Configuration Directives

Directive	Description
<code>export_utf8_concepts</code>	Export an additional CONCEPTS binary file named <code>&lt;language&gt;.utf8.concepts</code> . This file contains the concept names in UTF-8 characters instead of the Latin-1 display names that are used internally by SAS Content Categorization Studio. This operation is for UTF-8 languages.
<code>should_rebuild_classifier</code>	When you rebuild the CONCEPTS binary file, also rebuild the CLASSIFIER binary file ( <code>.concepts</code> file). This file is embedded inside the CONCEPTS binary file.
<code>should_rebuild_concepts</code>	Rebuild the CONCEPTS binary file ( <code>.concepts</code> file) before the user does any testing.
<code>should_rebuild_context</code>	Rebuild the CONTEXT binary file embedded inside of this CONCEPTS binary file when you rebuild the CONCEPTS binary file.
<code>concepts_file=&lt;concepts_file&gt;</code>	Specify the name of the CONCEPTS binary file that SAS Content Categorization Studio automatically loads when SAS Content Categorization Studio starts.

The directives in the table below are no longer necessary. SAS Content Categorization Studio automatically sets the values that were formerly set using these directives. These directives are supplied for users who purchased

---

custom data files. Each of these deprecated directives applies to concept extraction.

Table B-5: Deprecated Directives

Deprecated Directive	Description
<code>tagger=&lt;tagger_file&gt;</code>	Specifies the tagger file.
<code>lexsyntax=&lt;lexsyntax_file&gt;</code>	Specifies the lexical syntax file.
<code>case_semantic=&lt;case_semantic_file&gt;</code>	Specifies the case semantic file.
<code>semantic_dictionary=&lt;semantic_dictionary_file&gt;</code>	Specifies the semantic dictionary file.
<code>character_map=&lt;character_map_file&gt;</code>	Specifies the character map file.
<code>utf8_tokenizer=&lt;utf8_tokenizer_file&gt;</code>	Specifies the UTF-8 tokenizer file.
<code>tokenizer=&lt;tokenizer_file&gt;</code>	Specifies the tokenizer file.
<code>user_semantic_dictionary=&lt;user_semantic_dictionary_file&gt;</code>	Specifies the user semantic dictionary file.
<code>precompiled_data=&lt;precompiled_data_file&gt;</code>	Specifies the precompiled data file.
<code>missing_concepts_project=&lt;missing_concepts_project_file&gt;</code>	Specifies the missing concepts project file.

---

**Note:** Like the active directives, the deprecated directives also operate in the context of a `concept=` directive. In other words, this tag precedes the usage of any of these directives.

---

---

## B.4 Configuration Examples

### B.4.1 A Single Language Project

See the following example of a .tk2 file for a SAS Content Categorization Studio project that contains a single English language project with both categorization and concept extraction.

*Example B-1: Single Language Project*

```
project=News
language=English
categorizer=English.directory.xml
uses_concepts
never_expand
should_rebuild_categories
concept=English.concept.xml
should_rebuild_classifier
should_rebuild_concepts
```

### B.4.2 Multiple Language Projects

See the following example of a .tk2 file for a SAS Content Categorization Studio project that contains multiple languages with categorization.

*Example B-2: Multiple Language Project*

```
project=MultiLang
language=English
categorizer=English.directory.xml
never_expand
should_rebuild_categories
language=Chinese
categorizer=Chinese.directory.xml
never_expand
should_rebuild_categories
```

---

## B.5 The Categorization XML File Format

### B.5.1 About the Categorization XML File Format

The categorizer in SAS Content Categorization Studio is defined by an XML file. The XML files that define the taxonomy for your categories is named:

```
<language>.directory.xml
```

where `<language>` is the language represented by the specified categorizer. For example, specify `English.directory.xml`. You can name this file whatever name you choose.

### B.5.2 The Language Encoding Specifications

Every categorization XML file begins with the lines shown in the example below:

*Example B-3: The First Two Lines of the Categorization XML File*

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<TeragramDirectoryStructureV3>
```

If the language that is represented by the categorizer is a UTF-8 language, the following line is substituted for the first line in the example above:

```
<?xml version="1.0" encoding="UTF-8"?>
```

### B.5.3 The Project Settings

The project settings tags listed in this section are specified in the file. These tags immediately follow the two lines displayed in Section B.5.2 *The Language Encoding Specifications* above. For more information, see Section B.5.6 *A Sample Categorization XML File* on page 525. All of these tags are optional. If these tags are not specified, the default settings for the categorization project are used. When specified, these tags affect the entire project.

The tags that are listed below should retain the specified format when these tags are used in the categorization XML file. Exceptions to this rule are cited in the relevant description of the specified tag.

---

Tag: you specify

value: is the specification for the tag

See the following example:

```
<Tag><![CDATA[value]]></Tag>
```

On the other hand, if the setting is already specified, the tag could be written as `<tag/>`. The project settings tags, with their respective values, are listed and described in the table below:

Table B-6: Categorization Project Settings XML Tags

Tag	Description and Values
Default CategoryBias	Specify a number that is added to all of the relevancy scores. For example, make the scores fall within a specified range.
QueryIP	Specify the IP address of the server to use for a server query.
QueryPort	Specify the port to connect to on the server that is specified by QueryIP.
QueryServer Fields	Specify the fields to return when the server is queried.
ResultsPerPage	Specify how many results are returned on each results page when the server is queried.
CaseSensitive Concepts	Match concepts in a case-sensitive manner.
UseClassifier	Optimize the .mco file for compilation speed. By default, this file is optimized for matching speed. <b>Note:</b> Unless you build a large taxonomy of concepts, the difference between the default setting and this specification is minimal.
Individual Anchors	Treat each instance of an XML field within one document as a separate block of text. In other words, if there are two occurrences of the <code>&lt;sec&gt;</code> tag in one document, both text blocks are searched as if they are separate. If this tag is not present, all instances of the <code>&lt;sec&gt;</code> tag are searched as one text block. <b>Note:</b> This setting can affect how certain Boolean rule operators perform matching. For example, how the caret symbol (^) and the dollar symbol (\$) are used. These symbols are used to match a term in the first or last, respectively, occurrences of an XML tag.

Table B-6: Categorization Project Settings XML Tags (Continued)

Tag	Description and Values
Default Relevancy Cutoff	Specify a relevancy number. Documents with a relevancy number that fall below this score are considered <i>conditionally</i> passing. This can also be set on a per-category basis.
Uncategorized PopulateDir	This tag specifies the directory where documents that did not match any categories in the taxonomy during the <b>Populate Testing Paths</b> operation are placed.
RelevancyType	This tag specifies the relevancy algorithm that is used when relevancy scores for documents are computed:  1: operator-based  2: frequency-based  3: zone-based
UseLongest Match	1: This number specifies that the longest substring match in a category rule is returned as a match. For example, if a category rule matches the terms <i>business</i> and <i>business travel</i> , <i>business travel</i> is returned as the match. <i>business</i> is not returned as a match.  0: This number specifies that all substrings are returned as matches. Using the example above, the phrase <i>business travel</i> returns matches for both <i>business</i> and <i>business travel</i> .
UTF8Testing Docs	1: The testing documents are in UTF-8 format. (This is possible even if the language of the categorizer is a language that uses Latin-1 encoding.)  0: The testing documents use Latin-1 encoding
UniqueCategory IDCheck	1: The unique ID metadata for each category is checked by SAS Content Categorization Studio to ensure that it is unique.  0: The check above is disabled. You can set duplicate IDs.
SyntaxCheckExe	Specifies the path for a custom syntax checker executable (This file is rarely necessary.).

## B.5.4 The Categories XML File

Each individual category is represented by a `<Topic>...</Topic>` block in the XML file. The tags that are described in this section go between the `<Topic>`



---

and `</Topic>` fields. If a tag is optional, the tag is not necessary but can be parsed.

For an example of the tag format, see Section B.5.3 *The Project Settings* on page 520. Any exceptions are noted in the table below:

Table B-7: Category XML Tags

Tag	Description and Values
StringID	The full path of the category is specified here. For example, see <code>Top/Sports/Basketball/NBA</code> .
catpath	Identical to <code>StringID</code> , this field is deprecated. This field is necessary for backwards compatibility with older projects.
Name	The name of the leaf node for a category. For example, see <i>NBA</i> .
ShortName	This tag is identical to the <code>Name</code> tag.
DisplayName	When the categorizer is Latin-1, the <code>DisplayName</code> is identical to <code>Name</code> and <code>ShortName</code> . For UTF-8 categorizers, this is the UTF-8 name that is displayed in the SAS Content Categorization Studio user interface and not the Latin-1 name that is used internally by SAS Content Categorization Studio.
rules	The rule that is associated with this category is written here. Unlike other fields, this field has a required attribute. For this reason, this tag is expressed not simply as <code>&lt;rules&gt;</code> , but as either of the following selections: <ul style="list-style-type: none"><li>- <code>&lt;rules type="BOOLEAN"&gt;</code>: These rules use Boolean operators.</li><li>- <code>&lt;rules type="LINGUISTIC"&gt;</code>: These rules consist of lists of words and strings.</li></ul>
ratio	<ul style="list-style-type: none"><li>- Linguistic rules: This tag specifies the percentage of terms that are required for a document to be considered a match.</li><li>- Boolean rules: This tag is a placeholder that has no effect on your project.</li></ul>
Thesaurus	(Optional) The terms that might be regarded as synonymous with the name of the category are specified here.
Query	(Optional) A search query that is used to find documents about this topic using a search engine is specified here.
Description	(Optional) A brief description of the category is specified here.

Table B-7: Category XML Tags (Continued)

Tag	Description and Values
Comments	(Optional) Comments about the category are listed here. For example, these comments might include a reviewer's notes.
RelatedLinks	(Optional) URLs that contain additional information about the category are listed here.
Author	(Optional) This tag specifies the author of this category rule.
RelevancyCutoff	(Optional) The relevancy number is specified here. Documents that fall below this number are considered <i>conditionally</i> passing for this category.
CategoryBias	(Optional) A number that is added to all of the relevancy scores for this category is specified here. For example, use this tag to make the scores fall within a certain range.
RelevancyBias	(Optional) This number is multiplied by all relevancy scores for this category. This operation boosts the relevancy for this category in relation to other categories in this taxonomy.
CreationDate	(Optional) The date that this category was created is specified in text format. For example, enter February 5, 2007.
ModificationDate	(Optional) The date that this category was last modified is specified in text format.
UniqueCategoryID	(Optional) A unique ID for the category is specified here. For example, specify an internal code or a Library of Congress number.
IsHidden	(Optional) The presence of this tag (in the format <IsHidden/>) indicates that a category is disabled. Disabled categories are used internally for rule matching. However, they do not appear as rule matches in SAS Content Categorization Studio, and are often used to define <i>filters</i> that other categories can reference. These filters should not be matched on their own. For example, a filter might be used to eliminate all documents that came from a particular source or author based on the XML tags.
AutoRules	(Optional) This tag specifies a list of words and phrases obtained by SAS Content Categorization Studio when the automatic rule generator tool is used to develop category rules.

Table B-7: Category XML Tags (Continued)

Tag	Description and Values
NumPopulateMatches	(Optional) The number of documents that matched this category the last time <b>Populate Testing Paths</b> was performed is specified here.
NumPopulateRelevantMatches	(Optional) The number of documents that matched this category and were above the <b>Relevancy Cutoff</b> the last time <b>Populate Testing Paths</b> was performed is specified here.
NumPopulateChildMatches	(Optional) The number of documents that matched the children of this category the last time <b>Populate Testing Paths</b> was performed is specified here.
TestPath	(Optional) The testing path for this category is specified here. The testing folder contains documents that are used to refine the category rule.
TrainPath	(Optional) The training path for this category is specified here. The training folder contains documents necessary for the statistical categorizer and the automatic rule generator tool.
SymbolicLinks	(Optional) If this category is a symbolic link, this tag specifies the name of the linked category.

### B.5.5 Closing the File

The categorization XML file is closed when you write a tag that matches the opening tag with the addition of the forward slash (/):

```
</TeragramDirectoryStructureV3>
```

To see the opening lines of this file, see Section B.5.2 *The Language Encoding Specifications* on page 520.

### B.5.6 A Sample Categorization XML File

The example below provides a sample of a categorization XML file that uses the tags specified in the following sections:

- Section B.5.2 *The Language Encoding Specifications* on page 520
- Section B.5.3 *The Project Settings* on page 520

- 
- Section B.5.4 *The Categories XML File* on page 522
  - Section B.5.5 *Closing the File* on page 525

#### Example B-4: Categorization XML File

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<TeragramDirectoryStructureV3>
<RelevancyType><![CDATA[1]]></RelevancyType>
<Topic>
<StringID><![CDATA[Top]]></StringID>
<catpath><![CDATA[Top]]></catpath>
<rules type="LINGUISTICS">
<![CDATA[]]>
</rules>
<TestPath><![CDATA[C:\tk240\Projects\TeragramDemo\
testing\Top]]></TestPath>
<DisplayName><![CDATA[Top]]></DisplayName>
<Name><![CDATA[Top]]></Name>
<ShortName><![CDATA[Top]]></ShortName>
</Topic>
<Topic>
<StringID><![CDATA[Top/Human Interest]]></StringID>
<catpath><![CDATA[Top/Human Interest]]></catpath>
<rules type="BOOLEAN">
<![CDATA[(OR,"human interest")]]>
</rules>
<ratio><![CDATA[10]]></ratio>
<TestPath><![CDATA[C:\tk240\Projects\TeragramDemo\
testing\Top\Human Interest]]></TestPath>
<DisplayName><![CDATA[Human Interest]]></DisplayName>
<Name><![CDATA[Human Interest]]></Name>
<ShortName><![CDATA[Human Interest]]></ShortName>
</Topic>
<Topic>
<StringID><![CDATA[Top/Human Interest/Animals]]>
</StringID>
<catpath><![CDATA[Top/Human Interest/Animals]]>
</catpath>
<rules type="LINGUISTICS">
<![CDATA[]]>
</rules>
<TestPath><![CDATA[C:\tk240\Projects\TeragramDemo\
testing\Top/Human Interest/Animals]]>
</TestPath>
<DisplayName><![CDATA[Animals]]></DisplayName>
```

---

```
<Name><![CDATA[Animals]]></Name>
<ShortName><![CDATA[Animals]]></ShortName>
</Topic>
</TeragramDirectoryStructureV3>
```

## B.6 The Concepts XML File Format

### B.6.1 About the Concepts XML File Format

Concepts extraction in SAS Content Categorization Studio is defined by an XML file and a set of text files located in a specified subdirectory of the project. Unlike categorization, the text of each concept definition is stored in a file in a specific subdirectory because of the size of the concept definitions. For more information, see Section B.6.3 *The Concept Files* on page 532. This section of this Appendix describes the format of the concepts XML file and describes where the text files are located. The XML files that define taxonomies for concepts are named:

```
<language>.concept.xml
```

<language> is the language represented by the specified concept extractor. For example, specify `English.concept.xml`. However, you can name this file any name that you choose.

#### B.6.1.A The Language Encoding Specifications

Every concept XML file begins with the following two lines shown in the example below:

*Example B-5: The First Two Lines of a Concept XML File*

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<TeragramConceptStructureV2>
```

If the language that is represented by this concept extractor is a UTF-8 language, the following line is substituted for the first line in the example above:

```
<?xml version="1.0" encoding="UTF-8"?>
```

---

### B.6.1.B The Project Settings

The tags for the project settings that are listed in this section are specified in the concepts file immediately following the two lines displayed in Example B-5 above. For more information, see Example B-6 on page 533. All of the project settings tags are optional. If these tags are not specified, the default settings for the concept extraction project are used. When specified, these tags affect the entire project.

For an example of the tag format, see Section B.5.3 *The Project Settings* on page 520. The concept project settings tags, with their respective values, are listed and described in the table below:

Table B-8: Concept Project Settings XML Tags

Tag	Description and Values
ShouldTokenize	The text of classifier concepts is automatically tokenized if this tag is present (in the format <code>&lt;ShouldTokenize/&gt;</code> ). Specify this tag for new projects. You can use this tag for older projects only when tokenization is not automatically performed, and the format of the classifier files reflects this lack of tokenization.
ShouldUppercase	Classifiers are matched in a case- <i>ins</i> sensitive manner when this tag is present in the format <code>&lt;ShouldUppercase/&gt;</code> .
ShouldNormalize	Latin-1 characters in classifier concepts are treated as their ASCII equivalents for the purposes of matching when this tag is present in the format <code>&lt;ShouldNormalize/&gt;</code> .
ProduceFPAT	The .CONCEPTS file is optimized for matching speed when this tag is present in the format <code>&lt;ProduceFPAT/&gt;</code> . By default, the file is optimized for compile speed.  <b>Note:</b> Unless you write large concepts, the performance difference between matching speed and compile speed is minimal.
ConcordanceSortTp	This tag specifies sorting for the concordance results, or matches:  0: according to the order of the concepts in the text  1: alphabetically  2: by concept name and then by the order in which the results appear in the text  3: by concept name and then alphabetically

Table B-8: Concept Project Settings XML Tags (Continued)

Tag	Description and Values
ConcordanceContextTp	The context used to interpret the ConcordanceNbBefore and ConcordanceNbAfter tags is specified with these settings:  0: characters  1: words  2: sentences
ConcordanceNbBefore	The number of characters, words, or sentences before a concept match in the concordance results is determined by this specification. Whether characters, words, or sentences are used depends on the value set in ConcordanceContextTp.
ConcordanceNbAfter	The number of characters, words, or sentences after a concept match are displayed in the concordance results is determined. (The value set in ConcordanceContextTp determines whether characters, words, or sentences are used.)
ConcordanceInsertMarker	A marker is placed in the concordance results to highlight the matched concepts when this tag is present, in the form <code>&lt;ConcordanceInsertMarker/&gt;</code> .
MatchTp	Concept matches are determined by this specification when there are overlapping strings. For example, <i>Boston</i> the city instead of the company <i>Boston Scientific</i> . The following values can be specified:  0: all matches are returned  1: the shortest match is returned. For example, see Boston.  2: the longest match is returned. For example, see Boston Scientific.
DefaultConceptRelevancyCutoff	This tag specifies the relevancy number. Below this number documents are considered <i>conditionally</i> matching. This tag can also be set on a per-concept basis.

Table B-8: Concept Project Settings XML Tags (Continued)

Tag	Description and Values
ConceptRelevancyType	This tag indicates that the relevancy algorithm is used when computing relevancy scores for documents. See the following values:  1: frequency-based 2: zone-based
UniqueConceptIDCheck	1: The unique ID metadata for each category is checked by SAS Content Categorization Studio to ensure that it is unique.  0: The check above is disabled. You can set duplicate IDs.

## B.6.2 The Concepts XML File Format

Each individual concept is represented by a `<Concept>...</Concept>` block in the XML file. The tags in this section go between the `<Concept>` and `</Concept>` fields. If a tag is described as optional in the table below, it is not required but can be parsed.

For an example of the tag format, see Section B.5.3 *The Project Settings* on page 520. Any exceptions are noted in the table below:

Table B-9: Concept XML Tags

Tag	Description and Values
StringID	Specifies the full path to the concept. For example, enter <code>Top/Sports/Football/Patriots</code>
definition	Specifies the name of the file that contains the text for this concept. For more information, see Section B.6.3 <i>The Concept Files</i> on page 532.
DefinitionType	Specifies the classifier type for this concept. The supported types are CLASSIFIER, GRAMMAR, and FILENAME.
DefinitionFile	Specifies the full path of an external file that contains the text for this concept. Use this tag only with concepts where the DefinitionType tag is FILENAME.



Table B-9: Concept XML Tags (Continued)

Tag	Description and Values
DisplayName	Specifies the name of the leaf node for the concept for Latin-1 languages. For example, <i>Patriots</i> might be the display name. For UTF-8 languages this is the UTF-8 name, not the Latin-1 name that is used internally by SAS Content Categorization Studio.
Thesaurus	(Optional) Specifies terms that can be regarded as synonymous with the name of the concept.
Query	(Optional) Specifies a search query that might be used to find documents about this topic using a search engine.
Description	(Optional) Provides a brief overview of the concept.
Comments	(Optional) Specifies comments about the concept. For example, these comments might be a reviewer's notes.
RelatedLinks	(Optional) Specifies URLs that contain additional information about the concept.
Author	(Optional) Lists the author of the definition for this concept.
CreationDate	(Optional) Lists the date that this concept was created, in text format. For example, the creation date might be December 12, 2006.
ModificationDate	(Optional) Lists the date that this concept was last modified, in text format.
UniqueConceptID	(Optional) Provides a unique ID for the concept. For example, this ID might be an internal code or a Library of Congress number.
IsHidden	<p>(Optional) Specifies that a concept is disabled. The format of this tag is in the format &lt;IsHidden/&gt;.</p> <p>A disabled concept does not return any results even if documents that match the concept are processed. This tag is useful when you have a project that combines classifier concepts that are always matched with grammar concepts that you do not want to match.</p>
TestPath	(Optional) Specifies the testing path for this concept that is the location of the folder containing the documents that are used to refine the concept definition.

---

### B.6.3 The Concept Files

Classifier concept definitions, unlike category rules, potentially consist of millions of lines of text. When you build a category project, the XML file that represents this taxonomy also contains each of the rules for the individual categories. However, due to the size of concept definitions, the text of each definition is stored in a file in a specific subdirectory. These files are called concept files.

Each concept SAS Content Categorization Studio project has a subdirectory associated with the project. The default location for this subdirectory is located in the following folder:

```
C:\Program Files\Teragram\tk240\Projects
```

---

**Note:** If you install SAS Content Categorization Studio in a different location, specify the correct location.

---

In the subdirectory there is another subdirectory named `<language>.concept` for each language in the SAS Content Categorization Studio project that contains a concept extractor. The concept definition files are found in that subdirectory. For example, if your concepts project is named `News` and uses the English language, the concept files could be located in the following directory:

```
C:\Program Files\Teragram\tk240\Projects\News\
English.concept
```

The `<language>.concept` subdirectories contain a number of text files that represent the text of the concepts in your SAS Content Categorization Studio project. One example of a typical filename is shown in the following example:

```
<name>.<number>.def
```

`<name>` is the name of the concept and `<number>` is an internal identifier used by SAS Content Categorization Studio. You can also choose to specify a name that you choose.

---

## B.6.4 Closing the File

### B.6.4.A About the Closing Tag

The concept extraction XML file is closed using a tag that matches the opening tag with the addition of a forward (/) slash:

```
</TeragramConceptStructureV2>
```

To see the opening lines of this file, see Section B.6.1.A *The Language Encoding Specifications* on page 527.

### B.6.4.B The Sample Concept Extraction XML File

The code in the example below provides a sample of a stand-alone concept XML file that uses the tags specified in the preceding sections:

- Section B.6.1 *About the Concepts XML File Format* on page 527
- Section B.6.2 *The Concepts XML File Format* on page 530
- Section B.6.3 *The Concept Files* on page 532
- Section B.6.4 *Closing the File* on page 533

#### Example B-6: Concept Extraction XML File

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<TeragramConceptStructureV2>
  <MatchTp><![CDATA[0]]></MatchTp>
  <ConceptRelevancyType><![CDATA[1]]></
ConceptRelevancyType>
  <UniqueConceptIDCheck><![CDATA[1]]></
UniqueConceptIDCheck>
  <Concept>
    <StringID><![CDATA[Top]]></StringID>
    <definition><![CDATA[Top.1000000000.def]]></definition>
    <DefinitionType><![CDATA[CLASSIFIER]]></DefinitionType>
    <DisplayName><![CDATA[Top]]></DisplayName>
  </Concept>
  <Concept>
    <StringID><![CDATA[Top/CONCEPT1]]></StringID>
    <definition><![CDATA[CONCEPT1.64.def]]></definition>
    <DefinitionType><![CDATA[GRAMMAR]]></DefinitionType>
    <DisplayName><![CDATA[CONCEPT1]]></DisplayName>
    <UniqueConceptID><![CDATA[FOO]]></UniqueConceptID>
```

---

```
</Concept>
<Concept>
<StringID><![CDATA[Top/CONCEPT4]]></StringID>
<definition><![CDATA[CONCEPT4.64.def]]></definition>
<DefinitionType><![CDATA[GRAMMAR]]></DefinitionType>
<DisplayName><![CDATA[CONCEPT4]]></DisplayName>
<ModificationDate><![CDATA[December 13, 2005]]></
ModificationDate>
<UniqueConceptID><![CDATA[BAZ]]></UniqueConceptID>
</Concept>
<Concept>
<StringID><![CDATA[Top/CONCEPT5]]></StringID>
<definition><![CDATA[CONCEPT5.64.def]]></definition>
<DefinitionType><![CDATA[CLASSIFIER]]></DefinitionType>
<DisplayName><![CDATA[CONCEPT5]]></DisplayName>
<CreationDate><![CDATA[October 28, 2005]]></
CreationDate>
</Concept>
</TeragramConceptStructureV2>
```

---

## Appendix: C

# Recommended Reading

---

The following books are recommended:

- *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*: Use this add-on application to SAS Content Categorization Studio to enable multiple users to build a single project.
- *SAS Content Categorization Server: Administrator's Guide*: Automate the application of the `.mco` and `.concepts` files to input documents.
- *SAS Contextual Extraction Studio: User's Guide*: Use this add-on application to SAS Content Categorization Studio to write complex concept definitions that can include multiple rule types within a single definition.
- *SAS Contextual Extraction Studio: Installation Guide*: Install SAS Contextual Extraction Studio.
- Use the language books for each language purchased to see the comprehensive list of part-of-speech tags that are available for grammar concepts.

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](http://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](mailto:sasbook@sas.com)

Web address: [support.sas.com/pubs](http://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

---

**automatic rule generator**

automatically generate a set of rules, based on your training set of documents, for all of the categories in the taxonomy.

**batch testing**

process of testing all of the testing documents in the testing set against a selected category. Alternatively, choose to test all of the documents in the testing set against the entire taxonomy. In the second case, use testing documents that were not specifically selected for a category to gain comprehensive testing results that simulate real usage.

**branch**

refers to either the category, or the concepts, section of the taxonomy tree. The first node in a branch is either the `Categorizer` or the `Concepts` node. If the project is built with more than one language, each language section is also referred to as a branch.

**categorization**

process of concisely defining the subject matter of a document, in other words, the main idea or subject of the document.

**central repository of documents**

place all of your testing documents into one directory instead of creating a directory structure that matches your taxonomy structure. Alternatively, you can use a central repository as a secondary source for testing documents.

**CJK languages**

abbreviate the Chinese, Japanese, and Korean languages with this acronym. The CJK languages require UTF-8 encoding to support their characters.

**classifiers**

specify a list-based set of terms that are extracted from your documents.

---

**concept**

define an autonomous piece of information such as movie, book, title, and so on.

**concordance**

use with concepts only. A concordance is an alphabetized list of matched terms in context.

**definition**

defines a concept is called a concept definition. Sometimes, this manual uses the word *rule* as a synonym for the word *definition*.

**dependencies**

enable you to use the *entire* rule or definition in another category or concept as *part* of the selected category rule.

**determiner**

is a noun modifier that references the noun within the context of the text.

**disambiguate**

differentiate, based on the context, between two occurrences of the same term. For example, a warm *coat* is not a fresh *coat* of paint.

**document**

refers to an input text. Also see *Text*.

**flat taxonomy**

define parent categories only. All categories are *created equal*, there are no subcategories.

**forward dependency**

specify a category rule or a concept definition that uses the rules of a second category or concept.

**frequency-based ranking**

refers to the number of matching terms that are found in a document.

**grammar**

is defined as the set of rules and conventions that govern the way that words are used.

**grammar concepts**

enable you to identify information and the relationships between these terms.



---

**hierarchical taxonomy**

build a taxonomy with subcategories and possibly subcategories or children of these child nodes in the taxonomy tree.

**IN-CATEGORY FILES**

assemble testing documents to fit the requirements of a selected category.

**inflected word form**

derive a word form from the root of the word. In SAS Content Categorization Studio you can obtain word inflections by appending an at sign (@) sign to a root form specified in a Boolean category rule.

**linguistic rules**

specify the key words that define the categories in the taxonomy.

**MCAT**

use the rule-based category API to categorize the SAS Search and Indexing index.

**match ratio**

specify the percentage of terms in the category rule to be matched.

**metadata**

is data on information.

**nested categories**

define the complex dependencies between category rules within a single taxonomy. Some rules use other category rules that, in turn, can contain another category rule. This complex interrelationship is compounded when one category rule depends on more than one category rule.

**nonterminal**

is a symbol that matches an entity other than itself. For example, a part-of-speech tag such as *N* is a nonterminal symbol. *N* matches any nouns that occur in the input document. Also see *Terminal*.

**OUT-OF-CATEGORY FILES**

assemble testing documents to meet the requirements of a category that is not the selected category. These texts can be part of a central repository of documents or they can be documents that were assembled to meet the requirements of categories other than the one tested.

---

**precision**

measure the relevancy of the matched documents. In other words, the category rule excludes possible matches that do not reflect the subject matter of the category. For example, texts that refer to *rock collections* are not matched for the category *Rock and Roll*.

**priority**

determines the matching concept when one input document matches two or more concepts and no other determiner makes one concept a better match.

**recall**

match all of the relevant texts with the category rule.

**relation**

identify a relationship between multiple single concepts. For example, link the name and title of a person.

**relevancy-biased ranking**

is the measure of the appropriateness of the match for one category within the overall taxonomy.

**relevancy range**

specify the appropriateness of the documents to a specific category.

**reverse dependency**

reference the source category or concept before the referencing node in the taxonomy of the Dependencies window.

**rule**

defines a category. This term is also used, within this manual, to refer to a concept definition.

**rule tree commands**

appear when you right-click on a *Boolean expression* in the Tree View mode of the Rules window for the rule-based categorizer.

**rule tree statement commands**

appear as a drop-down list when right-clicking on a *Boolean statement* in the Tree View mode of the Rules window for the rule-based categorizer.

**simple concept**

is an autonomous piece of information. For example, a simple concept might be the name of a person or entity.

---

**source category**

contains the rule that is used by the target category as its entire rule (symbolic link) or as part of its rule (dependency).

**statement**

specify a unique linguistic term in the category rule that is separated from a Boolean operator in the Tree View mode. The statement forms its own node on the tree.

**string**

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

**structured text field names**

specify searchable field names in an HTML, SGML, or XML document for a rule-based categorizer using Boolean terms.

**target category**

point to the rule of another category (symbolic link), or a rule that uses the entire rule of the source category as part of its rule (dependency).

**taxonomy**

organize a classification structure that can be either a flat or a hierarchical system.

**terminal**

is a type of symbol that matches only itself. Stings are terminal symbols because a match occurs only when the exact string is located in an input document. Also see *Nonterminal*.

**testing Set of documents**

is the set of texts that you use to test the categorizer or concepts extractor.

**testing Taxonomy**

is defined as a directory of testing folders whose structure is identical to that of the categories or concepts that you defined in the Taxonomy window.

**text**

form that a written document, or a Web page takes, can be called a text. Also see *Document*.

---

**threshold**

specify the minimum weight that is necessary to be considered a member of the selected category when a weighted linguistic rule is specified. This numerical threshold is specified in the Rules window using `__Threshold`, followed by the number to be matched or exceeded for each specified term. If the total weight of the occurrences of terms in a selected document equal or exceed this number, the document might be a match for this category.

**training Set**

consists of the 20 documents that you assemble for each category in the taxonomy structure. The training set is used by SAS Content Categorization Studio to automatically generate category definitions for the statistical categorizer and rules for the Automatic Rule Generator tool.

**tokenizer**

processes input documents, breaking streams of characters into words with this binary file.

**weight**

specify a number that equals the weight assigned to each term in a weighted category rule. Assign the most important terms higher weights than those of less significance.

---

# Index

---

-	special symbol .....	279
--	category rule .....	279
	special symbol .....	279
!	category rule .....	279
	special symbol .....	279, 304
! symbol	usage .....	476
# symbol	usage .....	476
#cap symbol	grammar rules .....	480
	usage .....	470, 480
#w symbol	usage .....	470, 480
\$	special symbol .....	304
*	special symbol .....	279, 303
* symbol	usage .....	476
**	special symbol .....	279
*.short.mco file	defined .....	266, 289
+	category rule .....	279
	special symbol .....	279
.mco file	usage .....	8
@	special symbol .....	278, 303
	usage .....	28
@N	special symbol .....	278, 303

---

@V	
special symbol	278, 303
__REGEX__	
classifier concept	452
__TGIF	
defined	453
__TGUNLESS	
defined	453
example	457
__C	
special symbol	279, 303
__C_Q	
special symbol	304
__L	
special symbol	279, 303
__L_Q	
special symbol	304
__Q	
special symbol	304
__tmac	
Boolean rule	339

## A

Abort Compiling Concepts	
Build menu	15
active categorizer	
Build Statistical Categorizer	193
Add Category	
Category menu	16
usage	73, 97
Add Concept	
Concept menu	17
Add Language	
Project menu	15
All categories	
defined	45
All categories and all concepts	
defined	45
All categories and all concepts option	
Best Matches window	45

---

test against .....	40
usage .....	386
All categories option	
define .....	40
usage .....	386
All Concepts	
defined .....	45
All concepts	
define .....	41
All Docs	
Category Test Report window .....	89
Allow Concepts in Rules	
Concept window .....	57, 429
Allow Duplicate ID's	
Concept window .....	58, 430, 473
defined .....	52
Allow Short Macro Names	
defined .....	51
Also populate from subdirectories	
central repository .....	362
Always rebuild before each test	
usage .....	48
Always rebuild before each test option	
.mco file .....	115
Always save before each test	
usage .....	48
Always save before each test option	
usage .....	115
AND operator .....	294, 296
associated data	
categories .....	163
Author field	
defined .....	36
Automatic Rule tab	
access .....	25
availability .....	46
defined .....	26
Automatic Rule window	
usage .....	46
Auto-Rule Generation Max Words	
defined .....	53

---

---

## B

Back button	
Browser .....	20
usage .....	389
Web view .....	45
Backward button	
Rule Matches window .....	86
batch testing	
benefits .....	379
defined .....	373, 374
Best Matches window	
All categories and all concepts .....	45
Category heading .....	81
Document window .....	375
relevancy score .....	81
Rule Matches comparison .....	86
usage .....	81, 86, 393
Best Quality	
defined .....	52
Best Speed	
defined .....	52
Boolean Morphological Expansion	
defined .....	52
Boolean rules	
__TGIF .....	456
__TGUNLESS .....	456
_tmac .....	339
concept .....	453
structured text .....	308
Web document .....	309
Browser	
Home button .....	20
Refresh button .....	20
Browser	
Back button .....	20
Forward button .....	20
Stop button .....	20
Browser View	
Testing menu .....	20



---

Browser View option	
defined .....	45
usage .....	43, 388, 389
Build menu	
Abort Compiling Concepts .....	15
Build Rulebased Categorizer .....	14
Build Statistical Categorizer .....	14
Compile Concepts .....	14
Upload Categorizer .....	14
Upload Concepts .....	15
Upload Liti .....	15
Build Rulebased Categorizer	
Build menu .....	14
Build Rulebased Categorizer option	
usage .....	408
Build Statistical Categorizer	
Build menu .....	14
Build Statistical Categorizer option	
active categorizer .....	193

## C

Case Insensitive Matching field	
defined .....	37
Case Sensitive Matching field	
defined .....	37
categories	
associated data .....	163
copy and paste .....	169
moving .....	174
rename .....	165
Text View .....	223
categorization	
XML file .....	520
Categorizer node	
drop-down menu .....	96
Category Bias field	
default setting .....	261
Weighting of results .....	36
Category heading	
Best Matches window .....	81

---

Category menu	
Add Category .....	16
Clear Generated Rules .....	17
Create Directory Tree .....	17
Delete All Selected Categories .....	16
Delete All Selected Concepts .....	17
Delete Category .....	16
Export All Generated Rules .....	17
Generate Rules Automatically .....	17
Generate Subcategories .....	17
Import Category from Repository .....	16
Rename Category .....	16
category node	
drop-down selections .....	97
category rule	
analyze .....	382
precision .....	373
Rules tab .....	273
Category Syntax Check window	
usage .....	80
Category Test Report window	
access .....	88
All Docs .....	89
In-Cat .....	89
In-Cat% .....	89
information .....	89
Neg .....	89
Neg % .....	89
N-Tot .....	89
Path .....	89
Pop Rel .....	89
Populate .....	89
Prec % .....	89
Total .....	89
usage .....	408
Category window	
relevancy .....	82
Zone-based relevancy option .....	248
central repository	
Also populate from subdirectories .....	362
defined .....	374
Populate Testing Paths .....	361

---

testing .....	374
testing paths .....	361
check info strings	
duplicates .....	451
usage .....	120
Check info strings for duplicates	
usage .....	48
Check match strings for duplicates	
usage .....	48, 120
children	
defined .....	248
classifier concept	
REGEX .....	452
Classifier radio button	
defined .....	30
Clear Generated Rules	
Category menu .....	17
Clear Suggested Concepts	
Concept menu .....	18
Clear Test Document	
Testing menu .....	19
Clear Test Document option	
usage .....	383
Comments field	
defined .....	38
Compatibility Date	
Misc window .....	55
Compile Concepts	
Build menu .....	14
Compile Concepts tab	
usage .....	79
Compile Speed	
defined .....	52
Compile Speed option	
defined .....	59, 431
Completed field	
defined .....	37
concept	
Boolean rules .....	453
copy .....	423
defined .....	413

---

concept definition	
precision .....	359
recall .....	359
concept extraction	
XML file .....	527
Concept menu	
Add Concept .....	17
Clear Suggested Concepts .....	18
Create Directory Tree .....	18
Delete Concept .....	17
Generate Suggested Concepts .....	18
Import all Suggested Concepts .....	18
Import Concept from Repository .....	18
Priorities .....	18
Rename Concept .....	17
Sort classifier .....	17
concept name	
unique .....	415
concept node	
drop-down selections .....	97
Concept Syntax Check	
window .....	122
Concept window	
Allow Concepts in Rules .....	57, 429
Allow Duplicate ID's .....	58, 430, 473
Default Classifier Matching .....	59, 93, 431
Default Relevancy Cutoff .....	60
Export CONCEPTS File with UTF-8 Display Names .....	58, 430, 472
Match Latin-1 equivalent characters .....	58, 430, 445
Match Terms in all Uppercase .....	58, 430
Match XML character references .....	58, 430, 445
Optimize for .....	59
Overlapping Concept Matches .....	59, 430, 472
Relevancy Type .....	60
Tokenize Classifier Terms .....	58, 429
concepts	
intermediate .....	485
moving .....	418
rename .....	420
writing definitions .....	432
Concepts node	
drop-down menu .....	96

---

Concordance	
defined .....	41
concordance view	
TEST button .....	42
Concordance window	
For each match show .....	61
Hide Filenames .....	62, 497
Insert text markers .....	63, 497
Project Settings .....	60, 134
Show Filename .....	62, 497
Show Full Path .....	62, 497
Show Relevancy .....	63, 498
Sort by .....	62
Test multiple files .....	62, 497
usage .....	41
concordance window .....	62
access .....	33
appears .....	42
match displays .....	61
Sort by .....	62
Test multiple files .....	62
copy	
concept .....	423
Copy All Selections	
Edit menu .....	12
Create Categorization from Directories	
Project menu .....	16
Create Categorizer from Directories	
usage .....	95
Create Categorizer from XML	
usage .....	95
Create Directory Tree	
Category menu .....	17
Concept menu .....	18
Create Folders option	
defined .....	38
Create Rule Text from Children	
categories .....	97
Created field	
defined .....	36
Custom Syntax Checker Executable	
Misc window .....	56

---

---

custom syntax checker executable	
language.directory.xml .....	162
Cut All Selections	
Edit menu .....	12

## D

dash symbol	
usage .....	450
Data tab	
categories .....	33
concepts .....	35
defined .....	11, 26
relevancy .....	82
Data window	
categories .....	163
Decrease Font Size	
Testing menu .....	20
Default Category Bias	
defined .....	50
Default Classifier Matching option	
Concept window .....	59, 93, 431
Default Relevancy Cutoff	
defined .....	50
Default Relevancy Cutoff option	
Concept window .....	60
usage .....	258
Definition tab	
defined .....	10, 26
Delete All Selected Categories	
Category menu .....	16
Delete All Selected Concepts	
Category menu .....	17
Delete Category option	
Category menu .....	16
usage .....	97
Delete Concept	
Concept menu .....	17
delete concept	
Dependencies window .....	421

---

Delete Language	
Project menu .....	15
usage .....	95
Delete Selected Test File	
Testing menu .....	18
Dependencies tab	
defined .....	10
Dependencies window	
appear .....	98
delete concept .....	421
Description field	
defined .....	37
Directory for Unmatched Populate Files	
Misc window .....	56
select .....	366
unmatched files .....	366
Disable Substring Matches	
defined .....	51
Disambiguation	
define .....	123
Display Names	
UTF-8 languages .....	120
DIST_n operator .....	295, 298
Document tab	
defined .....	11, 26
Document window	
usage .....	39, 41, 44, 85, 375
Web browser .....	43
duplicate classifier entries	
disambiguation .....	124

## E

Edit menu	
Copy All Selections .....	12
Cut All Selections .....	12
Find in All Rules .....	13
Options .....	14
Paste .....	13
Paste as Macro .....	13
Paste Macro into Rule .....	13

---

Paste Single Node .....	13
Paste Symbolic Link .....	13
Text Find .....	13
Text Replace .....	13
Tree Find .....	13
Tree Replace .....	13
editing and testing definitions	
concepts .....	432
ellipses button	
defined .....	30
Enable Categorization	
Project menu .....	15
Enable Categorizer	
usage .....	95
Enable Concepts	
Project menu .....	16
usage .....	95, 416
END_n operator .....	295, 301
Enter Display Name	
Enter Names window .....	74
Enter name for internal data files	
Enter Names window .....	74
Enter Names window	
access .....	73
Enter Display Name .....	74
Enter name for internal data files .....	74
usage .....	73
Use same name for both fields .....	73
entity	
defined .....	413
existing project	
access .....	100
Exit option	
File menu .....	12
Expand all word forms	
defined .....	53
Expand Forms button	
usage .....	28
Expand Full	
usage .....	94
Expand Fully	
usage .....	96, 98

---



---

Expand Fully option	
usage .....	97
Expand word forms with '@' sign	
defined .....	53
Export All Generated Rules	
Category menu .....	17
Export CONCEPTS File with UTF-8 Display Names	
Concept window .....	58, 430, 472
Export MCO file with UTF-8 Display Names	
defined .....	51
Export Short MCO file	
defined .....	51

## F

FAIL message	
defined .....	377
failing documents	
defined .....	374
File menu	
New Project .....	12
Open Project .....	12
Save Project .....	12
Save Project As .....	12
Filename radio button	
defined .....	30
Find field	
Tree Find window .....	77
Find in All Rules	
Edit menu .....	13
Find Next button	
Tree Find window .....	77
Flag categories/concepts with no definitions	
operation .....	126
Options window .....	48
Flag categories/concepts with no dependencies	
option .....	128
Options window .....	48
For each match show	
Concordance window .....	61

---

Forward button	
Browser .....	20
defined .....	45
usage .....	86, 389
Forward Dependency	
usage .....	98
forward dependency	
defined .....	239
frequency	
defined .....	248
frequency/children	
defined .....	248
Frequency-Based	
defined .....	51
frequency-based relevancy	
defined .....	248
Frequency-Based selection	
Relevancy Type .....	60
Full Test Report	
Testing menu .....	19
Full Test Report option	
get .....	408
usage .....	88
Full Test Report window	
usage .....	87

## G

Generate Rules Automatically	
Category menu .....	17
Generate Subcategories	
Category menu .....	17
usage .....	98
Generate Suggested Concepts	
Concept menu .....	18
Generate Suggested Concepts option	
usage .....	96
Go button	
defined .....	44
usage .....	389

---

grammar concepts	
benefits .....	470
Grammar radio button	
defined .....	30
grammar rules	
intermediate concepts .....	485
overview .....	483
using the #cap symbol .....	480

## H

Hide Display Names for UTF-8 Languages option	
usage .....	49
Hide Filenames	
Concordance window .....	62, 497
Home button	
Browser .....	20
defined .....	45
usage .....	389

## I

icons	
Standard toolbar .....	21
ID field	
defined .....	36
Identical Path Name option	
defined .....	38
Import all Suggested Concepts	
Concept menu .....	18
Import Categorization from XML	
Project menu .....	16
Import Categorizer from XML	
usage .....	95
Import Category from Repository	
Category menu .....	16
Import Concept from Repository	
Concept menu .....	18
Import Failing Test Files	
Testing menu .....	18

---

Import Test Files	
Testing menu .....	18
Import Test Files option	
usage .....	368
In-Cat	
Category Test Report window .....	89
In-Cat%	
Category Test Report window .....	89
In-category files	
usage .....	381
Increase Font Size	
Testing menu .....	20
Indent button	
usage .....	28
Individual Field Anchors	
Misc window .....	56
Insert text markers	
Concordance window .....	63, 497
interface	
view .....	9
intermediate concepts .....	485

## L

language fonts	
UTF-8 encoding .....	103
language node	
drop-down menu .....	95
language.directory.xml	
custom syntax checker executable .....	162
linguistic rules .....	269
benefits .....	264
Ln	
defined .....	29
Load Text	
defined .....	30
Load Text button	
usage .....	28, 271

---

## M

main window	
components .....	10
Match case option	
Tree Find window .....	77
Match Latin-1 equivalent characters	
Concept window .....	58, 430, 445
Match Ratio	
Data window .....	276
default setting .....	267
defined .....	37
usage .....	264, 276
match ratio	
number of terms .....	277
optimize .....	276
special symbols .....	267, 278
Match Terms in all Uppercase	
Concept window .....	58, 430
Match XML character references	
Concept window .....	58, 430, 445
Matching Speed	
defined .....	52
Matching Speed option	
defined .....	59, 431
MAXOC_n operator .....	294, 297
MAXPAR_n operator .....	295, 301
MAXSENT_n operator .....	295, 302
Menu bar	
defined .....	10
MIN_n operator .....	294, 297
MINOC_n operator .....	294, 297
Misc tab	
XML Tags to Ignore .....	56
Misc window	
Compatibility Date .....	55
Custom syntax Checker .....	56
Directory for Unmatched Populate Files .....	56
Individual Field Anchors .....	56
Paragraph Separator .....	55
usage .....	54

---

Use UTF-8 Test Files .....	55
XML Default Field .....	56, 318, 319
Modified field	
defined .....	36

## N

navigating categories and concepts .....	135
Neg	
Category Test Report window .....	89
Neg %	
Category Test Report window .....	89
Never expand word forms	
defined .....	52
usage .....	29
New Project	
File menu .....	12
New Project window	
access .....	12
Next Match button	
usage .....	392
nformation string	
defined .....	448, 452
NOT operator .....	294, 296
NOTDIST_n operator .....	295
NOTIN operator .....	295, 299
NOTINPAR operator .....	295, 300
NOTINSENT operator .....	295, 300
noun form	
category rule .....	278, 303
N-Tot	
Category Test Report window .....	89
Number of Taxonomy Nodes	
View menu .....	14
Number of Taxonomy Nodes option	
usage .....	75
Number of Taxonomy Nodes window	
usage .....	75

---

## O

Open Project	
File menu .....	12
Open Test Document	
Testing menu .....	19
Open window	
access .....	12
Operator-Based	
defined .....	50
Operator-Based selection	
Relevancy Type .....	60
Optimize .....	52
Optimize Auto-Rule Generation for	
defined .....	52
Optimize for	
defined .....	52
Optimize for option	
Concept window .....	59
Options	
Edit menu .....	14
reset .....	113
Options window	
access .....	114
OR operator .....	294, 296
ORD operator .....	295, 299
ORDDIST_n operator .....	295, 299, 301
Out-of-Category files	
usage .....	381
Overlapping Concept Matches	
Concept window .....	59, 430, 472

## P

PAR operator .....	295, 298
Paragraph Separator	
Misc window .....	55
PARPOS_n operator .....	295, 302
part-of-speech tags	
codes .....	508
usage .....	477

---

PASS message	
defined .....	377
Paste	
Edit menu .....	13
Paste as Macro	
Edit menu .....	13
usage .....	98
Paste Macro into Rule	
Edit menu .....	13
Paste Single Node	
Edit menu .....	13
usage .....	97, 424
Paste Symbolic Link	
Edit menu .....	13
usage .....	97
Path	
Category Test Report window .....	89
Pending field	
defined .....	37
pipe symbol	
usage .....	450
Pop Rel	
Category Test Report window .....	89
Populate	
Category Test Report window .....	89
Populate Testing Paths	
Testing menu .....	19
Populate Testing Paths option	
central repository .....	361
Prec %	
Category Test Report window .....	89
precision	
category rules .....	408
concept definition .....	359
define .....	7
Priorities	
Concept menu .....	18
Priority field	
defined .....	37
Program and Project title bar	
defined .....	10



---

project	
projects folders .....	512
Project menu	
Add Language .....	15
Create Categorization from Directories .....	16
Delete Language .....	15
Enable Categorization .....	15
Enable Concepts .....	16
Import Categorization from XML .....	16
Remove Categorization .....	16
Remove Concepts .....	16
Settings .....	16
project name node	
drop-down menu .....	94
Project Settings	
Concordance window .....	60, 134
Project Settings windows	
usage .....	49
Propagate Button	
usage .....	38
Propagate Options	
defined .....	38

## Q

qualified linguistic rules	
special symbols .....	269
writing .....	269
Query field	
defined .....	38
Query Report Fields	
defined .....	54

## R

recall	
concept definition .....	359
define .....	7

---

Refresh button	
Browser .....	20
defined .....	45
usage .....	389
Refresh Tree	
View menu .....	14
Refresh Tree button	
usage .....	362, 396
REGEX	
classifier concept .....	452
Related Links field	
defined .....	38
relevancy	
frequency/children .....	248
relevancy bias	
increase .....	260
Relevancy Bias field	
default setting .....	261
defined .....	36
Relevancy column	
Best Matches window .....	393
defined .....	257
relevancy cutoff	
defined .....	258
Relevancy Cutoff field	
defined .....	36
relevancy scores	
Best Matches window .....	81
documents .....	258
Relevancy Type	
defined .....	50
Frequency-Based selection .....	60
Operator-Based selection .....	60
Zone-Based selection .....	60
Relevancy Type option	
Concept window .....	60
Remove Categorization	
Project menu .....	16
Remove Concepts	
Project menu .....	16
usage .....	96

---

Remove Tags	
Testing menu .....	20
Remove Tags option	
defined .....	45
usage .....	389
rename	
category .....	165
concepts .....	420
Rename Category	
Category menu .....	16
usage .....	97
Rename Concept	
Concept menu .....	17
Replace button	
Tree Find window .....	77
Report duplicate entries	
option .....	121, 124
Report duplicate entries when checking classifier concepts	
usage .....	48
Reports Per Page	
defined .....	54
Restore Populate Results	
Testing menu .....	19
Result column	
heading .....	257
Reverse Dependency	
usage .....	98
reverse dependency	
defined .....	239
Rule Matches window	
Best Matches comparison .....	86
usage .....	85, 86
rules	
concepts .....	435
precision .....	408
Rules tab	
defined .....	10, 26
Rules window	
default setting .....	271

---

## S

Save Project	
File menu .....	12
Save Project As	
File menu .....	12
Save Test Document	
Testing menu .....	20
Save Test Results	
Testing menu .....	18
Save Test Results option	
Testing .....	378
Saved Result column	
usage .....	378
Select a Directory window	
access .....	361
Unmatched Populate Files .....	366
Select a Language window	
usage .....	70, 71, 94
Selected category	
test against .....	45
Selected category option	
define .....	40
Rule Matches window .....	391
Selected concept	
define .....	41
defined .....	45
SENT operator .....	294, 298
Server Address	
defined .....	54
Server Port	
defined .....	54
Server Query button	
usage .....	29
Settings	
Project menu .....	16
Show best matches when testing all	
Test all files everywhere .....	48
usage .....	48
Show Filename	
Concordance window .....	62, 497

---

Show Full Path	
Concordance window .....	62, 497
Show Graphical Populate Results	
Testing menu .....	19
Show Last Test Report	
Testing menu .....	19
Show Relevancy	
Concordance window .....	63, 498
Sort by .....	62
Concordance window .....	62
Sort classifier	
Concept menu .....	17
Sort taxonomies automatically	
option .....	48, 117
special symbol	
- .....	279
-- .....	279
! .....	279, 304
\$ .....	304
* .....	279, 303
** .....	279
+ .....	279
@ .....	278, 303
@N .....	278, 303
@V .....	278, 303
_C .....	279, 303
_C_Q .....	304
_L .....	279, 303
_L_Q .....	304
_Q .....	304
special symbols	
frequency-based relevancy .....	253
linguistic rules .....	253, 264, 277
match ratio .....	267, 278
modify linguistic rules .....	264
qualified linguistic rules .....	269
rule qualifiers .....	267
Standard toolbar	
icons .....	21
standard toolbar	
defined .....	10

---

Start	
menu .....	9
START_n operator .....	295, 300
Statistical categorizer	
build process .....	153
Status Bar	
usage .....	21
status window	
defined .....	44
stemming	
Category rule .....	303
category rule .....	278
Stop button	
Browser .....	20
defined .....	44, 45
usage .....	389
structured text	
Boolean rules .....	308
defined .....	308
subcategory	
rename .....	165
subnodes	
count .....	76
symbol	
usage .....	476
symbolic links	
create .....	281
Syntax Check button	
defined .....	30
Syntax Checking	
window .....	48

## T

Taxonomy as Text	
View menu .....	14
window .....	48
Taxonomy as Text option	
usage .....	127
taxonomy messages	
removed .....	383

---

Taxonomy tab	
defined .....	10
Taxonomy window	
navigation within .....	135
usage .....	375
Test all files everywhere	
Show best matches when testing all .....	48
TEST button	
defined .....	45
usage .....	86
Test Disabled field	
defined .....	37
Test File column	
heading .....	257, 382
Test File field	
defined .....	44
Test File option	
import Web page .....	389
Test File window	
usage .....	368
Test files for this category option	
select .....	380
Test multiple files	
Concordance window .....	62, 497
concordance window .....	62
Testing	
Save Test Results .....	378
Testing menu	
Browser .....	20
Clear Test Document .....	19
Decrease Font Size .....	20
Delete Selected Test File .....	18
Full Test Report .....	19
Import Failing Test Files .....	18
Import Test Files .....	18
Increase Font Size .....	20
Open Test Document .....	19
Populate Testing Paths .....	19
Remove Tags .....	20
Restore Populate Results .....	19
Save Test Document .....	20
Save Test Results .....	18

---

---

Show Graphical Populate Results .....	19
Show Last Test Report .....	19
View Saved Results .....	18
testing messages	
defined .....	377
Testing Path field	
defined .....	38
testing paths	
central repository .....	361
set .....	87
testing process	
customize .....	374
testing rules	
concepts .....	435
testing set	
UTF-8 encoding .....	103
Testing tab	
defined .....	11, 26
Testing window	
operations .....	47
testing results .....	382
usage .....	31, 375
Text Find	
Edit menu .....	13
Text Replace	
Edit menu .....	13
Text View	
defined .....	29
text view	
Web page .....	389
Text View option	
write rules .....	271
Text View selection	
write rules .....	223
tg_status.xml	
defined .....	370
The .....	362
Thesaurus field	
defined .....	37
threshold weight	
total weight .....	273
weighted linguistic rules .....	270

---



---

Tokenize Classifier Terms	
Concept window .....	58, 429
Total	
Category Test Report window .....	89
total weight	
threshold weight .....	273
Training Path field	
defined .....	38
Tree Find	
Edit menu .....	13
Tree Find icon	
usage .....	76
Tree Find window	
access .....	77
Find field .....	77
Find Next button .....	77
Match case option .....	77
Replace button .....	77
usage .....	76
Tree Replace	
Edit menu .....	13
Tree Replace window	
usage .....	78

## U

unmatched files	
Directory for Unmatched Populate Files .....	366
Unmatched Populate Files	
Select a Directory .....	366
unqualified linguistic rules	
writing .....	269
Upload Categorizer	
Build menu .....	14
Upload Categorizer to CatCon Server	
window .....	91
Upload Concepts	
Build menu .....	15
Upload Liti	
Build menu .....	15

---

Use Project Default field	
defined .....	37
Use same name for both fields	
Enter Names window .....	73
Use UTF-8 Test Files	
Misc window .....	55
user interface	
display .....	10
UTF-8 encoding	
language fonts .....	103
testing set .....	103

## V

verb form	
category rule .....	278, 303
View menu	
Number of Taxonomy Nodes .....	14
Refresh Tree .....	14
Taxonomy as Text .....	14
View Rule Matches	
usage .....	85
View Rule Matches option	
defined .....	45
View Rule Matches window	
usage .....	40, 41
View Saved Results	
Testing menu .....	18

## W

Web browser	
Document window .....	43
Web document	
Boolean rules .....	309
Web page	
text view .....	389
weight	
weighted category rule .....	273
weighted category rule	
weight .....	273

---

weighted linguistic rules	
threshold weight .....	270
writing .....	269
Weighting of results .....	36
writing and testing rules	
concepts .....	435

## X

XML Default Field	
Misc window .....	56, 318, 319
XML file	
categorization .....	520
concept extraction .....	527
XML Tags to Ignore	
Misc tab .....	56

## Z

Zone-Based	
defined .....	51
Zone-based frequency	
algorithm .....	248
Zone-based relevancy option	
Category window .....	248
Zone-Based selection	
Relevancy Type .....	60

